



Universität Ulm | 89069 Ulm | Germany

ulm university universität
uulm

**Fakultät für
Ingenieurwissenschaften
und Informatik**

Institut für Datenbanken und
Informationssysteme

MASTERARBEIT

COLLABORATIVE MODELING OF BUSINESS PROCESSES ON CO-LOCATED TABLETOP SYSTEMS

VORGELEGT VON

Sebastian Ronis

sebastian.ronis@uni-ulm.de

GUTACHTER

Prof. Dr. Manfred Reichert

Prof. Dr. Peter Dadam

BETREUER

Jens Kolb

2014

Fassung 25. September 2014

© 2014 Sebastian Ronis

KURZZUSAMMENFASSUNG

Most current Business Process Management Systems (BPMS) refer to single users, working at a desktop PC individually. But especially, during the creation of process models, domain and modeling experts work together. Therefore, a collaborative BPMS offers possibilities to work in a team environment. The advantages of collaborative process modelling are improved quality and accuracy of process models. Thus, the user's workload is reduced and the users learn from each other.

This thesis presents *Process-Touch*, which is a collaborative BPMS, using a tabletop and additional tablets or smartphones to create process models collaboratively. *Process-Touch* offers an easy to use Natural User Interface (NUI), sketch-based input and the possibility to work with tablets or smartphones as a private interaction device. Users can create and edit parts of the process model on their tablets and smartphones and merge them on the tabletop using a tap gesture to transfer the process model from the mobile device to the tabletop.

This thesis contributes a system for collaborative BPMS, general concepts and requirements and a prototypical implementation. Moreover, the concept is evaluated by experimental research, using the prototypical implementation. Hence, improvements of the gesture-set, interaction design and implementation are identified and discussed.

DANKSAGUNG

Besonders bedanken möchte ich mich bei meinem Betreuer Jens Kolb, mit dem ich während des Studiums interessante Projekte bearbeiten konnte, der mich immer gefördert hat und mich durch die Masterarbeit geleitet hat.

Bei Herrn Prof. Dr. Manfred Reichert bedanke ich mich herzlich dafür, dass ich die Arbeit im Institut anfertigen konnte, für die gute Ausstattung im Institut und die Begutachtung der Arbeit.

Aus tiefsten Herzen möchte ich mich bei meinen Eltern bedanken, die mir die Möglichkeit gegeben haben einen langen und guten Ausbildungsweg zu beschreiten. Ohne deren Unterstützung wäre es für mich nicht möglich gewesen das Studium zu absolvieren.

INHALT

1	Einleitung.....	1
1.1	Motivation.....	1
1.2	Problemstellung.....	1
1.3	Beitrag der Arbeit.....	2
1.4	Aufbau der Arbeit.....	2
2	Related Work.....	3
3	Grundlagen	7
3.1	Business Process Model and Notation.....	7
3.2	Hardware	11
3.2.1	Near Field Communication.....	11
3.2.2	Microsoft Pixel Sense.....	12
3.3	Touch-Eingabe mit mobilen Geräten.....	13
3.3.1	Erkennung einer Touch-Aktion auf mobilen Geräten	13
3.3.2	Präparierte Oberflächen.....	14
3.3.3	Tabletop-System	15
3.4	Skizzenbasierte Eingabe	15
3.5	Gestaltgesetze	17
4	Konzept Process-Touch.....	21
4.1	Anforderungen an das Konzept.....	21
4.2	Gesamtkonzept.....	30
4.2.1	<i>Alternativer Aufbau</i>	32
4.2.2	Erstellung von Prozessmodellen.....	33
4.2.3	Bearbeitung von Prozessmodellen.....	34
4.2.4	Interaktion mit gedruckten Prozessmodellen	34
4.3	Zusammenfassung.....	36
5	Interaction-Design	39
5.1	Allgemeine Interaktion.....	39
5.2	Gesten-Set für Benutzereingaben.....	42
5.3	Benutzereingabe mit mobilen Geräten.....	47
5.4	Prüfung der korrekten BPMN Syntax.....	48
5.5	Layouting von Prozesselementen	49

5.6	Zusammenfassung.....	52
6	Visual Design.....	53
6.1	Farben in Process Touch.....	53
6.2	Icons und Typographie.....	55
6.3	Formen der Interaktionselemente.....	57
6.4	Interaktionselemente.....	58
6.5	Rückmeldungen.....	62
6.5.1	Syntaxprüfung.....	63
6.5.2	Tablet-Touch-Aktionen.....	63
6.6	Zusammenfassung.....	65
7	Proof-of-Concept-Implementierung.....	67
7.1	Softwarearchitektur.....	68
7.1.1	Das Design-Pattern Model-View-ViewModel.....	69
7.1.2	Portable Class Library.....	70
7.2	Umsetzung des Prototypen.....	71
7.2.1	Model-Klassen.....	71
7.2.2	ViewModel-Klassen.....	72
7.2.3	View des Tabletop-Systems.....	73
7.2.4	View der Windows-Store-App.....	77
7.3	Kommunikation und Datenaustausch.....	77
7.4	Skizzenbasierte Benutzereingabe.....	82
7.4.1	Vereinfachung eines Polygons.....	83
7.4.2	Symbolerkennung.....	85
7.4.3	Multiselektion von Prozesselementen.....	89
7.5	Layouting der Prozesselemente.....	90
7.6	Zusammenfassung.....	91
8	Experimentelle Validation.....	93
8.1	Forschungsfrage und Experimentdefinition.....	93
8.1.1	Kontextauswahl und Zieldefinition.....	94
8.1.2	Hypothesendefinition.....	94
8.1.3	Aufbau des Experiments.....	96
8.1.4	Gestaltung des Experiments.....	102
8.1.5	Validität des Experiments.....	103

8.2	Ablauf des Experiments	105
8.2.1	Vorbereitung.....	105
8.2.2	Durchführung	105
8.2.3	Validation der Daten.....	107
8.3	Datenanalyse und Interpretation.....	107
8.3.1	Datenanalyse und Beschreibende Statistik.....	107
8.3.2	Hypothesentest.....	110
8.4	Qualitative Auswertung.....	110
8.4.1	Tabletop-System	111
8.4.2	Gesten.....	112
8.4.3	Layouting von Prozessmodellen.....	113
8.4.4	Menüs	114
8.4.5	Vollständigkeit von BPMN.....	115
8.4.6	Implementierungsfehler	115
8.5	Zusammenfassung	116
9	Zusammenfassung und Ausblick.....	119
A.	ISO-Norm 9241/10.....	ix
B.	User Experience Questionnaire	xvii
C.	Allgemeine Fragen	xix
D.	Leitfaden für den Experimentleiter	xxi
E.	Einverständniserklärung.....	xxiii
F.	Anwendungsszenarien.....	xxiv
G.	Prozessmodelle der Probanden	xxvii
H.	Daten des Experiments.....	xxxvii

1 EINLEITUNG

1.1 MOTIVATION

Die Vorteile von *Business Process Management*, oder auch *Geschäftsprozess-Management* genannt, liegen klar auf der Hand: Transparenz, Effizienz und Agilität der Organisation wird maßgeblich erhöht und folglich die Kosten gesenkt [1]. Ausgereifte Technologien erlauben einen breiten Einsatz verschiedenster Prozesse auf unterschiedlichen Prozessplattformen. Jedoch ist die Erstellung eines Prozessmodells aufwändig und oft auch zeitintensiv. So werden 40% der Arbeitszeit in Unternehmen dazu verwendet, Prozessmodelle zu erstellen [2]. Der Aufwand rechtfertigt sich aus der Tatsache, dass die Geschäftsprozessmodellierung eine der strategisch wichtigsten Aufgaben von Unternehmen ist, um dem Grundsatz der *Arbeitsteilung* und *Spezialisierung* nachzukommen [2]. So ist das primäre Ziel die Verbesserung von betrieblichen Arbeitsabläufen.

1.2 PROBLEMSTELLUNG

Aktuell sind zwei Arten der Kollaboration bei Software-Systemen weit verbreitet. Benutzer sitzen getrennt an ihren jeweiligen Computern oder Notebooks und erledigen kollaborative Aufgaben gemeinsam. Bei eng gekoppelten Arbeiten bedarf dies einer detaillierten Absprache und Trennung der jeweiligen Aufgaben und geht mit einem enormen Mehraufwand einher. Damit Benutzer Lösungen gemeinsam erarbeiten und diskutieren können, kann auch gemeinsam ein Computer verwendet werden. Jedoch ist diese Art der Kollaboration ineffizient, da immer nur ein Benutzer das System bedienen kann und zeitgleiche Eingaben nicht möglich sind. Bei eng gekoppelten Aufgaben, wie die Zusammenarbeit bei komplexen Problemen, ist die kollaborative Arbeit mit Blickkontakt (*face-to-face*) die beste Art der Zusammenarbeit [3] und kann durch Tabletop-Systemen realisiert werden.

Tabletop-Systeme ermöglichen durch Multi-Touch mehreren Benutzern die zeitgleiche Interaktion mit dem System. Traditionelle BPMS sind jedoch nicht für Multi-Touch-Systeme entworfen und nutzen folglich nicht das volle Potential der Geräte [4].

Daher stellt sich als einleitende Frage, wie auf Tabletop-Systemen kollaborative Modellierung verbessert werden kann, um eng gekoppelte Aufgaben effizient zu erledigen, gemeinsam Lösungen zu finden und diese gemeinsam diskutieren zu können.

1.3 BEITRAG DER ARBEIT

Der Beitrag dieser Arbeit ist die *Identifikation* und *Kombination* geeigneter Konzepte aus dem Bereich der Human-Computer-Interaction (HCI), für die kollaborative Geschäftsprozessmodellierung auf Tabletop-Systemen zu einem konsistenten Gesamtsystem. Aus einem Teil des Gesamtsystems wird ein Software-System entworfen und prototypisch implementiert. Anschließend wird mithilfe des Prototyps der das System in einem Experiment validiert und Verbesserungen werden aufgezeigt. Der Fokus liegt dabei auf der Implementierung und Validation.

1.4 AUFBAU DER ARBEIT

Kapitel 3 gibt einen Überblick über Geschäftsprozessmodellierung sowie technische Grundlagen, welche für die Arbeit relevant sind. Spezielle Hardware, wie das SUR40 Tabletop-System von Samsung, wird vorgestellt und die dadurch möglichen Interaktionsformen werden erklärt. Anschließend wird ein kurzer Abriss in die Wahrnehmungspsychologie geben, um die Grundlagen der Arbeit zu definieren (siehe Kapitel 3).

In Kapitel 4 wird anhand aktueller Literatur ein Gesamtkonzept für die kollaborative Geschäftsprozessmodellierung entworfen.

Um das Gesamtkonzept in ein Software-System zu überführen, beschäftigt sich Kapitel 5 mit Detaillösungen und widmet sich speziell dem Interaction-Design. Das feingranulare Konzept der Software wird anschließend (siehe Kapitel 6) graphisch aufbereitet um ästhetischen, als auch hardwarebedingten Anforderungen gerecht zu werden. Die technische Umsetzung wird in Kapitel 7 beschrieben. Um das Konzept des lauffähigen Software-Systems zu validieren, wird in Kapitel 8 ein Experiment definiert, durchgeführt und ausgewertet. Abschließend wird die Arbeit zusammengefasst und ein Ausblick wird gegeben (siehe Kapitel 9).

2 RELATED WORK

[5] untersucht, ob Tabletop-Systeme für die Prozessmodellierung geeignet sind und kommt zu dem Schluss, dass es besser ist als ein Desktop-PC zu verwenden, wenn Domänen- und Modellierungs-Experten zusammenarbeiten. Der Aufbau der Oberfläche schränkt jedoch die Kollaboration stark ein. So ist nur eine einzige virtuelle Tastatur auf dem Tabletop-System vorhanden und durch die Verwendung der Hardware (d.h. Microsoft Surface 1) ist es nicht möglich eine bequeme Sitzposition einzunehmen. Die Eingabe ist menübasiert und erinnert stark an Systeme, welche mit Maus und Tastatur bedient werden. Das Menü ist auch nur von einer Seite zugänglich. Diese Schwachstellen werden in der vorliegenden Arbeit verbessert.

Das System *Augemented Surfaces* zeigt, wie Notebooks mit einem gemeinsamen Display für kollaboratives Arbeiten verbunden werden können [6]. Das Display kann zum einen auf den Tisch projiziert werden, zum anderen aber auch an die Wand. Die Interaktion basiert auf Eingaben mit Maus und Tastatur. Der Fokus des Systems wurde auf das einfache und intuitive Teilen von Dokumenten gelegt. So kann mittels Drag&Drop ein Dokument geteilt oder zu eigen gemacht werden. Dokumente können ausgedruckt werden, indem sie mittels Drag&Drop auf den Drucker gezogen werden, der sich innerhalb des projizierten Displays befinden muss. Der Einsatz des Systems ist Aufgrund des komplexen Aufbaus (Kameras und Projektoren) nur in speziell hierfür ausgestatteten Räumen möglich.

[7] weist mittels Studien die Vorteile von kollaborativer Geschäftsprozessmodellierung (*Collaborative Business Process Modeling*; CBPM) nach. So zeigen sie auf, dass die Kommunikation erleichtert wird, Ideen besser ausgetauscht werden können und Aufgaben einfacher zu lösen sind. Weiter konnten die Probanden von ihren Teamkollegen lernen. Im Versuchsaufbau verwendeten sie Tablets welche mit Touch-Eingaben bedient wurden.

[8] weist in einer vergleichenden Studie nach, dass bei CBPM sich Softwareunterstützung positiv auf die Qualität der Modelle auswirkt. Die Probanden arbeiten bei dem Versuchsaufbau jeweils an einem Computer mit Maus und Tastatur.

Tabletop Concept Mapping beschreibt ein Werkzeug, mit diesem Benutzer individuelle Gedanken und Ideen auf einem Tabletop-System modellieren und teilen können [9]. Die verwendete Syntax der Modelle ist hierbei den Benutzern überlassen. Für die Modellierung werden physische Objekte verwendet, welche die Benutzer auf den Tabletop legen und mithilfe der angebotenen Funktionen verbinden können. Beschriftet werden können diese Objekte mit Post-Its oder digital mittels einer externen Tastatur. Bei der digitalen Beschriftung wird der Name unterhalb des Elements auf dem Tabletop-System angezeigt.

[10] zeigt zwei verschiedene Ansätze um verteilte Prozessmodelle auf Tabletop-Systemen miteinander zu verbinden. Dabei bedient er sich zum einen der Metapher der Knopf-basierten Interaktion, zum anderen der Ablage-basierten. Bei der Knopf-basierte Interaktion können Benutzer durch drehen eines physischen Interaktionselements auf dem Tabletop-System Werte auswählen. Die Ablage-basierte Interaktion überträgt gewohnte Menüs aus Graphical User Interfaces (GUI) auf Tabletop-Systeme. Dabei können Benutzer ein Interaktionselement, welches auf einem anderen Tabletop-System erstellt wurde, aus einer Liste auswählen und per Drag&Drop auf das Tabletop-System übertragen. Die Interaktionselemente der Liste müssen zuerst vom entfernten Tabletop-System angeboten werden und dienen als Verbindungselement der beiden Prozessmodelle.

In [11] werden neue Ansätze der Prozessmodellierung erforscht. Basierend auf der Tatsache, dass Domänen-Experten ihr Wissen meist nicht selbstbewusst in Modellierungssitzungen ausdrücken können, verwenden sie papiergestützte Modellierungsverfahren. Dadurch soll es den Domänen-Experten, die sich mit Prozessmodellierung wenig auskennen, erleichtert werden, ihr Wissen mitzuteilen. Dieses Verfahren ist bei benutzerzentrierten Anwendungen und im Produktdesign gängig.

[12] beschreibt die Interaktionstechnik *Touch&Interact*, um mit Smartphones Touch-Aktionen an jeder Stelle eines präparierten Monitors durchzuführen. In einem experimentellen Aufbau, in welcher diese Technik mittels eines Computerprogramms für Touristen validiert wurde, haben sie gezeigt, dass die Performanz dieser Interaktionstechnik, mit der eines Touchscreen vergleichbar ist. Die Zufrie-

denheit der Probanden variiert, ist jedoch immer im positiven Bereich und das Konzept stößt nicht auf Ablehnung seitens der Benutzer. In einer zweiten Studie, in der Fotos zwischen einem Smartphone und einem Computer ausgetauscht wurden, war die Zufriedenheit bei Touch&Interact sogar höher, als bei der Verwendung eines normalen Touchscreens. Dies zeigt, dass abhängig des Systems und der Software die Verwendung von Smartphones als Touch-Eingabegerät nicht störend, sondern als fördernd angesehen werden kann.

[13] zeigt die Möglichkeit auf, kollaborativ auf Tabletop-Systemen Prozessmodelle zu erstellen. Benutzer können mehrere virtuelle Tastaturen verwenden, um gleichzeitig Text einzugeben. Zusätzlich werden individuelle Ansichten angeboten, um Kopien eines Bereichs des Interfaces zu erstellen. Um Konflikte zu lösen, wenn mehrere Benutzer ein geteiltes Interaktionselement bearbeiten, wird ein Änderungsverfahren angeboten. Weiter zeigt [13] auf, dass auch Benutzer mit wenig Kompetenz in der Prozessmodellierung sich durch das verwendete System gut in die gemeinschaftliche Gruppenarbeit einbringen können. Dabei sind sie mit dem Modellierungsergebnis zufrieden und verstehen das Prozessmodell.

In [14] wird ein System für die kollaborative Prozessmodellierung auf Tabletop-Systemen beschrieben. Die Hauptansicht des Systems ist dabei in vier Bereiche für jeweils einen Benutzer untergliedert. Die Erstellung von Prozessmodellen basiert auf Menüs und es ist nur eine virtuelle Tastatur vorhanden. Dabei werden Texteingaben durch einen modalen Dialog dargestellt, welcher die gleichzeitige Interaktion mehrerer Benutzer unterbindet.

3 GRUNDLAGEN

Dieses Kapitel erläutert die Grundlagen für die Konzeption eines kollaborativen Prozessmodellierungssystems. Dabei gibt Kapitel 3.1 einen Überblick über BPMN, um den Kontext und das Domänenwissen dieser Arbeit zu erläutern. In Kapitel 0 zeigt auf, wie mit Tablets und Smartphones Touch-Aktionen durchgeführt werden können. Kapitel 3.4 beschreibt Ansätze aus dem Bereich der *Human-Computer-Interaction* (HCI), um mit skizzenbasierter Eingabe die Prozessmodellierung zu unterstützen. Grundlegende Gestaltgesetze werden in Kapitel 3.5 aufgezeigt.

3.1 BUSINESS PROCESS MODEL AND NOTATION

Die *Business Process Model and Notation* (BPMN) ist eine graphische Spezifikationsprache, um *Prozessmodelle* zu modellieren. Ein Prozessmodell ist dabei ein gerichteter Graph, welcher aus *Kanten* und *Knoten* besteht. Die unterschiedliche Darstellung und Semantik der Kanten und Knoten wird von der BPMN definiert. Die BPMN Version 2.0 ist von der *Object Management Group* entwickelt worden und findet in dieser Arbeit ihren Einsatz.

Die BPMN 2.0 gliedert sich in drei Modellarten, dem *Choreography Diagram*, dem *Conversation Diagram* und dem *Collaboration Diagram* [15].

Prozesselemente der BPMN lassen sich, wie in Abbildung 3-1 gezeigt, in die fünf Hauptkategorien *Flussobjekte*, *Daten*, *Verbindende Objekte*, *Verantwortlichkeitsbereich* und *Artefakte* unterteilen.

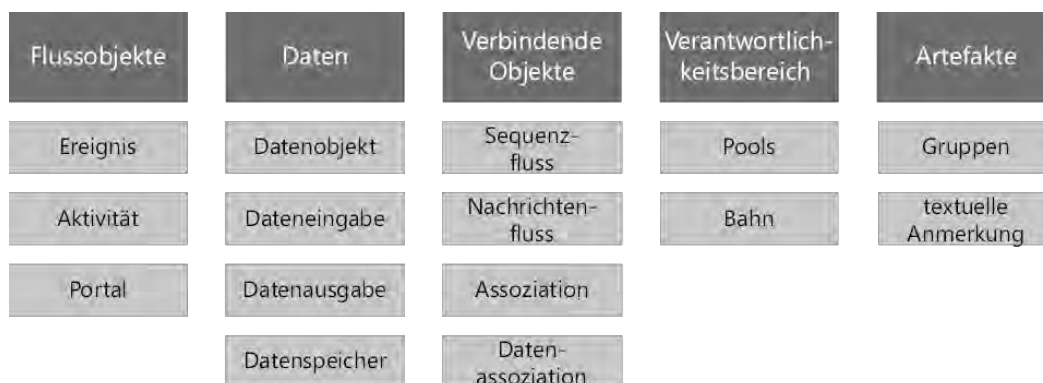


Abbildung 3-1: BPMN-Kategorien

Nachfolgend werden jene Prozesselemente vorgestellt, welche für diese Arbeit relevant sind [15].

Ereignis

Ein *Ereignis* (engl.: *Event*) ist etwas, das während dem Ablauf eines Prozesses *pas-siert* (siehe Abbildung 3-2). Solch ein Ereignis beeinträchtigt den Ablauf des Prozessmodells. Es gibt drei verschiedenen Arten von Ereignissen, je nachdem an welcher Stelle sich diese im Prozessmodell befinden. Ein *Start-Ereignis* hat keine eingehenden Kanten und steht den Startpunkt des Prozesses dar. Ein *Zwischen-Ereignis* hat eingehende wie auch ausgehende Kanten, wobei ein *Ende-Ereignis* nur eingehende Kanten besitzt.



Abbildung 3-2: Ereignisse in BPMN

Sequenzfluss

Ein *Sequenzfluss* (engl.: *Sequence Flow*) wird verwendet, um die Reihenfolge der Prozessausführung aufzuzeigen (siehe Abbildung 3-3).

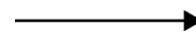


Abbildung 3-3: Sequenzfluss in BPMN

Nachrichtenfluss

Ein *Nachrichtenfluss* (engl.: *Message Flow*) wird verwendet um den Fluss von Nachrichten zwischen zwei Teilnehmern zu visualisieren (siehe Abbildung 3-4). Verschiedene *Pools* repräsentieren dabei verschiedene Teilnehmer.

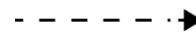


Abbildung 3-4: Nachrichtenfluss in BPMN

Assoziation

Eine *Assoziation* (engl.: *Association*) verbindet Artefakte (z.B. Datenobjekte, textuelle Anmerkungen) mit Prozesselementen (siehe Abbildung 3-5). Die Pfeilspitze gibt die Richtung der Assoziation an.

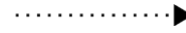


Abbildung 3-5: Assoziation in BPMN

Aktivität

Eine *Aktivität* (engl.: *Activity*) ist ein generischer Ausdruck für *Arbeit*, welche das Unternehmen in dem Prozess verrichtet. Eine Aktivität kann dabei *atomar* oder *nicht-atomar* sein. Ein *Subprozess* stellt selbst wieder ein Prozessmodell dar und dient somit als atomare Aktivität. Ein *Tasks* stellt eine einzelne Aufgabe innerhalb eines komplexen Prozesses dar (siehe Abbildung 3-6).

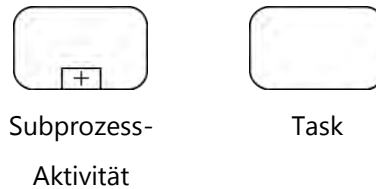


Abbildung 3-6: Aktivitäten in BPMN

Portal

Ein *Portal* (engl.: *Gateway*) wird verwendet, um den Sequenzfluss in Prozessmodellen zu steuern. Ein Gateway kann sowohl einen Sequenzfluss aufspalten oder zusammenführen. AND-Gateways symbolisieren eine parallele Aufspaltung, bzw. eine Zusammenführung paralleler Sequenzflüsse. XOR-Gateways repräsentieren eine exklusive Aufspaltung, bzw. Zusammenführung verschiedener Sequenzflüsse (siehe Abbildung 3-7).

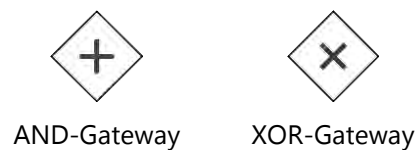


Abbildung 3-7: Portale in BPMN

Datenobjekt

Ein *Datenobjekt* (engl.: *Data Object*) stellt Informationen für andere Prozesselemente bereit oder nimmt Informationen von diesen entgegen (siehe Abbildung 3-8). Datenobjekte können sowohl eine einzelne Variable, als auch eine Vielzahl von Variablen repräsentieren.



Abbildung 3-8: Datenobjekt in BPMN

Pool

Ein *Pool* ist eine graphische Repräsentation eines Teilnehmers innerhalb des Prozessmodells und dient als Container für andere Prozesselemente (z.B. Aktivitäten und Gateways) (siehe Abbildung 3-9).



Abbildung 3-9: Pool in BPMN

Bahn

Eine *Bahn* (engl.: *Lane*) ist ein Teilabschnitt eines Prozessmodells oder Pools. Sie erweitert einen Prozessmodell und werden verwendet, um Aktivitäten zu organisieren und kategorisieren (siehe Abbildung 3-10).

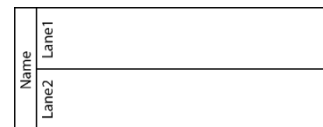


Abbildung 3-10: Lane in BPMN

Textuelle Anmerkung

Textuelle Anmerkungen (engl.: *Text Annotation*) bieten Modellierern die Möglichkeit, Erklärungen und Informationen einzugeben und werden über *Assoziationen* mit einem Prozesselement verbunden (siehe Abbildung 3-11).

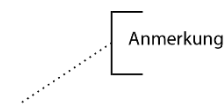


Abbildung 3-11: Anmerkung in BPMN

Ein Beispiel eines Prozessmodells ist in Abbildung 3-12 zu sehen. Das Start-Ereignis gibt den Startpunkt des Prozessmodells an. Anschließend wird durch das AND-Gateway der Sequenzfluss aufgespalten. Folglich werden die beiden Aktivitäten *Task A* und *Task B* gleichzeitig ausgeführt und durch das nächste AND-Gateway wieder zusammengeführt. Der Sequenzfluss wird erst fortgesetzt, wenn beide Aktivitäten abgearbeitet sind. Der nächste Block, welcher aus dem *zusammenführenden XOR-Gateway*, *Task C* und dem *verzweigenden XOR-Gateway* besteht, stellt eine Schleife dar. Im ersten Durchgang hat das zusammenführende XOR-Gateway keine Funktion und *Task C* wird ausgeführt. Das verzweigende XOR-Gateway prüft die *Bedingungen*, welche neben den Sequenzflüssen angegeben werden und entscheidet, abhängig des Wertes von *i*, über die weitere Ausführung. Solange die Bedingung $i < 3$ zutrifft, wird *Task C* ausgeführt. Sonst wird der Sequenzfluss fortgeführt und das Ende-Ereignis zeigt das Ende des Prozesses an.

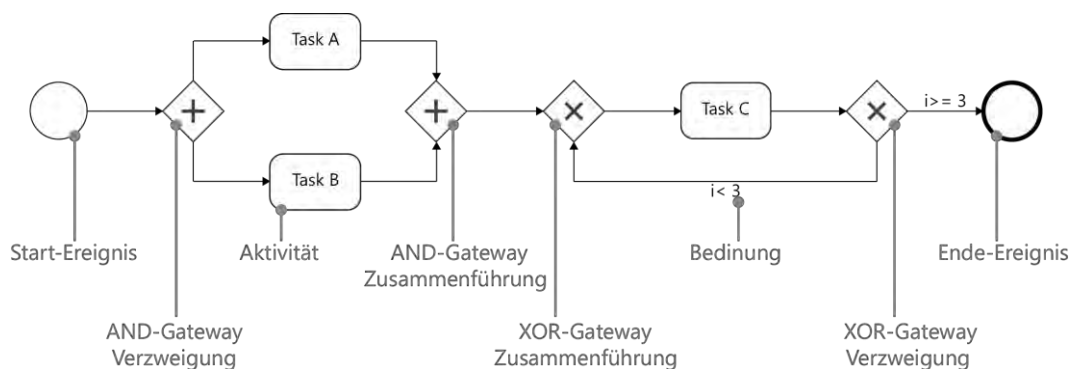


Abbildung 3-12: Prozessmodell

3.2 HARDWARE

3.2.1 NEAR FIELD COMMUNICATION

Near Field Communication (NFC) wird verwendet um zwischen zwei Geräten eine drahtlose Verbindung aufzubauen und Informationen zu übermitteln. Basierend auf der *RFID*-Technologie bietet *NFC* eine Kommunikation, welche räumlich auf nur wenige Zentimeter beschränkt ist. Die Kommunikation kann zwischen einem passiven *NFC*-Gerät (*NFC-Tag*) und einem aktiven *NFC*-Gerät lesend wie auch schreibend, nur lesend, sowie zwischen zwei aktiven *NFC*-Geräten bidirektional statt-finden [16].

Ein *NFC-Tag* bezeichnet ein passives, elektronisches Gerät, welches selbst an keine Stromquelle angeschlossen ist. Kommt ein aktives NFC-Gerät, welches an eine Stromquelle angeschlossen ist, in die Nähe des Tags, so wird dieser drahtlos via Induktion mit Strom versorgt. Dadurch können Informationen vom NFC-Tag ausgelesen werden, um z.B. Webseiten zu öffnen, andere Verbindungen (z.B. WLAN oder Bluetooth) zu initiieren oder spezifische Daten auszulesen. Alternativ können auch Informationen auf den NFC-Tag geschrieben werden, sofern dieser den schreibenden Zugriff unterstützt.

Die limitierte Reichweite stellt hingegen der augenscheinlichen Schwäche die größte Stärke dieser Technologie dar. So kann das Vorhandensein einer Verbindung als Zustimmung einer Aktion angesehen werden. Anders ausgedrückt: Sobald ein anderes NFC-Gerät in Reichweite ist, kann davon ausgegangen werden, dass dies vom Benutzer beabsichtigt ist. Die Information kann ohne weitere Zustimmung des Benutzers ausgelesen werden und Aktionen können gestartet werden [16].

3.2.2 MICROSOFT PIXEL SENSE

Der Sur40 von Samsung, welcher Microsofts *Pixel Sense* Technologie unterstützt, ist ein 40 Zoll großer, *berührungsempfindlicher LCD-Bildschirm* (Touchscreen), welcher horizontal aufgestellt und als Tabletop-System verwendet, oder vertikal an die Wand gehängt werden kann [17]. Die Detektion der Benutzereingaben erfolgt bei dem, 2007 von Samsung vorgestellten Tisch, optisch. Bei konventionellen Touchscreens werden Eingaben über Ladungsunterschiede erkannt. Der Pixel Sense hingegen filmt von unten alles was über dem Display geschieht und erkennt verschiedene physische Objekte. Der Vorteil der optischen Detektion besteht darin, dass sich ein *Finger*, ein *Objekt* oder ein *getagtes Objekt* unterscheiden lassen. Wird ein Objekt auf das Tabletop-System gelegt, erscheinen, je nach Software, Interaktionselemente, um mit diesem zu interagieren. Durch die Auflösung des Displays von 1920×1080 wird von ca. zwei Millionen Bildpunkten ausgegangen. Da jeder achte Pixel als Kamera fungiert, werden folglich ca. 26.000 Touchpunkte erkannt [17].

3.3 TOUCH-EINGABE MIT MOBILEN GERÄTEN

Dieser Abschnitt zeigt Interaktionskonzepte auf, wie Benutzer ihre eigenen *mobilen Geräte* (z.B. Smartphone oder Tablet) verwenden können, um mit einem Software-System zu interagieren.

Der Benutzer verwendet dabei mobile Geräte, um mit diesen eine Oberfläche zu Berühren und dadurch Aktionen auszulösen. Wird eine Berührung nicht nur von der Oberfläche, sondern zusätzlich von dem mobilen Gerät erkannt, lässt sich daraus schließen, welcher Benutzer eine bestimmte Aktion ausführt. Dazu wird auf beiden Geräten der exakte Zeitpunkt der erkannten Touch-Aktion gespeichert und über ein drahtloses Netzwerk miteinander geteilt. Anschließend werden die Zeitpunkte miteinander verglichen. Korreliert der Zeitpunkt der Touch-Aktionen beider Geräte wird ein *TabletTouch* erkannt.

Die folgenden Abschnitte zeigen auf, wie eine Berührung auf den mobilen Geräten und auf unterschiedlichen Oberflächen erkannt werden kann.

3.3.1 ERKENNUNG EINER TOUCH-AKTION AUF MOBILEN GERÄTEN

Als universelles Verfahren können die Beschleunigungssensoren der Geräte ausgelesen werden, welche bei einer Berührung kurz und stark ausschlagen [18]. Bei normalem Gebrauch der mobilen Geräte in der Hand, sind die Beschleunigungswerte deutlich geringer und lassen sich somit eindeutig von einer Berührung unterscheiden. Durch diese Eigenschaft lässt sich auf einfache Art eine Berührungsdetektion umsetzen.

Als weiteres Verfahren kann zusätzlich zu den Beschleunigungswerten das Mikrofon des mobilen Geräts, mit dem Interagiert werden soll, verwendet werden [18]. Da jede Berührung an einem Gegenstand mit einem spezifischen Geräusch einhergeht, kann dieses vom Mikrofon erfasst werden. Die Methode lässt sich jedoch nicht in Räumen mit hohen Umgebungsgeräuschen verwenden. Aber es kann zusätzlich zu den Beschleunigungswerten als Indiz angesehen werden.

Der Nachteil der indirekten Bestimmung einer Berührung ist, dass beide Verfahren nur mit harten Oberflächen, wie z.B. einer Tischplatte funktionieren. So kann mit den beiden Verfahren nicht erkannt werden, ob ein Smartphone oder Tablet

z.B. ein weiches Kissen berührt, da es weder eine starke und kurze Abbremsung, noch ein spezifisches Geräusch gibt.

An welcher Stelle die Berührung stattgefunden hat, kann mittels der Beschleunigungssensoren oder dem Mikrofon nicht erkannt werden. Hierfür lässt sich das System auf verschiedene Arten erweitern, welche nachstehend erklärt werden.

3.3.2 PRÄPARIERTE OBERFLÄCHEN

Gedruckte Poster, Monitore oder Projektionsflächen können mit einer NFC-Tag-Matrix versehen werden, um zusätzliche Funktionalität anzubieten [19]. Diese können unter anderem dazu verwendet werden, die Position der mobilen Geräte zu bestimmen. Die Berührung der Oberfläche wird wiederum durch die mobilen Geräte erkannt. Jeder NFC-Tag speichert seine Koordinaten, die dann bei einer Berührung (vgl Kapitel 3.3.1) ausgelesen werden. Mittels einer zuvor initiierten Verbindung (z.B. Bluetooth oder WLAN) können die Koordinaten an den Computer übertragen werden, wie in Abbildung 3-13 zu sehen ist. Zusätzlich kann jedes Smartphone oder Tablet anhand der MAC-Adresse identifiziert und folglich unterschiedliche Benutzer erkannt und unterschieden werden.

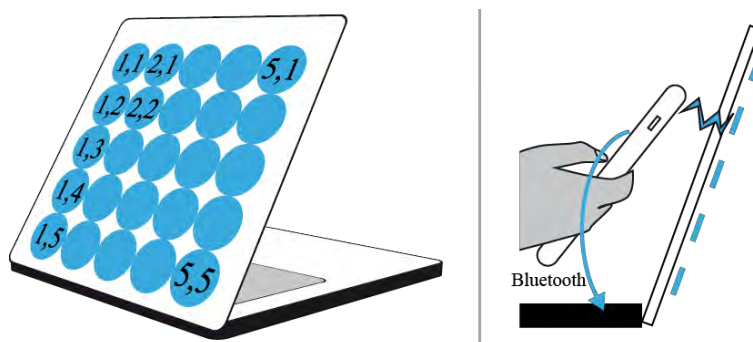


Abbildung 3-13: Prinzip von Touch&Interact [12]

Durch die Verwendung einer NFC-Tag-Matrix ist die Auflösung stark begrenzt und eine feine Granularität der Touchpunkte ist nicht möglich, da ein normaler NFC-Tag ca. 3,5cm x 2cm groß ist.

Das Beispiel in Abbildung 3-13 zeigt eine NFC-Tag-Matrix mit 25 NFC-Tags. Somit sind nur 25 Touchpunkte unterscheidbar. Die Möglichkeit durch unterschiedliche Signalstärken der einzelnen Tags eine Interpolation (d.h. Triangulation) und

somit eine genauere Positionsbestimmung durchzuführen, wurde noch nicht untersucht.

3.3.3 TABLETOP-SYSTEM

Werden auf dem Tabletop-System Benutzereingaben optisch detektiert, kann das vom Tabletop-System aufgenommene Bild dazu verwendet werden, Berührungen mit mobilen Geräten zu erkennen. Dabei unterscheidet sich die Größe der Fläche der Touch-Eingabe von Fingern und mobilen Geräten. Bei mobilen Geräten ist die erkannte Fläche kleiner als bei Eingaben mit dem Finger. Der Aufbau ist in Abbildung 3-14 skizziert.

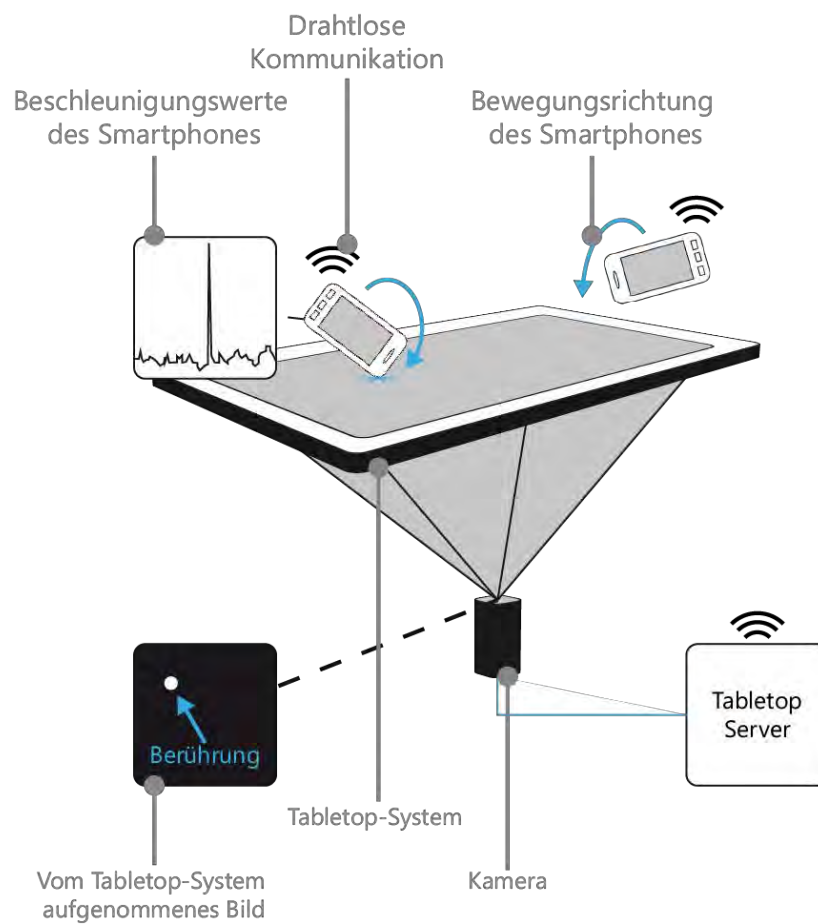


Abbildung 3-14: Aufbau von PhoneTouch [18]

3.4 SKIZZENBASIERTE EINGABE

Die Idee skizzenbasierter Eingabe ist, dass Prozessmodelle per Hand gezeichnet werden und anschließend vom Computer analysiert und korrekt interpretiert werden, um sie anschließend automatisch in das System zu überführen. Da das Sys-

tem die gemalten Formen, Linien und Pfeile korrekt interpretieren muss, ist der Benutzer dazu gezwungen, genormte Zeichen und Symbole zu verwenden. Die Erstellung von Prozessmodellen mittels Skizzen kann mit einem Stift auf Papier oder mit dem Finger auf einem Touchscreen erfolgen.

Die erste Möglichkeit, Prozessmodelle zu skizzieren, besteht darin, die Prozessmodelle mittels Stift auf Papier zu skizzieren und anschließend die Skizze einzuscannen oder zu fotografieren (siehe Abbildung 3-15). Das Bild wird vom System analysiert und anschließend das Prozessmodell anhand der Skizze erstellt [20]. Die Erkennung von Handschrift ist prinzipiell möglich, ist aber meist fehleranfällig. Eine Lösung ist, dass bei nicht erkannten Wörtern das Bild der Handschrift angezeigt wird und der Benutzer nachträglich den Text eingeben kann. Das erkannte, im System vorhandene Prozessmodell kann anschließend auf Validität und Vollständigkeit überprüft werden, als Teilstück in ein größeres Prozessmodell eingefügt, oder zur späteren Weiterverarbeitung gespeichert werden.

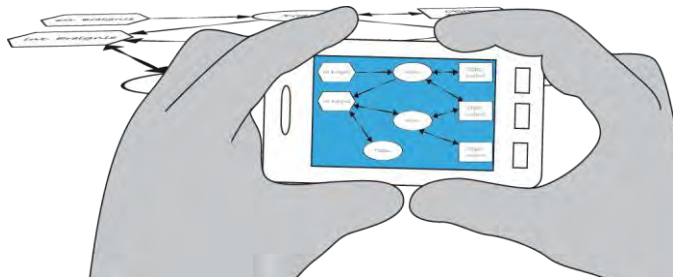


Abbildung 3-15: Automatische Extraktion von Modellen aus einem Bild

Die zweite Möglichkeit der Skizzenbasierten Erstellung von Prozessmodellen besteht darin, dass die Eingabe mittels des Fingers, auf einem Tablet-PC oder Smartphone erfolgt. Diese Interaktion ist auch als elektronisches Whiteboard bekannt [21]. Skizzierte Prozesselemente werden unmittelbar vom System analysiert und das Ergebnis wird angezeigt (siehe Abbildung 3-16). Wird eine Eingabe nicht erkannt, kann sofort eine Rückmeldung gegeben werden und entsprechende Schritte, seitens des Benutzers, können eingeleitet werden. So kann z.B. aus einer Vorschlagsliste ausgewählt werden, welches Objekt der Benutzer skizziert hat, oder der Benutzer wiederholt seine Eingabe [22, 21].

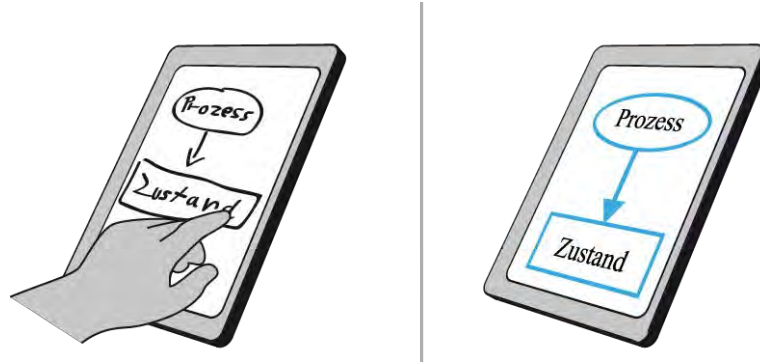


Abbildung 3-16: Skizzenbasierte Eingabe

3.5 GESTALTGESETZE

Dieser Abschnitt beschäftigt sich mit den Grundlagen der menschlichen Wahrnehmung in Bezug auf die Gestaltung einer Benutzerschnittstelle. Es wird gezeigt, wie sich Elemente anordnen und darstellen lassen, um diese zu Gruppieren und dem Benutzer eine schnelle Touch-Interaktion mit dem System zu ermöglichen. Zusätzlich wird auf die Orientierung von Interaktionselementen eingegangen.

Durch die relative Anordnung von Elementen zueinander, lassen sich diese optisch zu *Gruppen* zusammenfassen. In Abbildung 3-17 wird das *Gesetz der Nähe* verwendet. Dieses besagt, dass auf dem Bild drei Spalten mit jeweils fünf Quadraten erkannt werden und nicht fünf Zeilen mit drei Quadraten. Allgemein ausgedrückt wird Zusammengehörigkeit durch Nähe betont und Unterschiede durch Distanz getrennt [23].

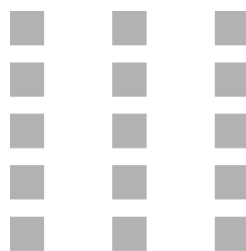


Abbildung 3-17: Gesetz der Nähe

Auf dem *Gesetz der Nähe* aufbauend besagt das *Gesetz der Ähnlichkeit*, dass Elemente mit gleichem Abstand zueinander, durch *Farbe* und *Form* in Gruppen eingeteilt werden können (siehe Abbildung 3-18). So sind in Abbildung 3-18-a die Zeilen, und in Abbildung 3-18-b die Spalten gruppiert. Neben Form und Farbe,

kann auch die *Helligkeit*, *Größe* oder *Orientierung* für die Gruppenbildung verwendet werden [23].

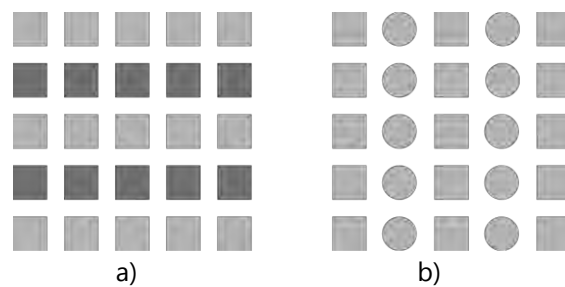


Abbildung 3-18: Gesetz der Ähnlichkeit

Das *Gesetz der Schließung* besagt, dass in unserer Wahrnehmung Konturen möglichst vervollständigt werden. Abbildung 3-19 zeigt ein Rechteck, welches von einem Kreis teils verdeckt wird. Die Hellgraue Figur wird von unserer Wahrnehmung zu einem Rechteck vervollständigt, auch wenn der genaue Verlauf der Kontur unbekannt ist. Der abgeschnittene Kreis wird auch als Kreis wahrgenommen.



Abbildung 3-19: Gesetz der Schließung

Zusammenfassend lässt sich sagen, dass bei einer Kombination der unterschiedlichen Gestaltgesetze keine genaue Aussage darüber getroffen werden kann, welches Gesetz stärker oder schwächer auf unsere Wahrnehmung wirkt. Es gilt jedoch das *Gesetz der guten Gestaltung*. Damit wird generell beschrieben, dass die Wahrnehmung nach möglichst *einfachen*, *regelmäßigen*, *symmetrischen* und *geschlossen* Figuren sucht [23].

Elemente können auch so angeordnet werden, dass Benutzer schneller mit dem System interagieren können. Die Art der Anordnung wird durch *Fitts Gesetz* beschrieben. Die Zeit, welche ein Benutzer braucht um ein Objekt mit dem Finger zu treffen wird dabei als Funktion der Distanz und Größe des Zielobjekts [24] darge-

stellt. Je *näher* das Zielobjekt an der aktuellen Position des Fingers ist, bzw. je *größer* das Zielobjekt ist, desto *schneller* kann der Benutzer dieses erreichen.

Die Orientierung von Elementen nimmt bei Tabletop-Systemen eine Schlüsselrolle ein [25, 26, 27, 28] und soll daher besonders herausgestellt werden. So sollen Tabletop-Systeme bei textlastigen Aufgaben nicht verwendet werden [25]. Da Prozessmodelle hauptsächlich aus beschrifteten Elementen und deren Verbindungen bestehen, scheint es adäquat, Tabletop-Systeme für die Prozessmodellierung zu verwenden. Der Aspekt der Orientierung soll jedoch nicht vernachlässigt werden.

Eine rein manuelle Ausrichtung der Interaktionselemente, d.h. Benutzer rotieren ohne Unterstützung des Tabletop-System Elemente in die gewünschte Richtung, hat den Vorteil, volle Flexibilität in der Interaktion zu bieten. Jedoch sinkt die Lesegeschwindigkeit der Beschriftungen der Interaktionselemente bei schlechter Ausrichtung.

Eng gekoppelte Aufgaben, bei denen eine starke Zusammenarbeit notwendig ist, werden mit rein *manueller* Ausrichtung von Interaktionselementen schneller erledigt als mit *automatischer* Ausrichtung [29]. Der Grund besteht darin, dass Benutzer die Orientierung nicht nur verwenden um selbst besser lesen zu können, sondern auch um Elemente zu teilen und um Informationen zu kommunizieren. Für lose gekoppelte Aufgaben ist kein signifikanter Unterschied zwischen *automatischer* oder *manueller* Ausrichtung vorhanden. Folglich wird für dieses System *keine automatische* Ausrichtung verwendet.

Untersuchungen mit *halbautomatischer* Ausrichtung [28, 27] anhand des *Personal Space* der jeweiligen Benutzer wurden nur an dokumentenorientierten Systemen untersucht und lassen sich folglich nicht auf diese Arbeit übertragen. Das Display soll nicht in Segmente aufgeteilt werden, um beliebig viele Benutzer zu unterstützen [30]. Somit wird auf eine halbautomatische Ausrichtung verzichtet.

Um Elemente in eine bestimmte Richtung zu orientieren, kann ein *Magnet* verwendet werden [31]. Dabei werden alle Elemente, welche sich innerhalb des Wirkungsbereichs des Magneten befinden, in die Richtung des Magneten ausgerich-

tet. Ein solcher Magnet kann zum einen ein spezielles physikalisches Objekt sein, welches auf dem Tabletop-System platziert wird, oder ein virtuelles Element, welches auf dem Display angezeigt wird.

4 KONZEPT PROCESS-TOUCH

Dieses Kapitel zeigt auf, wie ein multi-modales Multi-User-System konzipiert werden kann, welches sich auf aktuelle Erkenntnisse der HCI und BPM stützt, um den Benutzern ein System für kollaborative Zusammenarbeit zu bieten, welches leicht zu bedienen ist. Der Name *Process-Touch* beschreibt dabei den Grundsatz, dass das zu erstellende Prozessmodellierungssystem nur mit Touch-Aktionen bedient wird.

Ausgangspunkt der Arbeit bildet die Forderung nach einem multi-modalen Multi-User-System. Benutzerzentriertes Design ist die Grundlage einer guten HCI [32]. Deshalb wird in dieser Arbeit und vor allem für das Interaction-Design, großer Wert auf einen benutzerzentrierten Ansatz gelegt. Interaction-Design definiert das Wechselspiel zwischen Software und Benutzerverhalten, Rückmeldungen und Touch-Interaktionen [30].

Kapitel 4.1 identifiziert *allgemeine Forderungen* aus der Literatur, welche definieren, wie ein solches System aufgebaut sein soll. Darauf aufbauend wird ein konkretes Systemkonzept entworfen (Kapitel 4.2).

4.1 ANFORDERUNGEN AN DAS KONZEPT

Da sich BPMN sich quasi als Standard für die Modellierung von *Prozessmodellen* durchgesetzt hat [11], wird in dieser Arbeit BPMN als Modellierungssprache verwendet (siehe Anforderung 1).

Anforderung 1 (BPMN als Modellierungssprache):

Als Modellierungssprache soll BPMN verwendet werden.

BPMN definiert neben der Semantik der einzelnen Prozesselemente auch die Syntax der Prozessmodelle. Um vor allem Benutzer zu unterstützen, welche mit der Syntax nicht vertraut sind, können Syntaxprüfungen von dem System angeboten werden. Versucht ein Benutzer z.B. zwei Prozesselemente miteinander zu verbinden, überprüft das System ob eine Verbindung dieser Elemente syntaktisch Kor-

rekt ist und zeigt dies dem Benutzer an. Dadurch wird die syntaktische Qualität der Prozessmodelle verbessert.

Anforderung 2 (Syntaxprüfung von Prozessmodellen):

Um die syntaktische Qualität der Prozessmodelle zu steigern, werden Benutzer bei der Prozessmodellierung durch Syntaxprüfungen unterstützt. Prozesselemente können nur verbunden werden, wenn dabei die korrekte Syntax eingehalten wird.

In dieser Arbeit wird ein Konzept für ein einfach zu bedienendes System erstellt, welches eine „natürliche“ Bedienung erlaubt und durch ein *Natural User Interface* (NUI) angeboten werden soll. Da der Begriff *NUI* überladen und verwirrend ist [33], wird in dieser Arbeit darunter ein Interface verstanden, welches sich an Metaphern der realen Welt bedient. Dies impliziert sowohl die Touch-Interaktion, als auch, dass jedes Interaktionselement *anfassbar*, *verschiebbar* und *rotierbar* ist (siehe Anforderung 3).

Anforderung 3 (Natural User Interface):

Die Interaktion muss sich Vorbildern aus der Realität bedienen.

Zentraler Bestandteil des Systems ist ein großes, interaktives Display, welches Touch-Interaktionen erlaubt. Der große Interaktionsraum unterstützt hierbei kollaboratives Arbeiten, da mehrere Benutzer gleichzeitig mit dem Display interagieren können [30]. Ein vertikal installiertes Display, welches kollaborativ verwendet wird, wird oft als unangenehm empfunden [34]. Wird das Display horizontal aufgebaut und als Tabletop-System verwendet, unterstützt dies die Benutzer, mehr Ideen zu erforschen. Zusätzlich können sie besser nachvollziehen was andere Benutzer machen [34]. Daraus ergibt sich Anforderung 4.

Anforderung 4 (Tabletop-System):

Das interaktive Display, welches für kollaborative Arbeiten verwendet wird, wird horizontal installiert und als Tabletop-System verwendet.

Während der Interaktion mit Tabletop-Systemen, können Objekte auf diesem abgelegt werden. Diese können *aufgabenbezogen* oder *nicht aufgabenbezogen* sein. *Aufgabenbezogenen Objekte* können mobile Geräte oder auch spezielle Objekte sein, um mit dem Tabletop-System zu interagieren und Aktionen auszulösen [9, 30]. *Nicht aufgabenbezogene Objekte* müssen von *aufgabenbezogenen Objekten* unterschieden werden, damit Benutzer das Tabletop-System auch als Tisch verwenden können, um z.B. eine Tasse abzustellen (siehe Anforderung 5).

Anforderung 5 (Verwendung von physikalischen Objekten):

Aufgabenbezogene Objekte müssen von *nicht aufgabenbezogene Objekten* unterschieden werden.

Die meisten Aufgaben, welche auf kollaborativ genutzten Tabletop-Systemen bearbeitet werden, sind nur ein Teil der gesamten gemeinschaftlichen Arbeit. Das Tabletop-System muss daher den Benutzern einen Übergang hin zu individueller Arbeit bieten, welche abseits des Tabletop-Systems stattfinden [25]. Dieser Übergang soll den Benutzern so einfach wie möglich gemacht werden (siehe Anforderung 6).

Anforderung 6 (Übergang von kollaborativer, zu individueller Arbeit):

Ein produktiv genutztes System besteht nicht nur aus kollaborativer Arbeit am Tabletop-System. Ein einfacher Übergang vom Tabletop-System zur individuellen Arbeit muss angeboten werden.

Tabletop-Systeme sollen Möglichkeiten für *direkte Manipulationen* von visuellem Inhalt anbieten (siehe Anforderung 7) [31, 30]. Somit soll auf Hard- und Software-Buttons verzichtet werden, um z.B. ein Interaktionselement zu drehen oder zu vergrößern.

Anforderung 7 (Direktes Interface):

Der Inhalt ist das Interface. Buttons, Menüs oder Slider, um z.B. Objekte zu rotieren, skalieren oder zu verschieben, sollen nicht angeboten werden.

Tabletop-Systeme müssen und können *neue Eingabemöglichkeiten* anbieten, um die Nichtverwendung von Maus und Tastatur zu kompensieren (siehe Anforderung 8) [31].

Anforderung 8 (Neuartige Eingabemöglichkeiten):

Neue Eingabemöglichkeiten, welche die Nichtverwendung von Maus und Tastatur kompensieren, müssen angeboten werden.

Ein einheitliches Eingabegerät für sämtliche Benutzer-Aktionen auf dem Tabletop-System hilft, die Interaktion zu vereinfachen, da keine Geräte gewechselt werden müssen [25]. Auf der anderen Seite, kann ein spezialisiertes Eingabegerät die Bewältigung einer bestimmten Aufgabe stark vereinfachen (z.B. Tastatur statt On-Screen-Tastatur). Um Anforderung 6 nachzukommen und den Übergang von kollaborativer, zu individueller Arbeit einfach und intuitiv zu gestalten, werden Tablets als private Displays verwendet (siehe Anforderung 8) und der Datentransfer mittels *Tablet-Touch-Aktionen* (siehe Kapitel 3.3.3) angeboten. Dies wird in Anforderung 9 zusammengefasst.

Anforderung 9 (Tablets als private Displays):

Neben dem Tabletop-System für kollaborative Arbeit sollen Tablets für individuelle Arbeit angeboten werden.

Die fundamentalste Art der Zusammenarbeit ist, dass Benutzer *Daten* und *Objekte* miteinander teilen. Auch wird das Teilen als Start einer Kommunikation zwischen Benutzern verwendet [29]. Bei Systemen, welche zusätzlich zu dem Tabletop-System ein separates Gerät mit privatem Display anbieten, kann die Kommunikation zwischen den Benutzern erschwert werden. Dies begründet sich darin, dass

die Blickrichtung und Zeigegesten der Benutzer von anderen Benutzern nicht interpretiert werden können, wenn sich dieser auf das private Display bezieht [6, 35]. Somit muss das Teilen von Daten auch zwischen den privaten Geräten und dem zentralen Tabletop-System unterstützt werden (siehe Anforderung 10).

Anforderung 10 (Unterstützung der Kommunikation):

Um die Kommunikation der Benutzer zu unterstützen, müssen Daten und Objekte zwischen allen Geräten ausgetauscht werden können.

Eine der natürlichsten Arten der Kommunikation zwischen Benutzern ist, auf Objekte zu *zeigen* oder auf diese zu *schauen*. Die Blickrichtung eines Benutzers wird von anderen intuitiv wahrgenommen. Diese Geste kann jedoch nur verwendet werden, wenn Objekte als *Unikat* dargestellt werden. Hat jeder Benutzer seine eigene Kopie des gleichen Objekts, fällt ein wichtiger Bestandteil der Kommunikation weg [36] (siehe Anforderung 11).

Anforderung 11 (Keine Duplikate):

Um die Kommunikation der Benutzer zu vereinfachen, sollen keine Duplikate erzeugt werden. Dadurch beziehen sich Benutzer immer auf dasselbe Objekt.

Der Übergang der Benutzung von einem persönlichen Gerät zum Tabletop-System muss dem Benutzer möglichst einfach gemacht werden. Dies kann unterstützt werden, indem Geräten ein ähnliches Nutzungsmodell haben [35]. Das bedeutet auch, dass ein einheitliches *Look & Feel* gegeben sein muss, um den mentalen Kontextwechsel der Benutzer gering zu halten. Dabei können, je nach Plattform auf welcher die einzelnen Programme ausgeführt werden, die Design- und Interaction-Design -Guidelines der einzelnen Plattformen in Konflikt zueinander stehen. Anforderung 12 beschreibt das einheitliche Konzept des Gesamtsystems.

Anforderung 12 (Übergänge von persönlicher zur Gruppenarbeit):

Der *mentale Kontextwechsel* der Benutzer muss gering gehalten werden. Unterschiedliche Geräte müssen ein einheitliches Erscheinungsbild und Bedienkonzept anbieten.

Die Art der Aufgabe hat Einfluss auf die bevorzugte Anordnung der Benutzer an dem Tabletop-System. Z.B. stehen bei eng gekoppelten Aufgaben Benutzer näher zusammen [30]. Bei Aufgaben mit hoher Kommunikation stehen sie sich lieber gegenüber. Folglich müssen unterschiedliche Anordnungen der Benutzer unterstützt werden.

Anforderung 13 (Anordnung von Benutzern):

Abhängig der Aufgabe, der Anzahl an Benutzern und der individuellen Eigenschaften der Benutzer, variiert die Anordnung der Benutzer an einem Tabletop-System. Das System muss flexibel darauf reagieren und darf eine freie Anordnung der Benutzer nicht einschränken.

Desktop-Systeme unterstützen keine Multi-User-Eingaben. Hingegen interagieren Benutzer von Tabletop-Systemen oft gleichzeitig mit dem System. So müssen simultane Benutzeraktionen von der Hard- und Software unterstützt werden (siehe Anforderung 14) [30].

Anforderung 14 (Simultane Benutzeraktionen):

Das System muss ständig die Möglichkeit bieten, dass mehrere Benutzer gleichzeitig an dem System arbeiten können.

Die zeitgleiche Benutzung des Systems auf einem Gerät bringt, im Vergleich zu einem Single-User-System, neue Probleme mit sich. So ist es nicht möglich, eine *Rückgängig-Funktion* anzubieten. Den Grund hierfür zeigt folgendes Beispiel auf:

Zwei Benutzer arbeiten kollaborativ an einem System. Benutzer A führt eine destruktive Aktion (z.B. Lösch-Aktion) aus. Er überlegt kurz, bis er zu dem Schluss kommt, die Aktion rückgängig machen zu wollen. In der Zwischenzeit hat Benutzer B einige konstruktive Aktionen getätigt (z.B. neues Element erstellen). Würde Benutzer A die Rückgängig-Funktion verwenden, würde er die Arbeit von Benutzer B zurücksetzen.

Da einzelne Aktionen nicht einem bestimmten Benutzer zugeordnet werden können, darf keine Rückgängig-Funktion angeboten werden. Ein weiteres Problem ergibt sich bei der Verwendung eines Multi-User-Systems, wenn die oben genannten Benutzer zusammenarbeiten und Benutzer A eine Zoom-Aktion ausführt, um z.B. seinen Arbeitsbereich zu vergrößern. Wenn Benutzer B an einer anderen Stelle arbeitet, kann es vorkommen, dass sein Bereich durch die Vergrößerung nicht mehr sichtbar ist.

Somit kann zusammengefasst werden, dass globale Aktionen nicht unterstützt werden können (siehe Anforderung 15) [30].

Anforderung 15 (Keine globalen Aktionen):

Aktionen eines Benutzers dürfen andere Benutzer in ihrer Arbeit nicht einschränken. Folglich werden *globale Aktionen* nicht unterstützt.

Wird zum Beispiel ein Modus für Skizzeneingabe und ein Modus für die Ausrichtung der Interaktionselemente angeboten, würde das einer globalen Aktion gleichkommen. Dies ist ein Widerspruch zu Anforderung 15. Daraus folgt, dass auf Modi verzichtet werden soll. Dadurch ergibt sich auch ein flüssiges Interaktionskonzept (siehe Anforderung 16).

Anforderung 16 (Flüssige Übergänge und keine Modi):

Um unnötige Kontextwechsel zu vermeiden, sollen alle Funktionen auf allen Geräten (Tabletop-System und mobile Geräte) angeboten werden. Verschiedenen Eingabemodi sollen vermieden werden.

Anforderung 3 und Anforderung 7 fordern ein natürliches und direktes User-Interface. Um diesen Anforderungen nachzukommen, soll eine skizzenbasierte Benutzereingabe angeboten werden (siehe Kapitel 3.4). Jedoch müssen hierfür folgende Kriterien erfüllt werden [21]:

- K1: Direkte und flüssige Touch-Interaktion
- K2: Unterstützung für kollaborative Arbeit
- K3: Erstellen von formalen und informellen Elementen
- K4: Unterstützung für große Modelle
- K5: Integration in bereits existierende Systeme

Kriterium K1 wird bereits durch Anforderung 7 repräsentiert, Kriterium K2 von Anforderung 14 und Kriterium K5 wird durch Anforderung 12 wiedergegeben. Durch die in Kapitel 3.1 vorgestellten Prozesselemente der BPMN werden formale und informelle Elemente bereitgestellt (Kriterium K3). Das Kriterium K4 wird durch Anforderung 17 repräsentiert.

Anforderung 17 (Unterstützung für große Modelle):

Ein großer Interaktionsbereich muss angeboten werden, um folglich auch große Modelle zu unterstützen. Zusätzlich soll die Möglichkeit gegeben werden, einzelne Prozelelemente in der Größe zu ändern.

Die Ausrichtung der Interaktionselemente nimmt bei Tabletop-Systemen eine Schlüsselrolle ein (siehe Kapitel 3.5). Eine halbautomatische Ausrichtung steht im Widerspruch zu Anforderung 13, da das Tabletop-System flexible Benutzeranordnungen unterstützen soll. Die Verwendung der Magnet-Metapher kommt einer globalen Aktion gleich und ist aufgrund von Anforderung 15 somit unzulässig. Daher wird eine manuelle Ausrichtung der Interaktionselemente verwendet (siehe Anforderung 18).

Anforderung 18 (manuelle Ausrichtung):

Die Ausrichtung der Interaktionselemente soll durch das Tabletop-System nicht beeinflusst werden. Interaktionselemente werden somit von den Benutzern manuell ausgerichtet.

Berührt ein Benutzer das Display des Tabletop-Systems, sollen immer Rückmeldungen auf eine Touch-Eingabe gegeben werden, auch wenn keine Aktion ausgeführt wird [30]. Dadurch wird dem Benutzer signalisiert, dass die Touch-Eingabe erkannt wurde. Das visuelle Feedback soll dabei so gestaltet sein, dass Benutzer beim Erlernen des Systems unterstützt werden (siehe Anforderung 19).

Anforderung 19 (Immer auf Benutzereingaben reagieren):

Das System muss immer auf Touch-Eingaben reagieren um dem Benutzer Feedback zu geben und den aktuellen Systemstatus anzuzeigen.

In einem real anmutenden System kann ein Interaktionselement z.B. nicht plötzlich verschwinden, wenn es gelöscht wird. Der Übergang muss dabei fließend sein, um dem Benutzer zu zeigen, was gerade passiert [30] (siehe Anforderung 20).

Anforderung 20 (Animierte Zustandsübergänge):

Jeder Zustandsübergang muss animiert sein, um den Benutzer diesen zu verdeutlichen.

Die aufgeführten Anforderungen bilden die Basis für ein kollaboratives Prozessmodellierungssystem auf einem Tabletop. Diese werden im nächsten Kapitel verwendet um Interaktionskonzepte zu definieren und ein Gesamtkonzept für die kollaborative Prozessmodellierung auf Tabletop-Systemen zu erstellen.

4.2 GESAMTKONZEPT

Um die kollaborative Prozessmodellierung zu verbessern, wird ein Tabletop-System verwendet, damit Benutzer gemeinschaftlich auf einem Gerät arbeiten können. Tabletop-Systeme weisen spezifische Stärken (z.B. Kollaboration) und Schwächen (z.B. Orientierung der Elemente) auf. Um die Schwächen des Tabletop-Systems zu kompensieren und Benutzern die Möglichkeit zu geben, individuelle zu arbeiten, wird verstärkt auf die zusätzliche Verwendung von mobilen Geräten, wie Smartphones und Tablets, gesetzt.

So kann jedes Gerät als Spezialist für einzelne Teilbereiche der kollaborativen Geschäftsmodellierung angesehen werden. Durch eine sinnvolle Kombination unterschiedlicher Geräte zu einem Gesamtsystem, sollen die Vorteile gestärkt und die Nachteile kompensiert werden, d.h. die Handhabung von Prozessmodellen geschieht auf dem Tabletop oder auch auf dem Smartphone oder Tablet (siehe Anforderung 9).

Die in Kapitel 4.1 bestimmten Anforderungen werden nachstehend zusammengefasst, um das System *Process-Touch* aufzubauen.

Mobile Geräte werden beim *Erstellen*, *Bearbeiten* und *Betrachten* der Prozessmodelle miteinbezogen. Dabei können sie jeweils ihre Stärke als individuelles Gerät mit Touchscreen ausspielen. Schnittstellen, an denen eine Kommunikation zwischen den Geräten und dem Tabletop nötig ist, müssen intuitiv gestaltet werden (siehe Anforderung 6).

Teile eines Prozessmodells können auf unterschiedlichen Geräten erstellt werden, um sie anschließend auf dem zentralen Tabletop-System zu verbinden (siehe Anforderung 10, Anforderung 4 und Abbildung 4-1). Dabei sollen Prozesselemente zum einen vom mobilen Gerät hin zum Tabletop-System übertragen werden, als auch vom Tabletop-System zum mobilen Gerät.

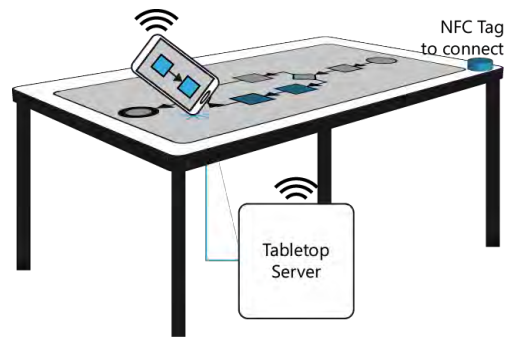


Abbildung 4-1: Tabletop als zentrale Instanz

In Abbildung 4-1 ist der Aufbau dargestellt. Der Tabletop dient als Server, der das Prozessmodell enthält und mit den unterschiedlichen Geräten kommuniziert. Der NFC-Tag dient dazu, die Verbindung zwischen Smartphone und Tabletop-Server benutzerfreundlich zu initiieren. Hierbei muss der Benutzer den NFC-Tag mit dem mobilen Gerät berühren (siehe Anforderung 5). Weiter wird in Abbildung 4-1 gezeigt, dass sich Teile des gesamten Prozessmodells, welche auf den mobilen Geräten erstellt wurden, mit einem einfachen Tap auf die Oberfläche des Tabletops an diese Stelle einfügen lassen (siehe Kapitel 3.3.3, Anforderung 5 und Anforderung 7). Durch die Positionierung der eingefügten Prozesselemente um die Position des mobilen Geräts herum, sind diese auf dem Tabletop-System sofort auffindbar und der mentale Kontextwechsel wird erleichtert (siehe Anforderung 12).

Wird ein Prozessmodell auf das Tabletop-System übertragen, erfolgt die Verbindung der einzelnen Teile des Prozessmodells anschließend mittels Touchbedienung auf dem Tabletop-System.

Um Teilsysteme auf den mobilen Geräten zu bearbeiten, können die Prozesselemente mit diesem eingekreist und anschließend darauf übertragen werden. Dies funktioniert auch mit mehreren Personen gleichzeitig, da eine Touch-Aktion (wie in Abschnitt 3.3.3 gezeigt) einem Benutzer eindeutig zugeordnet werden kann (siehe Anforderung 14). Zudem ist es auch möglich, ganze Prozessmodelle nur mithilfe des Tabletop-Systems zu erstellen. Die Erstellung von Prozesselementen und deren Verbindung ist sowohl auf dem Tabletop-System, als auch auf mobilen Geräten möglich. Dabei unterscheidet sich die Art und Weise der Interaktion nicht (siehe Anforderung 12 und Anforderung 16).

Um mobilen Geräten ohne NFC ebenfalls eine komfortable Verbindung zum Server zu ermöglichen, wird zusätzlich zu dem NFC-Tag ein QR-Code angeboten.

Das Konzept von Process Touch kann auch ohne Tabletop-System umgesetzt werden. Dazu zeigt Kapitel 4.2.1 einen alternativen Aufbau auf. In Kapitel 4.2.2 werden die unterschiedlichen Möglichkeiten der Erstellung und in Kapitel 4.2.3 die unterschiedlichen Möglichkeiten der Bearbeitung von Prozessmodellen aufgezeigt. Kapitel 4.2.4 erläutert, wie mit bereits gedruckten Prozessmodellen interagiert werden kann.

4.2.1 ALTERNATIVER AUFBAU

Durch die geringe Verbreitung von Tabletop-Systemen, zeigt dieses Kapitel einen alternativen Systemaufbau. Der alternative Systemaufbau kann jedoch auch als Erweiterung des bereits bestehenden Systemaufbaus verwendet werden.

Das Konzept ist auch ohne Tabletop-System möglich und wird in Abbildung 4-2 gezeigt [37]. So dient ein Desktop-PC mit einem präparierten Monitor als zentrale Instanz des Systems und ersetzt somit das Tabletop-System. Mithilfe des präparierten Monitors wird die Kommunikation zwischen mobilen Geräten und dem Desktop-PC stark vereinfacht. Wird z.B. ein Smartphone an den Rand des Monitors gehalten, erscheint das Prozessmodell, welches auf dem mobilen Gerät erstellt wurde, an dieser Stelle. Anschließend lässt sich das Prozessmodell mit einer Drag&Drop-Aktion verschieben und mittels Mausinteraktionen mit dem Gesamtsystem verbinden.

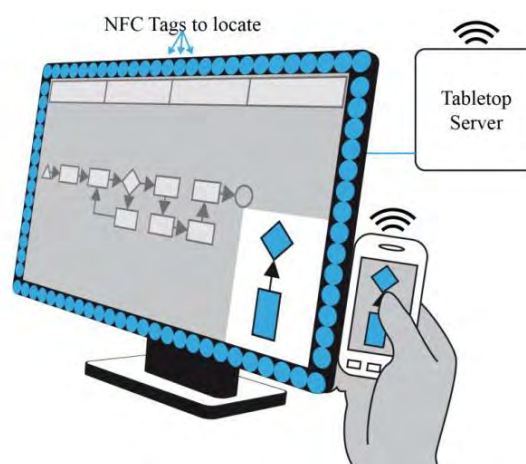


Abbildung 4-2: Alternativer Systemaufbau

Ebenso lässt sich ein zuvor selektierter Teil des Prozessmodells des Desktop-Pcs auf die eingeblendete Fläche ziehen, um sie dem mobilen Gerät zur weiteren Bearbeitung zur Verfügung zu stellen. Die eingeblendete Fläche (siehe Abbildung 4-2, weiße Fläche) kann als gemeinsamer Bereich betrachtet werden. Der Datenaustausch zwischen Server und dem mobilen Gerät wird hiermit ermöglicht. Werden zwei Geräte gleichzeitig an den Monitor gehalten, erscheint für jedes ein eigener Bereich. Damit lassen sich auch Daten zwischen zwei mobilen Geräten austauschen (siehe Anforderung 10). Im Hintergrund findet der Datentransfer über WLAN oder Bluetooth statt. So kann, wie in Kapitel 3.3.2 beschrieben, auch ein präparierter Monitor als zentrale Instanz dienen (siehe Anforderung 8). Der Nachteil des alternativen Aufbaus besteht darin, dass durch die unterschiedliche Bedienung (Touch-Interaktion vs. Maus- und Tastatur-Interaktion) kein flüssiger Übergang realisierbar ist (siehe Anforderung 16). Der Vorteil liegt jedoch darin, dass durch den einfachen Austausch ein Übergang zur individuellen Arbeit gegeben ist (siehe Anforderung 6) und durch die Verwendung eines Desktop-PCs, dessen Vorzüge ausgespielt werden können. Wird an einem Prozessmodell länger gearbeitet, kann eine bequeme Sitzposition eingenommen, Text kann schneller eingegeben und durch die Verwendung der Maus kann präziser interagiert werden.

4.2.2 ERSTELLUNG VON PROZESSMODELLEN

Auf mobilen Geräten, als auch auf dem Tabletop-System, können Prozesselemente mittels *Menüs* oder *Skizzeneingabe* erstellt werden. Die Skizzeneingabe wird dabei auf unterschiedliche Arten angeboten (siehe Abbildung 4-3).

Prozesselemente können erstellt werden, indem Benutzer den Umriss des gewünschten Prozesselements mit dem Finger skizzieren (siehe Anforderung 8 und Abbildung 4-3-b). Nachdem der Finger von der Oberfläche gelöst wird, interpretiert das System die Skizzeneingabe. Durch die sofortige Interpretation der Skizzeneingabe registriert der Benutzer falsche Interpretationen seitens des Systems sofort.

Die zweite Möglichkeit der Skizzeneingabe basiert darauf, Prozessmodelle auf einem Whiteboard oder Papier zu. Wird das skizzierte Prozessmodell vom Benut-

zer fotografiert und anschließend vom System automatisch in ein Prozessmodell übersetzt (siehe Anforderung 8), kann er dieses anschließend in das Gesamtsystem einbinden (siehe Abbildung 4-3-a).

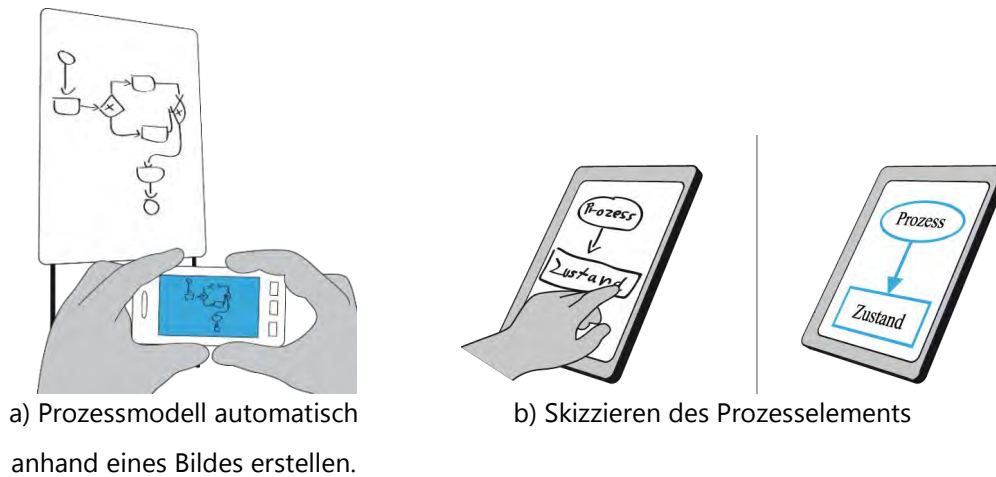


Abbildung 4-3: Skizzenbasierte Eingabe

4.2.3 BEARBEITUNG VON PROZESSMODELLEN

Prozessmodelle können auf dem Tabletop-System sowie auf mobilen Geräten bearbeitet werden. Um Teile des Prozessmodells auf mobilen Geräten zu bearbeiten, müssen Prozesselemente zuerst vom Benutzer ausgewählt werden. Hierfür selektiert der Benutzer die gewünschten Prozesselemente mittels einer Lasso-Geste, welche mit dem mobilen Gerät ausgeführt wird (siehe Abbildung 4-4). Anschließend werden diese auf das mobile Gerät übertragen (siehe Anforderung 6).

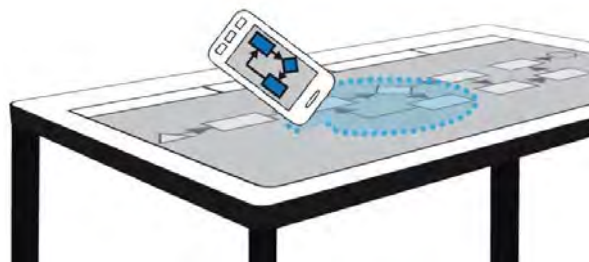


Abbildung 4-4: Auswahl der Prozesselemente

4.2.4 INTERAKTION MIT GEDRUCKTEN PROZESSMODELLEN

Werden Prozessmodelle ausgedruckt, können diese normalerweise nur betrachtet werden. Dieses Kapitel zeigt die Möglichkeit auf, wie mit gedruckten Modellen

interagiert werden kann um z.B. Prozesselemente umzubenennen oder zu löschen (siehe Anforderung 5). Somit können mobile Geräte auch gedruckte Prozessmodelle verändern (siehe Anforderung 8).

Den Startpunkt stellt das gedruckte Prozessmodell dar. Dieses leidet meist unter dem Problem, dass nicht alle Informationen, Attribute und Beziehungen mitgedruckt werden können, um die Komplexität zu reduzieren und die Verständlichkeit zu maximieren (siehe Anforderung 17). Weiter können z.B. Änderungen der Beschriftung von Prozesselementen nicht erfasst werden, wenn das Prozessmodell bereits gedruckt ist.

Um gedruckte Prozessmodelle zu verändern und geänderte Werte eines bereits gedruckten Prozessmodells anzeigen zu können, lassen sich verschiedene Prinzipien anwenden. Diese werden in den folgenden Kapiteln aufgezeigt.

4.2.4.1 Präparierte Prozessmodelle

Ausgedruckte Prozessmodelle können mit NFC-Tags präpariert werden [38]. Dadurch wird jedes Prozesselement mit einem eigenen NFC-Tag eindeutig markiert. Bei einer Berührung des NFC-Tags mit dem mobilen Gerät, wird die vom NFC-Tag gespeicherte ID drahtlos zum Server übermittelt. Anschließend stellt der Server die zusätzlichen Informationen dem Benutzer über das mobile Gerät zur Verfügung (siehe Abbildung 4-5).

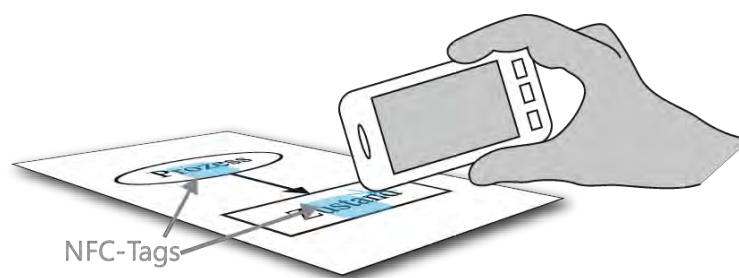


Abbildung 4-5: Gedrucktes und mit NFC-Tags präpariertes Prozessmodell

4.2.4.2 Mobile Augmented Reality

Um zusätzliche Informationen zu gedruckten Prozessmodellen zu erhalten, kann das *Augmented Reality Pattern* [39, 40] verwendet werden. Hierbei verwendet der Benutzer ein Smartphone und richtet die Kamera auf das Prozessmodell. Das Bild des Prozessmodells ist auf dem Display mit zusätzlichen Interaktionselementen

zu sehen. So können für jedes Prozesselement zusätzliche Informationen oder ein Interaktionselement dargestellt werden (siehe Abbildung 4-6).

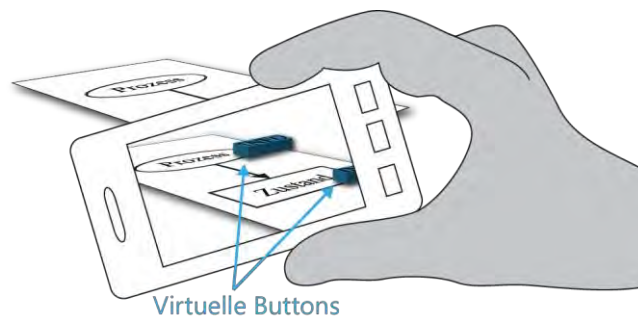


Abbildung 4-6: Mobile Augmented Reality

4.2.4.3 Projizierte Information

Smartphones wie z.B. das Samsung Galaxy Beam besitzen einen integrierten Projektor [41]. Damit ist es möglich zusätzliche Informationen direkt auf das gedruckte Prozessmodell projizieren (siehe Abbildung 4-7).

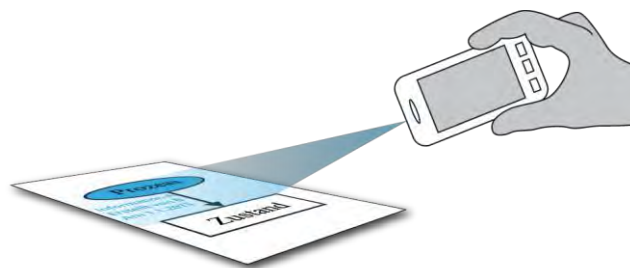


Abbildung 4-7: Projizierte Information

Der Vorteil besteht darin, dass kein *Kontextwechsel* zwischen dem gedruckten Prozessmodell und dem Smartphone-Display nötig ist. Die projizierten Informationen sind für alle Benutzer sichtbar, was je nach Anwendungsfall ein Vor- aber auch ein Nachteil sein kann.

4.3 ZUSAMMENFASSUNG

Dieses Kapitel zeigt auf, welche Möglichkeiten bestehen, mithilfe mobiler Geräte, Prozessmodelle zu *erstellen*, zu *bearbeiten* und die *Betrachtung* von Prozessmodellen mittels mobiler Geräte zu *erweitern*. Das Gesamtkonzept von Process-Touch wird beschrieben, welches mobile Geräte als individuelle Eingabemedien verwendet.

Das hier vorgestellte Konzept unterstützt den Benutzer, Prozessmodelle zu erstellen, zu betrachten und zu erweitern. Aufgrund der hohen Komplexität, ist es im Rahmen dieser Arbeit nicht möglich, das Konzept vollständig umzusetzen. So wird in der weiteren Arbeit nur die Erstellung von Prozessmodellen, mithilfe eines Tabletop-Systems und mobilen Geräten, betrachtet und ausgearbeitet.

5 INTERACTION-DESIGN

In diesem Kapitel wird ein Teil des in Kapitel 4 erstellten Konzepts, wird in diesem Kapitel ausgearbeitet und verfeinert. Dabei wird der Fokus auf das zentrale Tabletop-System und die Interaktion mit mobilen Geräten gelegt. Im Detail wird auf die allgemeine und menübasierte Interaktion (siehe Kapitel 5.1), ein Gestenset für die skizzenbasierte Interaktion erstellt (siehe Kapitel 5.2) und die Interaktion mit mobilen Geräten aufgezeigt (siehe Kapitel 5.3). Die Regeln der Syntaxprüfung werden erläutert (siehe Kapitel 5.4) und ein Layouting-Verfahren für Prozessmodelle erarbeitet (siehe Kapitel 5.5)

5.1 ALLGEMEINE INTERAKTION

Dieser Abschnitt zeigt den generellen Aufbau des Systems Process-Touch mittels Touch-Eingaben. Touch-Eingaben können sowohl auf dem Tabletop-System, als auch auf mobilen Geräten getätigt werden. Die Entscheidung wird dabei dem Benutzer überlassen. Um Anfängern und auch fortgeschrittenen Benutzern eine adäquate Bedienung anzubieten, greift diese Arbeit den Ansatz eines *Interaction-Rich-Systems* [33] auf, welcher eine Vielzahl an verschiedenen Touch-Eingaben und *Gesten* vorsieht. Eine Geste beschreibt dabei eine Touch-Eingabe, welche aus einer Kombination aus *Berührung* und *Bewegung* eines oder mehrerer Finger besteht.

In einem Interaction-Rich-System kann eine Aktion von verschiedenen Gesten ausgelöst werden, jedoch kann auch eine Geste in unterschiedlichen Kontexten, unterschiedliche Aktionen auslösen. Eine *Pinch*-Geste, wobei zwei Finger einer Hand zueinander, bzw. voneinander weg bewegt werden, kann beispielsweise zum Zoomen oder auch zum Öffnen einer Gruppe von verwendet werden [32] (siehe Kapitel 5.2).

Neben Gesten, sollen Benutzern zusätzlich Menüs angeboten werden, um Prozesselemente zu erstellen und mit diesen zu interagieren. So gibt es zwei verschiedene Menüs. Mithilfe des *Hauptmenüs* können neue Prozesselemente erstellt werden und bietet somit einen Einstiegspunkt für die Erstellung von Prozessmodellen. Das Menü wird mittels einer Pinch-Geste auf dem Hintergrund geöffnet

(siehe Abbildung 5-1). Da das Tabletop-System von Benutzern auch als Ablagefläche verwendet wird und Benutzer sich darauf aufstützen, muss eine Geste als Einstiegspunkt verwendet werden, welche eine versehentliche Öffnung des Menüs vermeidet [30] (siehe Anforderung 5).

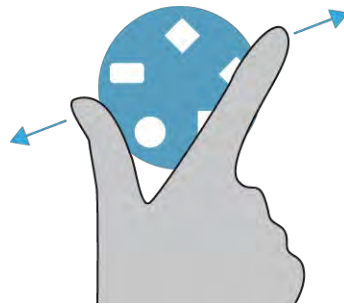


Abbildung 5-1: Hauptmenü öffnen

Prozesselemente können durch *Kanten* miteinander verbunden werden. Benutzer versuchen häufig mit diesen Kanten zu interagieren [4]. Um den Ansatz des *Interaction-Rich-Systems* fortzuführen, soll das System eine Interaktion mit diesen ermöglichen. Kanten werden folglich mit einem Kreis versehen (siehe Abbildung 5-2-a), der zum einen als Einstiegspunkt für die menübasierte Erstellung von Prozesselementen dient (siehe Abbildung 5-2-b), zum anderen als Indikator, um an diese Stelle Prozesselemente mittels Drag&Drop einzufügen (siehe Abbildung 5-2-c). So können auch unerfahrene Benutzer einfach mit dem System arbeiten, ohne die Gesten für die Erstellung von Prozesselemente zu kennen.



Abbildung 5-2: Arbeiten mit Verbindungspfeilen

Neben dem Hauptmenü soll für Prozesselemente ein *Kontextmenü* angeboten werden. Dieses bietet Benutzern die Möglichkeit, Eigenschaften zu ändern und Aktionen auf dem Prozesselement auszuführen. Die Aktionen, welche vom Kontextmenü angeboten werden, sind in Tabelle 5-1 aufgeführt.

Aktion	Beschreibung
<i>Subprozess entpacken</i>	Die Aktion <i>Subprozess entpacken</i> ist nur bei Subprozessen verfügbar und <i>entpackt</i> (engl.: <i>Unwrap</i>) alle Prozesselemente, welche sich innerhalb des Subprozesses befinden.
<i>Subprozess erstellen</i>	Die Aktion <i>Subprozess erstellen</i> ist nur bei Aktivitäten verfügbar und erstellt aus einer Aktivität einen <i>leeren Subprozess</i> .
<i>Prozesselement duplizieren</i>	Das Prozesselement, welches das Kontextmenü öffnet, wird <i>dupliziert</i> .
<i>Prozesselement löschen</i>	Das Prozesselement, welches das Kontextmenü öffnet, wird <i>gelöscht</i> .
<i>Beschriftung ändern</i>	Eine virtuelle Tastatur erscheint neben dem Prozesselement, welches das Kontextmenü öffnet, um die Beschriftung zu ändern.

Tabelle 5-1: Aktionen des Kontextmenüs

Die Beschriftung des Prozesselements soll zusätzlich über eine Doppel-Tap-Geste geändert werden können, um für eine häufig verwendete Aufgabe eine Abkürzung anzubieten. Da es zu Problemen kommen kann, wenn nur eine einzige Tastatur angezeigt werden kann [5], muss für jede Texteingabe eine separate Tastatur bereitstehen. Dadurch können mehreren Benutzern gleichzeitig Texteingaben machen (siehe Anforderung 14). Tastaturen werden in unmittelbarer Nähe zum Prozesselement positioniert (vgl. *Gesetz der Nähe*) Abbildung 5-3 zeigt die Tastatur, welche an einem Prozesselement ausgerichtet ist.

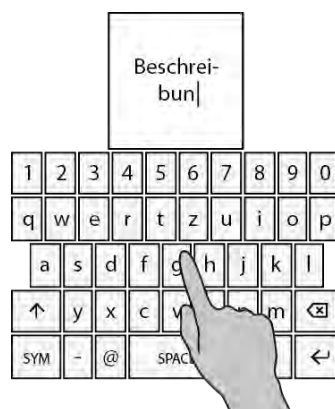


Abbildung 5-3: Virtuelle Tastatur

Jedes Menü und Prozesselement kann von Benutzern *vergrößert*, *verkleinert*, *verschoben* und *rotiert* werden. Wird eine Tastatur angezeigt, ist diese mit dem korrespondierenden Prozesselement verbunden. Das bedeutet, dass eine Verschiebung oder Rotation der Tastatur eine entsprechende Aktion des Prozesselements auslöst, um die relative Ausrichtung der beiden Elemente zueinander konstant zu halten (siehe Anforderung 7). Dadurch wird das Prozesselement mit der Tastatur in einer Einheit zusammengefasst um unnötige Kontextwechsel zu vermeiden (siehe Anforderung 16).

5.2 GESTEN-SET FÜR BENUTZEREINGABEN

Neben der menübasierten Eingabe, sollen dem Benutzer auch *Gesten* angeboten werden, um mit dem Inhalt zu interagieren. Im Detail sind die möglichen Gesten die elementaren Bausteine des Interaction-Designs. Somit ist die Erstellung und Auswahl einzelner Gesten und die Zusammenstellung eines kompletten Gestensets fundamental für das Interaction-Design.

In der Literatur sind zahlreiche Gestensets für unterschiedliche Kontexte (u.A. auch BPM) zu finden [42, 21, 43, 4, 32, 30]. Einige Gesten können durch die verwendete Hardware oder in einem Multi-User-System nicht umgesetzt werden. Durch die teils geringe Probandenzahl der Studien, mit denen die Gesten-Sets erstellt und validiert werden, werden diese als Indiz verstanden und es wird für dieser Arbeit eine Kombination der vorgeschlagenen Gestensets verwendet. Generell achten Benutzer kaum auf die Anzahl der verwendeten Finger für eine Geste [32]. Hingegen bevorzugen sie *Einhandgesten* im Vergleich zu *Zweihandgesten*. Viele Gesten-Sets sind von den Systemherstellern vorgegeben und die Gesten nach verschiedenen Prinzipien logisch zusammengestellt [32, 43]. Das Benutzerverhalten ist jedoch selten systematisch.

Die Kombination von menübasierter und skizzenbasierter Eingabe erscheint als wichtig, wenn beide Ansätze möglich sind. Dies gilt speziell für die *Erstellung* und *Löschung* von Elementen. Die Verwendung von Gesten ändert sich jedoch nach ca. einer Woche, wenn das System anhaltend verwendet wird [33]. Benutzer bevorzugen nach der Eingewöhnungsphase öfter *Zweihandgesten*. Daraus folgt,

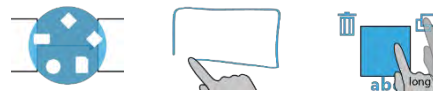
dass verschiedene Gesten im Sinne des *Interaction-Rich-Systems*, angeboten werden müssen.

Es kann auch Vorteile haben, eine Kombination aus Stift und Fingereingabe zu verwenden [42, 44]. Dabei werden mit der *nichtdominanten Hand* (NDH) Touch-Aktionen und mit der *dominanten Hand* (DH) Stiftaktionen ausgeführt. Der Vorteil liegt darin, dass Stiftaktionen präziser als Touch-Aktionen sind. So bezeichnen Benutzer die Interaktion mit zwei Händen, wobei die DH einen Stift verwendet, als *Einfach, Schnell* und *Natürlich* [44]. Entscheidend ist hierbei, dass die DH präzise und komplexe Aufgaben bearbeitet, und die NDH dabei den Kontext erschließt [45, 44].

Beispielsweise können Zweihandgesten auf dem Tabletop-System nicht verwendet werden, da die beiden Hände eines Benutzers nicht als zusammengehörig erkannt werden können, wenn mehrere Benutzer gleichzeitig am Tisch arbeiten. Das erstellte Gesten-Set beinhaltet auch die skizzenbasierte Erstellung (siehe Kapitel 3.4) von Prozesselementen und ist im Folgenden aufgeführt.

Erstellen einer Aktivität

Aktivitäten können mit dem Hauptmenü (Abbildung 5-4-a), der Skizzeneingabe (siehe Abbildung 5-4-b) oder mittels einer Kopie einer vorhandenen Aktivität (siehe Abbildung 5-4-c) erstellt werden.



- a) Menü-
basiert b) Skizzen-
basiert c) Kopie

Abbildung 5-4: Aktivität erstellen

Erstellen eines Gateways

Gateways können mit dem Hauptmenü (siehe Abbildung 5-5-a), der Skizzeneingabe (siehe Abbildung 5-5-b) oder einer Kopie eines vorhandenen Gateways (siehe Abbildung 5-5-c) erstellt werden.

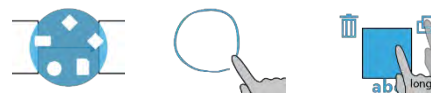


- a) Menü-
basiert b) Skizzen-
basiert c) Kopie

Abbildung 5-5: Gateway erstellen

Erstellen eines Ereignisses

Ereignisse können mit dem Hauptmenü (siehe Abbildung 5-6-a), der Skizzeneingabe (siehe Abbildung 5-6-b) oder einer Kopie eines vorhandenen Ereignisses (siehe Abbildung 5-6-c) erstellt werden.



a) Menü-basiert b) Skizzen-basiert c) Kopie

Abbildung 5-6: Ereignis erstellen

Prozesselemente verschieben

Elemente können mit einem Finger verschoben werden (siehe Abbildung 5-7).



Abbildung 5-7: Prozesselemente verschieben

Kante erstellen

Wird ein Prozesselement mit einem Finger berührt (A) und auf ein anderes Prozesselement (B) geschoben, wird eine neue Kante (Sequenzfluss) erstellt. Prozesselement A nimmt daraufhin die ursprüngliche Position auf dem Display wieder ein (siehe Abbildung 5-8).

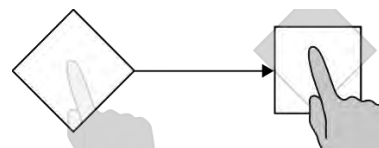


Abbildung 5-8: Elemente verbinden

Selektieren von Prozesselementen

Wird ein Prozesselement mit einem Finger kurz berührt und wieder losgelassen (siehe Abbildung 5-9-a) oder mit einem Finger eingekreist (siehe Abbildung 5-9-b), wird dieses selektiert. Werden mehrere Prozesselemente mit einem Finger eingekreist (siehe Abbildung 5-9-c), werden sie als Gruppe selektiert.

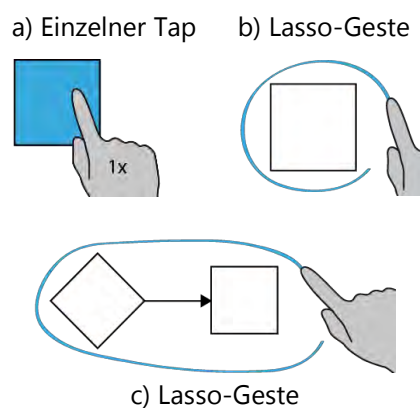


Abbildung 5-9: Selektierung von Prozesselementen

Deselektion von Prozesselementen

Wird in der Nähe der selektierten Prozesselemente eine Tap-Geste ausgeführt, wird die komplette Gruppe von Prozesselementen deselektiert (siehe Abbildung 5-10).

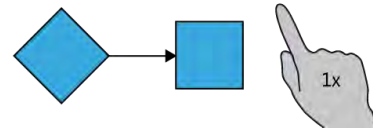


Abbildung 5-10: Tap auf Hintergrund

Prozesselemente zoomen

Prozesselemente können vergrößert werden, indem zwei Finger ein Prozesselement berühren und sich dann voneinander entfernen. Berühren beide Finger ein Prozesselement und nähern sich einander an, wird das Prozesselement verkleinert (siehe Abbildung 5-11).

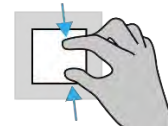
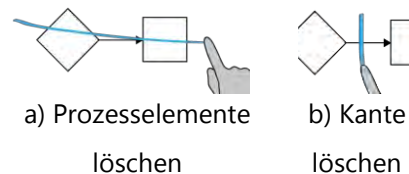


Abbildung 5-11: Zoomen

Löschen von Prozesselementen und Kanten

Wird der Hintergrund mit einem Finger berührt und ein Strich durch bestehende Prozesselemente (siehe Abbildung 5-12-a) oder durch bestehende Kanten (siehe Abbildung 5-12-b) gezogen, werden diese gelöscht.



a) Prozesselemente löschen

b) Kante löschen

Abbildung 5-12: Löschgeste

Subprozess erstellen

Durch die abstrakte Art der Aktion *Subprozess erstellen* wurde auf eine Geste verzichtet. So erscheint nach der Selektion der Prozesselemente ein Button, um aus den ausgewählten Prozesselementen einen Subprozess zu erstellen (siehe Abbildung

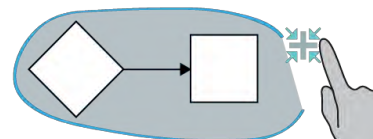


Abbildung 5-13: Subprozess erstellen

5-13).

Eine Subprozess-Aktivität wird im normalen Zustand wie eine Aktivität mit einem *Plus-Zeichen* dargestellt. Vergrößert der Benutzer diese mittels einer Pinch-Geste, wird der enthaltene Subprozess angezeigt (siehe Abbildung 5-14). Dies folgt Anforderung 7 nach einem direkten Interface und setzt den Ansatz eines *Interaction-Rich-Systems*.

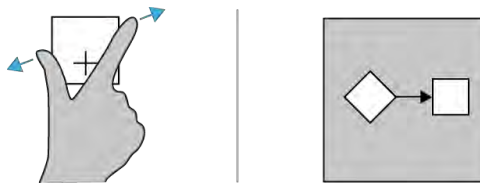


Abbildung 5-14: Subprozess öffnen

Da keine globale Skalierung angeboten wird (siehe Anforderung 15), soll es dennoch für Benutzer möglich sein, teile des Prozessmodells vergrößert zu betrachten. Hierzu eignet es sich, mit zwei Fingern der NDH einen Bereich zu berühren, der vergrößert werden soll. Anschließend kann die DH auf der Vergrößerung operieren (siehe Abbildung 5-15) [45]. Die Vergrößerung ist solange sichtbar, solange der Benutzer mit der NDH den Bildschirm berührt, um den Bereich der Vergrößerung zu definieren. Auf der Vergrößerung kann mit der DH interagiert werden.

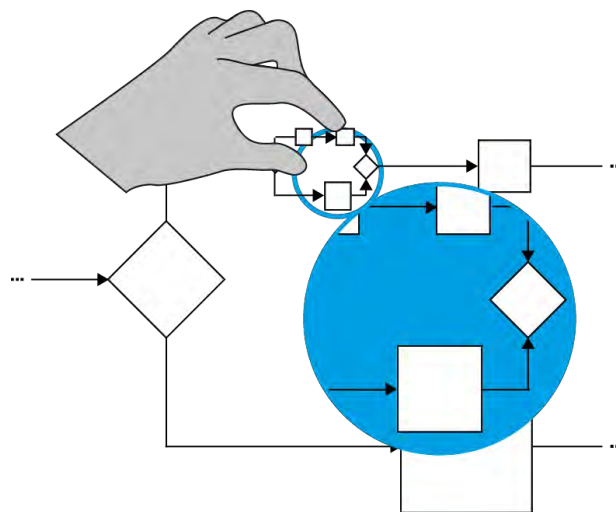


Abbildung 5-15: Lokale Vergrößerung

5.3 BENUTZEREINGABE MIT MOBILEN GERÄTEN

Wird mit dem Smartphone oder Tablet eine Tablet-Touch-Aktion, soll sich diese identisch der Fingereingabe verhalten. So ist es möglich mit mobilen Geräten zu skizzieren oder Menüs aufzurufen. Die Vorteile die es bietet, ein individuelles Eingabegerät in der Hand zu haben, werden hierbei genutzt um dem Benutzer eine einfache Bedienung des Systems zu erlauben (siehe Anforderung 8). Der daraus folgende Kontextwechsel wird Zugunsten der schnelleren und einfacheren Eingabe in Kauf genommen. Um dem Benutzer diesen Wechsel zu vereinfachen, wird auf dem Tabletop ein Hinweis gegeben, welcher suggeriert, dass die Eingabe auf dem mobilen Gerät erfolgt.

Berührt ein Benutzer den Einstiegspunkt eines Squenzflusses mit einem mobilen Gerät, erscheint anstelle des radialen Menüs eine Auswahlliste auf dem Gerät des Benutzers (siehe Abbildung 5-16). Der Vorteil, dass auf Smartphones mittels dem Daumen einfach durch horizontale Listen gescrollt werden kann, wird hier ausgenutzt und das Menü als Liste visualisiert.

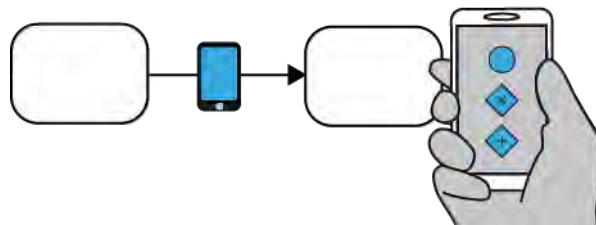


Abbildung 5-16: Menübasierte Auswahl auf dem Smartphone

Wird ein mobiles Gerät dazu verwendet, ein neues Prozesselement zu erstellen, oder umzubenennen, wird dieses als Spezialist für die Texteingabe gesehen. Durch Optimierungen der Texteingabe bei Smartphones [46, 47], kann die Geschwindigkeit der Eingabe erhöht werden. So kann jeder Benutzer auch individuell seine bevorzugte Art der Texteingabe verwenden, da die Einstellung auf dem eigenen mobilen Gerät stattfindet und nicht als globale Einstellung des Tabletops gespeichert wird. Um bei Texteingaben auf mobilen Geräten den Kontext nicht zu verlieren, wird dem Benutzer dieser auf dem mobilen Gerät dargestellt (siehe Abbildung 5-17).

Smartphones bieten als individuelle Ein- und Ausgabegeräte weitere Möglichkeiten der Interaktion. Durch die Nutzung des Mikrofons für Sprachaufnahmen und des Lautsprechers für deren Ausgabe, können auch Sprachanmerkungen *aufgenommen* (siehe Abbildung 5-18) und *abgespielt* werden ohne andere Benutzer abzulenken.

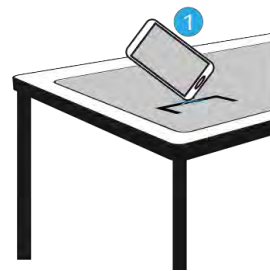
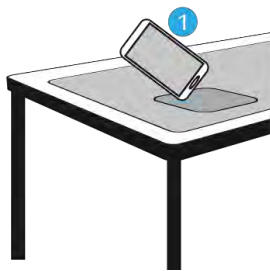


Abbildung 5-17: Smartphone als Tastatur

Abbildung 5-18: Sprachanmerkung

5.4 PRÜFUNG DER KORREKTEN BPMN SYNTAX

Um die syntaktische Qualität der Prozessmodelle zu erhöhen, wird vor einer Kantenstellung überprüft, ob dadurch die korrekte Syntax von BPMN eingehalten wird. Die verwendeten Regeln für die Syntaxprüfung sind in Tabelle 5-2 aufgeführt. Wird eine der aufgeführten Regeln nicht eingehalten, können Prozesselemente von Benutzern nicht miteinander verbunden werden.

Regel	Beschreibung
<i>Eine eingehende und ausgehende Kante bei Aktivitäten</i>	Jede Aktivität darf maximal eine eingehende und maximal eine ausgehende Kante besitzen. Eine implizite Aufspaltung des Sequenzflusses durch Aktivitäten, welche in BPMN einer AND-Aufspaltung gleichkommt, wird nicht unterstützt. Ebenso wird auf eine implizite Zusammenführung des Sequenzflusses durch Aktivitäten, welche in BPMN einer OR-Zusammenführung entspricht, verzichtet.
<i>Nur eingehende oder nur ausgehende Kanten bei Ereignissen</i>	Es werden nur Start- und Ende-Ereignisse unterstützt. Ein Start-Ereignis hat keine eingehenden, jedoch beliebig viele ausgehende Kanten. Ein Ende-Ereignis hat beliebig viele eingehende, jedoch keine ausgehenden Kanten.

<i>Entweder aufspalten oder zusammenführen des Sequenzflusses bei Gateways</i>	Gateways können Sequenzflüsse aufspalten oder zusammenführen. Die Kombination der beiden Möglichkeiten in einem Gateway ist von BPMN zwar erlaubt, macht Prozessmodelle jedoch unübersichtlich. In Process-Touch wird dies jedoch nicht unterstützt. Daher kann ein einzelnes Gateway Sequenzflüsse entweder Aufspalten, oder Zusammenführen.
<i>Schleifen nur mit Gateways</i>	Um Endlosschleifen innerhalb von Prozessmodellen zu verhindern, wird überprüft, ob sich innerhalb einer Schleife ein Gateway befindet.
<i>Keine Mehrfachkanten</i>	Prozesselemente können nicht mehrfach miteinander verbunden werden.
<i>SESE bei Subprozessen</i>	Ein <i>SESE-Block</i> (Single Entry Single Exit) ist ein Teil des Prozessmodells, wobei der gesamte SESE-Block genau eine eingehende und genau eine ausgehende Kante besitzt [48]. Ein Subprozess kann nur aus einem SESE-Block erstellt werden.

Tabelle 5-2: Regeln der Syntaxprüfung

5.5 LAYOUTING VON PROZESSELEMENTEN

Um dem Benutzer die Anordnung von Prozesselementen zu vereinfachen und der Anforderung nach einem natürlichen, an der Realität angelehnten System nachzukommen (siehe Anforderung 3), dürfen keine Raster verwendet werden [30]. Globale Aktionen, wie z.B. ein Layouting-Button, widerspricht Anforderung 15, welche globale Aktionen ablehnt. Daher sollen Prozesselemente während der Interaktion immer *korrekt* angeordnet werden. Die Anordnung der Prozesselemente muss hierbei animiert von statten gehen, damit es für den Benutzer nachvollziehbar ist, wohin unterschiedliche Prozesselemente verschoben werden. Ziel ist es, ein sich selbst layoutendes Prozessmodell zu erstellen. Die Verwendung eines *Force-Directed-Graph-Algorithmus* ist dafür geeignet [49]. Die Wirkung auf den Benutzer ist ein zusammenhängendes System, welches sich „lebendig“ anfühlt.

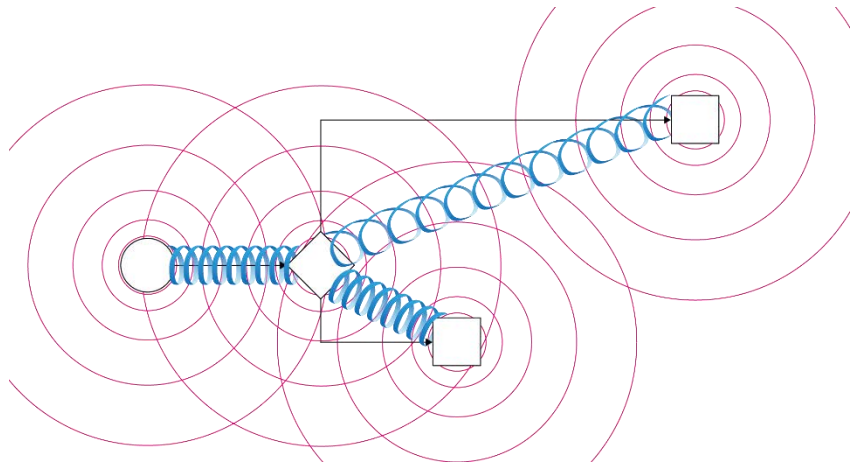


Abbildung 5-19: Anziehung (Blau) und Abstoßung (rot) der Knoten

Das Prinzip des Layouting-Verfahrens ist es, dass Kanten zwischen zwei Prozesselemente als Feder fungieren und folglich diese zusammenhalten. Wird eines der verbundenen Prozesselemente verschoben, wie durch eine Feder verbunden (siehe Abbildung 5-19, blaue Federn). Um eine mögliche Überlagerung der Prozesselemente zu verhindern, stoßen sich Prozesselemente zusätzlich gegenseitig ab. Die Abstoßung ist ähnlich wie bei gleichgepolten Magneten (siehe Abbildung 5-19, rote Felder).

In Abbildung 5-20 ist ein Beispiel des Layouting-Verfahrens zu sehen. Zu Beginn sind die Kräfte F_1 , F_2 und F_3 , welche auf die korrespondierenden Prozesselemente wirken, so gering, dass die Prozesselemente an der jeweiligen Position bleiben (siehe Abbildung 5-20-a). Der Benutzer verschiebt das zusammenführende Gateway, sodass auf die beiden Aktivitäten eine höhere Kraft wirkt, da sie mit dem zusammenführenden Gateway verbunden sind (siehe Abbildung 5-20-b). Daraufhin werden die Aktivitäten so lange in die Richtung der korrespondierenden Kraft verschoben, bis sich diese wieder unter einem bestimmten Schwellwert befinden (siehe Abbildung 5-20-c). Durch die Verschiebung der Aktivitäten, erhöht sich die Kraft F_1 (siehe Abbildung 5-20-c), und das aufspaltende Gateway wird in Richtung der Kraft F_1 verschoben (siehe Abbildung 5-20-d).

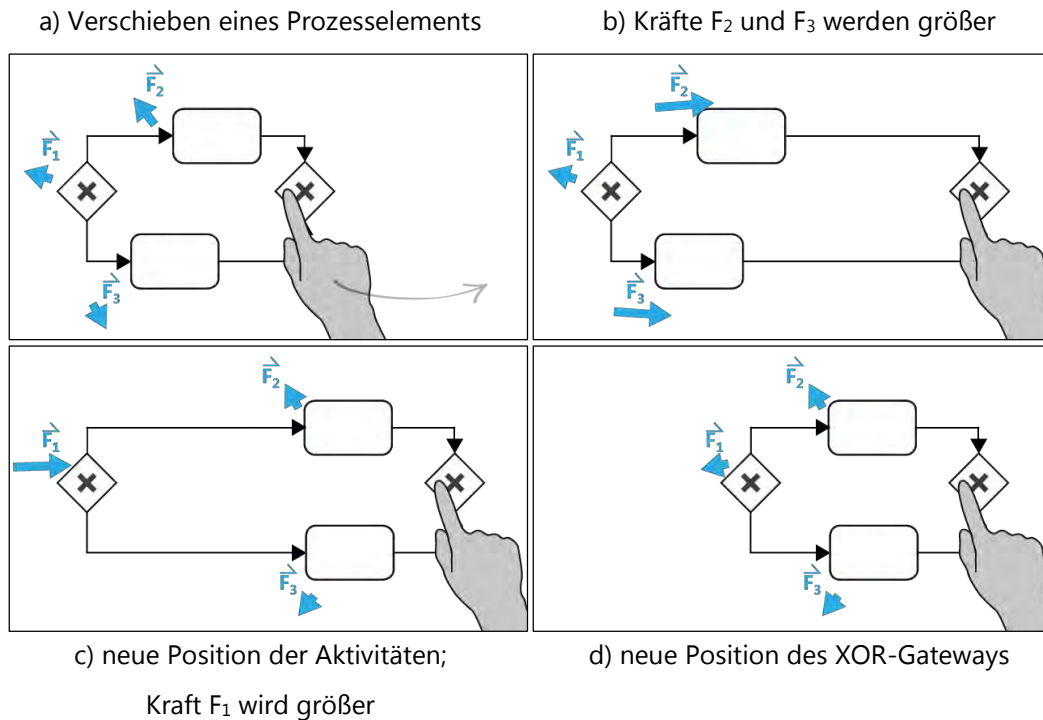


Abbildung 5-20: Ablauf des Layouting-Verfahrens

Dieses Verfahren steht jedoch im Widerspruch zu einigen Gesten und bedarf daher Ausnahmen:

- Prozesselemente werden verbunden, indem das Quellelement auf das Zielelement gezogen wird. Das Layouting würde diese Interaktion jedoch nicht zulassen, da das Quellelement das Zielelement verdrängen würde. Somit muss hierfür eine Ausnahme gelten: Ist es nach den Regeln von BPMN möglich, das Quellelement (wird gerade vom Benutzer verschoben) mit einem Prozesselement zu verbinden, darf keine abstoßende Kraft zwischen dem Quellelement und dem Prozesselement wirken. Diese Ausnahme gilt nur während der Benutzer ein Prozesselement verschiebt.
- Die Interaktion mit Subprozessen stellt eine weitere Ausnahme dar, sofern die Kindelemente des Subprozesses angezeigt werden. Um Prozesselemente innerhalb eines Subprozesses zu verschieben, darf der gesamte Subprozess nicht von dem Quellelement verdrängt werden.
- Datenflüsse erstrecken sich oft über große Teile eines Prozessmodells. Damit anziehende Kräfte zwischen Prozesselementen mit gemeinsamen Datenfluss das Layout nicht zerstören, haben Datenflüsse keine anziehenden Kräfte. Implizit lassen die gestrichelten Linien der Datenflüsse den Be-

nutzer darauf schließen, dass hier keine bzw. weniger Kräfte als bei durchgezogenen Linien wirken.

5.6 ZUSAMMENFASSUNG

Das Kapitel zeigt auf, wie das System Process-Touch von Benutzern verwendet werden kann und welche Aktionen angeboten werden. Dabei wird die Notwendigkeit von gesten- und menübasierter Benutzereingabe aufgezeigt. Das Gestenset wird bestimmt und die Orientierung von Prozesselementen wird diskutiert. Es wird erläutert, wie sich mobile Geräte einbringen lassen und wie das Layoutingverfahren funktioniert.

Das in diesem Kapitel erstellte Feinkonzept wird im nächsten Kapitel visuell angereichert um ein einheitliches Look & Feel zu schaffen.

6 VISUAL DESIGN

Das Konzept von Process-Touch wird in diesem Kapitel visuell ausgearbeitet. Hierbei gilt es, Forderungen der Guidelines nach spezieller optischer Gestaltung unterschiedlicher Plattformen, wie z.B. Microsoft Pixel Sense und Windows 8, nachzugehen, um ein einheitliches Look&Feel des Gesamtsystems zu gewährleisten [30, 50]. Forderungen der Guidelines können dabei einen technischen oder ästhetischen Ursprung haben, oder berufen sich auf plattformweite Standards. Da das System Process Touch nicht für spezielle Geräte und Plattformen konzipiert ist, werden allgemeine Grundsätze der Gestaltung definiert.

Um ein einheitliches und konsistentes Interface anzubieten, werden folgende Grundsätze in allen Interaktionselementen verwendet [30]:

- Alle Elemente sollen ohne Rahmen dargestellt werden um den Fokus auf den Inhalt zu richten (Anforderung 7). Es sollen auch keine Texturen oder Farbverläufe verwendet werden, um das Interface möglichst schlicht zu halten.
- Linien sollen mindestens 2px breit sein, damit sie auch in rotiertem Zustand auf dem Display sauber dargestellt werden können.
- Alle klickbaren Elemente sollen mindestens 18mm breit und 18mm hoch sein. Der Abstand klickbarer Elemente zueinander soll mindestens 3mm betragen. Dieser Grundsatz gilt für alle Elemente in ihrer Standardgröße.
- Auf jede Benutzereingabe soll das System unmittelbar reagieren (Anforderung 19).

Darauf aufbauend, werden in Kapitel 6.1 Farben für die einzelnen Interaktionselemente definiert. Kapitel 6.2 zeigt die verwendeten Icons auf und beschäftigt sich mit der Typographie der Beschriftungen der Interaktionselemente. Deren Formen werden in Kapitel 6.3 bestimmt. Kapitel 6.5 beschäftigt sich mit den Rückmeldungen, welche dem Benutzer während der Interaktion gegeben werden.

6.1 FARBEN IN PROCESS TOUCH

Dieser Abschnitt beschäftigt sich mit der Farbwahl für den Hintergrund der einzelnen Interaktionselemente.

Große Displays, wie Tabletop-Systeme, können den Benutzer optisch blenden, wenn helle und leuchtende Farben großflächig verwendet werden [30]. Dies kann vor allem bei längerer Benutzung des Systems unangenehm sein. Folglich soll für das visuelle Design ein dunkler Hintergrund verwendet werden. Kleinere Flächen und Linien sollen mit satten und kräftigen Farben hervorgehoben werden [30]. Damit die Anwendung nicht zu bunt und verspielt wirkt, wird als Hintergrundfarbe ein dunkles Grau verwendet (siehe Abbildung 6-1).

Farben werden je nach kulturellem Kontext mit Eigenschaften oder Aussagen assoziiert. Diese Arbeit bezieht sich im Weiteren auf den westlichen Kulturraum. So wird die Farbe *Rot* als Warnfarbe verstanden, wobei *Grün* für Erfolg und Bestätigung steht. *Gelb* wird als neutrale Farbe verwendet um auf eine bestimmte Aktion oder auf einen bestimmten Systemzustand aufmerksam zu machen. Diese drei Farben werden mit der beschriebenen Semantik verwendet, um Benutzern Rückmeldungen zu geben (siehe Kapitel 6.5). Die verwendeten Farbtöne sind in Abbildung 6-1 mit den RGB-Werten dargestellt.

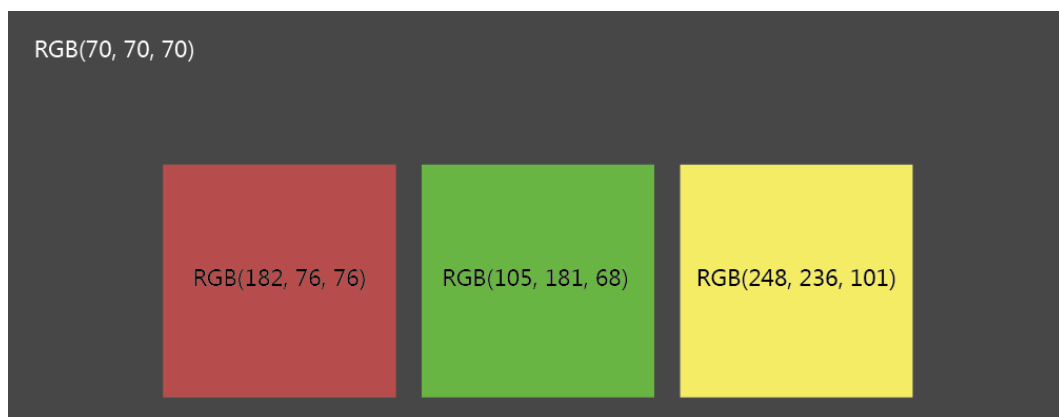


Abbildung 6-1: Farbtöne für Rückmeldungen auf der Hintergrundfarbe mit RGB-Werten

Neben Farbtönen, welchen eine Semantik zugordnet werden kann, sollen unterschiedliche Bereiche des Systems durch semantisch neutrale Farben gruppiert werden. So wird ein sattes Blau als Informationsfarbe für Markierungen, dem Kontextmenü und der Tastatur verwendet. Somit werden Funktionen, welche Prozesselemente verändern, gruppiert. Das Hauptmenü und die Interaktionselemente der Verbindungspfeile werden in der Farbe Hellgrün eingefärbt. Datenelemente und Datenflüsse werden in Magenta dargestellt, um sich von Aktivitäten,

Events, Gateways und Sequenzflüssen abzuheben. Der Eingabepfad, der die Lasso-Geste repräsentiert, wird mit einem hellen Blau gefüllt, um sich deutlich von der Skizzeneingabe abzugrenzen und die innenliegende Fläche des Eingabepfads zu betonen. Die Farben sind in Abbildung 6-2 dargestellt.

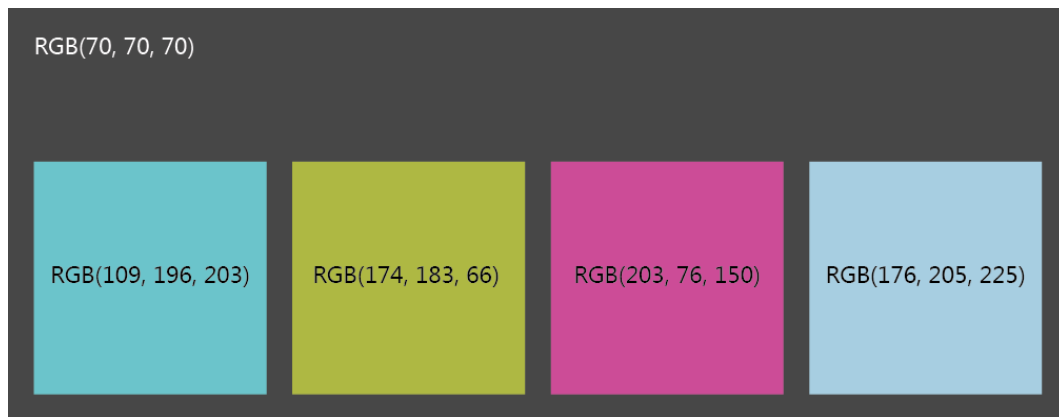


Abbildung 6-2: Informationsfarben auf der Hintergrundfarbe mit RGB-Werten

Neben den vorgestellten Farben, wird noch die Farbe Weiß für Prozesselemente, und die Farbe Schwarz bzw. Dunkelgrau für Beschriftungen verwendet. Das dunkle Grau wird vor allem bei Beschriftungen verwendet, welche auf hellen Flächen positioniert sind. Dadurch verringert sich der Kontrast und die Beschriftung wird besser lesbar, da sie auf Displays besser wiedergegeben werden kann [30]. Dies macht sich bei rotierten Elementen besonders bemerkbar.

6.2 ICONS UD TYPOGRAPHIE

Icons dienen der visuellen Beschreibung von Aktionen und unterstützen Benutzer, diese leichter zu verstehen und eine gewünschte Aktion schneller zu finden [51]. Dafür müssen Icons folgende Anforderungen erfüllen [30]:

- Icons müssen eine einfache Geometrie aufweisen.
- Icons dürfen nicht zu detailliert sein.
- Icons müssen skalierbar sein.
- Icons müssen sich guten Metaphern aus der Realität bedienen.
- Icons müssen ein ähnliches visuelles Gewicht besitzen.

Die Bedeutung der Icons muss dennoch von Benutzern erst erlernt werden. Deshalb sollen Icons zusätzlich beschriftet werden und ein konsistentes Erscheinungsbild wird erschaffen.

Für das Hauptmenü, in dem neue Prozesselemente erstellt werden können, werden kleine Vorschaubilder der Prozesselemente als Icons verwendet. Abbildung 6-3 zeigt die Icons des Hauptmenüs.



Abbildung 6-3: Icons des Hauptmenüs

Mittels dem Kontextmenü der Prozesselemente können deren Eigenschaften verändert, und Aktionen, welche sich auf das Prozesselement beziehen, ausgeführt werden. Die Icons des Kontextmenüs der Prozesselemente sind in Abbildung 6-4 zu sehen.



Abbildung 6-4: Icons des Kontextmenüs

Neben Icons, werden auch Prozesselemente Beschriftet und im ersten Schritt muss die Beschriftung vom Benutzer gelesen werden können. Vor allem in NUIs, wo Interaktionselemente und deren Beschriftung frei rotierbar sind, muss darauf geachtet werden, dass alle Buchstaben in unterschiedlichen Winkeln auf einem Display gut dargestellt werden können und somit gut lesbar sind. Die Schriftart *Segoe360* wurde speziell dafür entwickelt [30]. So sind z.B. bei den Buchstaben *a*, *e*, *c*, die *Counters* (dt.: Öffnungen innerhalb von Buchstaben) weiter geöffnet, um diese im gedrehten Zustand deutlicher zu erkennen. Zusätzlich ist der Buchstabenabstand größer, damit Buchstaben im gedrehten Zustand nicht miteinander verschmelzen und einige Buchstaben sind angepasst, um sie besser lesen zu können, wenn diese auf dem Kopf stehen. So ist das kleine „l“ verändert worden, um es von einem großen „I“ besser zu unterscheiden. Auch das kleine „q“ ähnelt bei vielen Schriftarten in gedrehtem Zustand dem kleinen „b“ (siehe Abbildung 6-5).



Abbildung 6-5: Anpassung von Buchstaben bei der Schriftart Segoe360 [30]

6.3 FORMEN DER INTERAKTIONSELEMENTE

Die Form der Prozesselemente ist in der BPMN bereits definiert (siehe Kapitel 3.1). Events werden als Kreise und Gateways als Rauten (um 45° rotierte Quadrate) dargestellt. Aktivitäten sind als Rechtecke definiert. Abbildung 6-6 zeigt die Gestaltung der verwendeten Prozesselemente (Aktivitäten, Events, Gateways und Datenelementen).

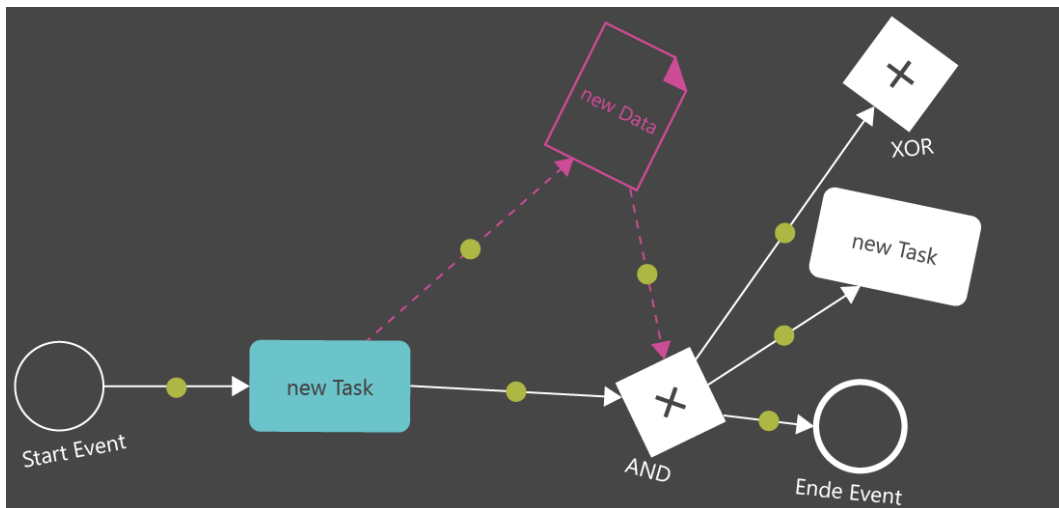


Abbildung 6-6: Prozesselemente in Process Touch

Die Kreise auf den Verbindungspfeilen sind in Farbe und Form dem Hauptmenü nachempfunden (siehe Gesetz der Ähnlichkeit in Kapitel 3.5). Ein Tap auf diese Kreise, bzw. eine Pinch-Geste auf den Hintergrund, öffnet ein neues Hauptmenü. Dabei ist Mittelpunkt des Hauptmenüs der Mittelpunkt der Pinch-Geste, bzw. der Mittelpunkt des Kreises auf einem Verbindungspfeil. Da die Auswahl an Prozesselementen keiner Priorisierung unterliegt, sind alle Icons im Hauptmenü gleich schnell auswählbar. Nach *Fitt's Law* müssen somit alle Elemente in gleicher Größe und gleichem Abstand zur aktuellen Position des Fingers dargestellt werden (siehe Kapitel 3.5). Dies lässt sich mit einem radialen Aufbau realisieren. Abbildung 6-7 zeigt das radiale Hauptmenü welches in der Farbe Hellgrün eingefärbt wurde.

Um den Kontrast zum Hintergrund zu erhöhen, werden Icons und deren Beschriftung in der Farbe Weiß dargestellt.

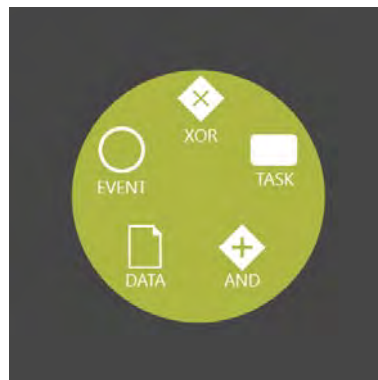


Abbildung 6-7: Hauptmenü

Das Hauptmenü überdeckt häufig angrenzenden Prozesselemente (siehe Abbildung 6-8). Aufgrund des *Gesetzes der Schließung* (siehe Kapitel 3.5) kann der Benutzer dennoch die teils verdeckten Elemente problemlos erkennen. Dadurch ist sichergestellt, dass dem Benutzer der Kontext des Menüs erhalten bleibt.

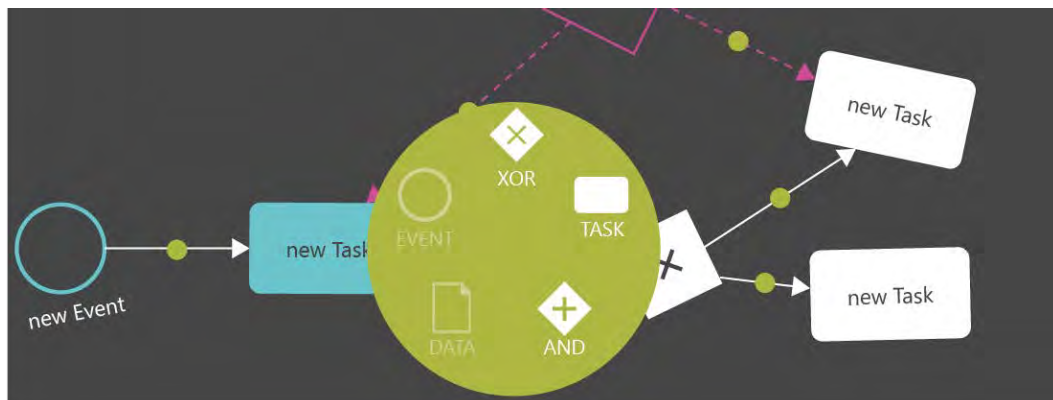


Abbildung 6-8: Verbindungsmenü

6.4 INTERAKTIONSELEMENTE

Mit einem langen Tap (1 Sek.) auf ein Prozesselement wird das Kontextmenü geöffnet. In diesem Menü kann der Benutzer das ausgewählte Prozesselement *duplizieren*, daraus einen *Subprozess erstellen*, einen Subprozess *entpacken*, den *Text ändern* und das Prozesselement *löschen*. Auch wenn ein langer Tap ausgeführt wird, muss das System darauf reagieren, um dem Benutzer mitzuteilen, dass die Eingabe erkannt wurde (siehe Anforderung 19). Dies wird mittels einer Animation,

welche über die Wartezeit hinweg einen vollen Kreis zeichnet, visualisiert (siehe Abbildung 6-9).

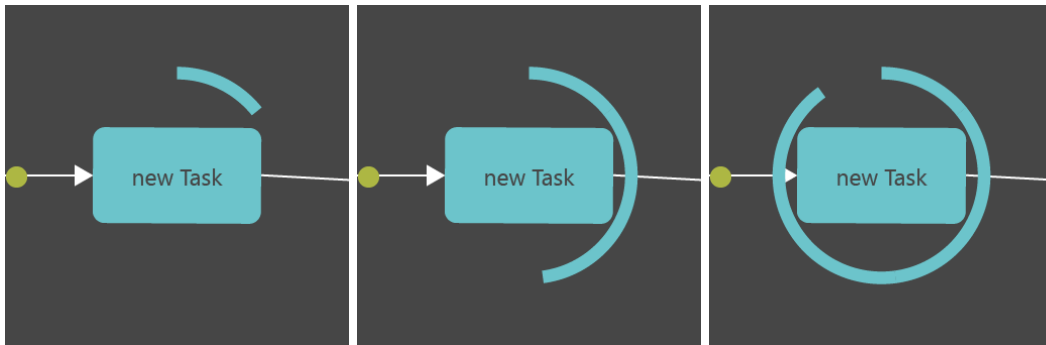


Abbildung 6-9: Animation des Kontextmenüs

Das Kontextmenü ist ebenfalls als radiales Menü dargestellt und die Darstellung unterscheidet sich von der Darstellung des Hauptmenüs. Die Zusammengehörigkeit des Prozesselements zu den Elementen des Menüs wird mittels dem *Gesetz der Ähnlichkeit* (gleiche Farbe und Orientierung; siehe Kapitel 3.5) und dem *Gesetz der Nähe* (siehe Kapitel 3.5) erreicht.



Abbildung 6-10: Kontextmenü eines Prozesselements

Die Aktion Text ändern, öffnet eine virtuelle Tastatur. Diese ist durch das *Gesetz der Ähnlichkeit* (gleiche Farbe und Orientierung) und dem *Gesetz der Nähe* mit dem Prozesselement optisch verbunden. Der Aufbau der Tastatur ist dabei einer typischen Tastatur von Smartphones nachempfunden. Das Ziel ist es, auf möglichst geringem Platz eine gut handzuhabende Tastatur anzubieten (siehe Abbildung 6-11).

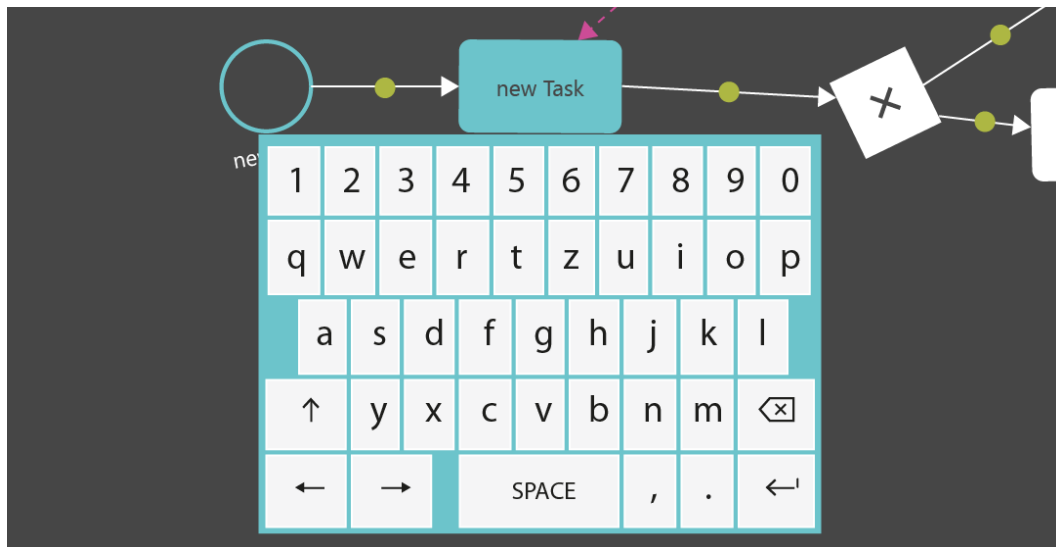


Abbildung 6-11: virtuelle Tastatur auf dem Tabletop-System

Um die Stärken der Tablets als individuelles *Single-User-System* auszunutzen, ist auf den Tablets das Hauptmenü, das Kontextmenü und die Tastatur anders als auf dem Tabletop-System dargestellt. Die hier vorgestellte Visualisierung stützt sich auf die Windows 8 Design Guidelines [50].

Das Hauptmenü wird auf jedem Tablet nur einmal dargestellt, da immer nur ein Benutzer mit einem Tablet arbeitet. Die Funktionen des Hauptmenüs, werden am unteren Rand des Tablet dargestellt und lässt sich mit einer Wischgeste, welche unterhalb des Displays startet und nach oben auf das Display führt, anzeigen (siehe Abbildung 6-12). Die Farbe des Hauptmenüs ist dieselbe wie auf dem Tabletop-System, damit sich Benutzer leichter orientieren können. Auf der linken Seite des Hauptmenüs sind, im Vergleich zu der Tabletop-Variante, spezielle Funktionen für ein *Single-User-System* gruppiert. Damit können alle Prozesselemente *selektiert*, die *Einstellungen* aufgerufen und eine Aktion *rückgängig* gemacht werden. Auf der rechten Seite des Hauptmenüs befinden sich die wichtigsten Funktionen, da die meisten Benutzer Rechtshänder sind und somit besser und schneller mit der rechten Hand interagieren können.

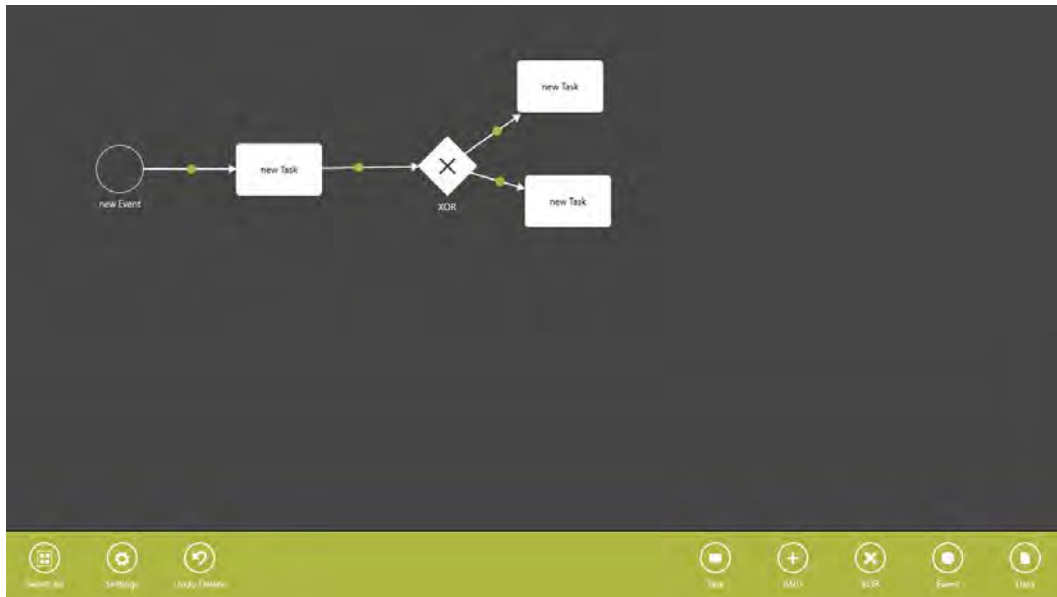


Abbildung 6-12: Hauptmenü auf dem Tablet

Das Kontextmenü ist auf den Tablets als Sidebar visualisiert, damit der Benutzer eine Übersicht über die möglichen Einstellungen und Aktionen bekommt. Die Sidebar ist auf der rechten Seite positioniert, um Rechtshändern eine bessere Interaktion zu bieten (siehe Abbildung 6-13). Wird das Kontextmenü angezeigt, wird das dargestellte Prozesselement als einziges Prozesselement auf der Oberfläche markiert.

Um Beschriftungen der Prozesselemente zu ändern, wird eine Textbox innerhalb der Sidebar angeboten. Als virtuelle Tastatur greift das System Process-Touch auf die vom Betriebssystem angebotene Tastatur zurück (siehe Abbildung 6-14).



Abbildung 6-13: Kontextmenü des Tablets

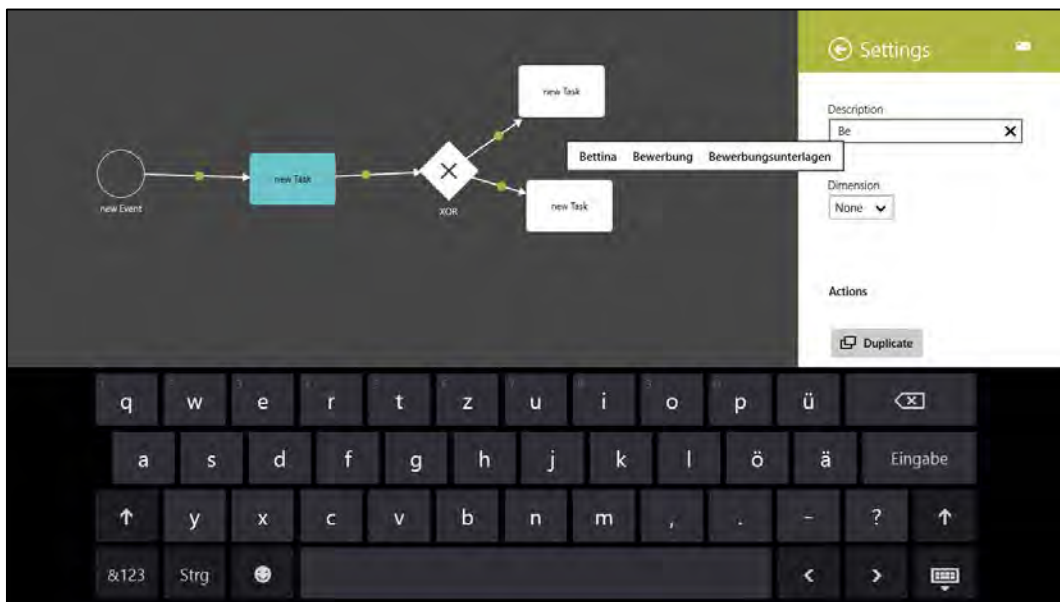


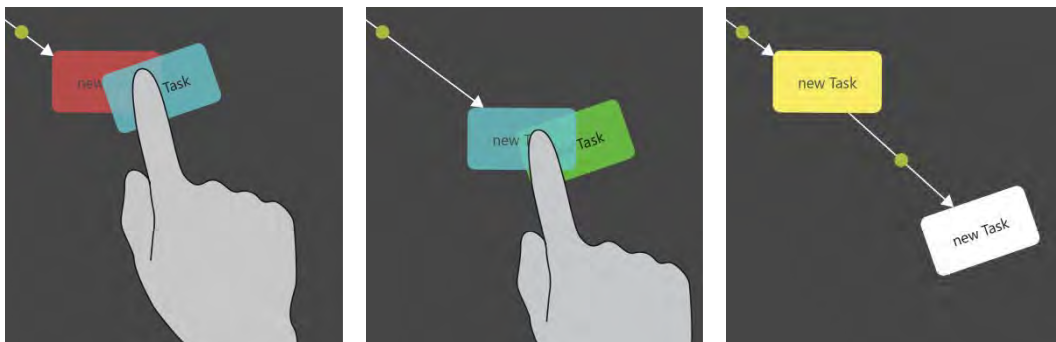
Abbildung 6-14: Tastatur auf dem Tablet mit Autovervollständigung

6.5 RÜCKMELDUNGEN

Die Rückmeldung des Systems auf Benutzereingaben werden in diesem Abschnitt vorgestellt. Zum einen wird auf die Rückmeldungen bei Verbindungsaktionen von Prozesselementen, bzgl. der Syntaxprüfung eingegangen (siehe Kapitel 6.5). Weiter zeigt das Kapitel auf, wie das System auf Tablet Touch Aktionen reagiert (siehe Kapitel 6.5.2).

6.5.1 SYNTAXPRÜFUNG

Rückmeldungen bei Verletzung bzw. Einhaltung der BPMN Syntax (siehe Kapitel 5.4) werden visuell durch unterschiedliche Farben dargestellt (siehe Anforderung 2). Versucht ein Benutzer eine Verbindungsoperation auszuführen, welche nicht syntaxkonform ist, wird das Zielelement mit der Farbe Rot eingefärbt (siehe Abbildung 6-15-a). Ist die Verbindungsaktion syntaktisch korrekt, wird das Zielelement mit der Farbe Grün eingefärbt (siehe Abbildung 6-15-b). Um die Farben deutlich wahrnehmen zu können, wird das verschobene Prozesselement halbtransparent dargestellt. Nach der Verbindungsaktion springt das Quellelement wieder an die ursprüngliche Position und wird für kurze Zeit mit der Farbe Gelb eingefärbt, um auf die neue Position aufmerksam zu machen (siehe Abbildung 6-15-c).



a) Verletzung der Syntax

b) Einhaltung der Syntax

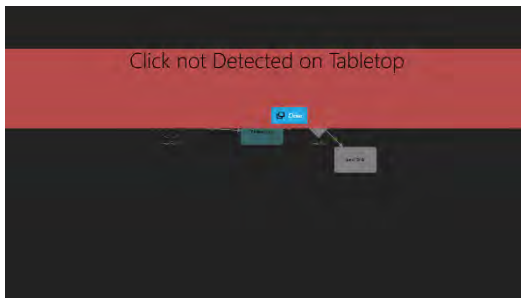
c) Nach Verbindung ist das Quellelement wieder an der ursprünglichen Position.

Abbildung 6-15: Rückmeldungen bei Verbindungsoperationen

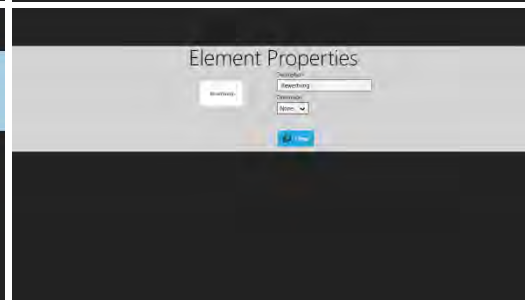
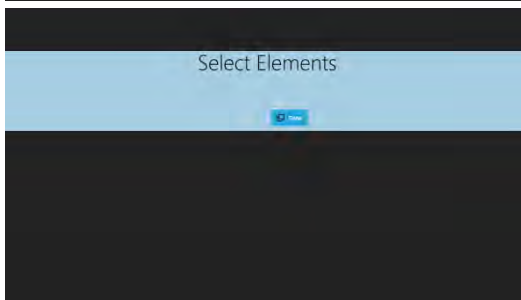
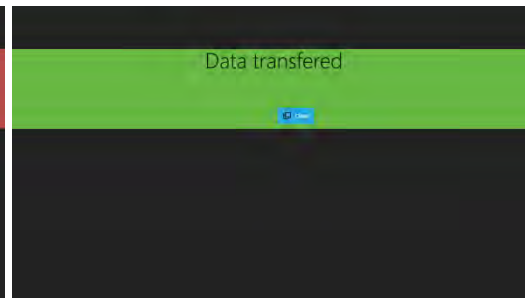
6.5.2 TABLET-TOUCH-AKTIONEN

Werden Tablet Touch-Aktionen ausgeführt, sollen deutliche und leicht erkennbare Rückmeldungen seitens der Tablets gegeben werden, da während der Benutzeraktion das Tablet vom Körper weggehalten wird somit für den Benutzer nicht gut sichtbar ist. Es werden vier verschiedene Rückmeldungen angeboten, welche als *modales Pop-up* dargestellt werden. Diese werden mittels großen und farbigen Flächen dargestellt und stellt einen bewussten Bruch der Regel dar, dass intensive Farben nur auf kleinen Flächen verwendet werden sollen. Drei verschiedene Arten der Benutzerinteraktion und ein Fehlerfall sind möglich (siehe Abbildung 6-16):

a) Nicht erkannter Tablet-Touch



b) Prozesselemente auf Tabletop übertragen



c) Prozesselemente mit Tablet selektieren

d) Prozesselement bearbeiten

Abbildung 6-16: Tablettouch Rückmeldungen

- Wird auf dem Tablet eine Tablet-Touch-Aktion erkannt, kann jedoch nicht auf dem Tabletop-System zugeordnet werden, erscheint eine *Fehlermeldung* (siehe Abbildung 6-16-a). Der Hintergrund des modalen Pop-up-Menüs ist dabei in der Farbe Rot eingefärbt. Das Pop-up-Menü verschwindet nach kurzer Zeit automatisch wieder. Alternativ kann der Benutzer das Pop-up-Menü mit dem Close-Button schließen.
- Transferiert der Benutzer Prozesselemente, welche auf dem Tablet erstellt wurden, mittels einer Tablet-Touch-Aktion auf den Tabletop, wird eine *positive Rückmeldung* angezeigt (siehe Abbildung 6-16-b). Der Hintergrund der positiven Rückmeldung in der Farbe Grün eingefärbt. Diese schließt sich auch automatisch, jedoch deutlich später als die Fehlermeldung. Dies hat den Grund, dass der Benutzer mehr Zeit zum lesen haben soll, da die Elemente vom Tablet *verschwunden* sind und er nicht von einem Fehler ausgehen muss.
- Startet der Benutzer mittels einer Tablet-Touch-Aktion eine *Selektion* von Prozesselementen, wird die Rückmeldung auf dem Tablet in der gleichen Farbe wie die Füllung der Selektion auf dem Tabletop-System, eingefärbt (siehe Abbildung 6-16-c). Damit wird dem Benutzer der Zusammenhang

zwischen der Selektion auf dem Tabletop-System und dem Tablet angezeigt.

- Berührt der Benutzer mit der Tablet-Touch-Aktion ein Prozesselement auf dem Tabletop-System, kann er dieses als Kopie auf dem Tablet bearbeiten (siehe Abbildung 6-16-d). Um ihm zu verdeutlichen, dass es sich um eine Kopie handelt und das Prozesselement nicht transferiert wurde, wird es nur in dem Pop-up-Menü angezeigt. Die Farbe des Pop-up-Menüs ist an der Farbe der Prozesselemente angelehnt und in Hellgrau gehalten.

6.6 ZUSAMMENFASSUNG

Dieses Kapitel zeigt das visuelle Design von Process Touch, welches ein schlichtes und funktionales Design widerspiegelt. Konsistente Farben, Menüs und Icons der Interaktionselemente unterstützen den Benutzer in der Interaktion. Durch die zusätzliche Verwendung von informierenden Farben im Fehler- und Erfolgsfall bekommt der Benutzer subtile, aber leicht erkennbare und gut verstehbare Rückmeldungen bzgl. des Systemzustands. Die teils unterschiedliche Gestaltung des Tabletop- und Tablet-Interfaces führt dazu, dass das Tablet seine Stärken als Single-User-System ausspielen kann.

Auf Basis der visuellen Ausarbeitung der einzelnen Interaktionselemente, Menüs und Dialoge, wird im nächsten Kapitel Process Touch implementiert.

7 PROOF-OF-CONCEPT-IMPLEMENTIERUNG

Developing the user interface of a professional software application is not easy. It can be a murky blend of data, interaction design, visual design, connectivity, multithreading, security, internationalization, validation, unit testing, and a touch of voodoo.

- Josh Smith [52]

In diesem Kapitel wird der Systemaufbau von Process Touch aus technischer Sicht betrachtet und die verwendeten Technologien, Software Pattern (siehe Kapitel 7.1) und Konzepte der Implementierung (siehe Kapitel 7.2) werden detailliert vorgestellt. Nicht alle Konzepte aus Kapitel 5 werden implementiert. Die lokale Vergrößerung (siehe Kapitel 5.2) und textuelle bzw. auditive Anmerkungen sind nicht umgesetzt. Zudem werden nur Tablets als mobile Geräte verwendet. Speziell wird auf die Kommunikation zwischen dem Tabletop-System und den Tablets (siehe Kapitel 7.3), auf die skizzenbasierte Benutzereingabe (siehe Kapitel 7.4) und auf das Layouting-Verfahren (siehe Kapitel 7.5) eingegangen.

Die Proof-of-Concept-Implementierung wird in der Programmiersprache C# entwickelt und verwendet das Grafik-Framework WPF (*Windows Presentation Foundation*), welches Teil des .NET-Frameworks ist [53].

Um das visuelle Design, das in Kapitel 6 vorgestellt wurde zu adaptieren, bietet das Grafik-Framework WPF viele Möglichkeiten. Durch die deklarative Auszeichnungssprache XAML die kann das Aussehen der einzelnen *Steuerelemente* von ihrer Funktionalität und Logik getrennt werden [54]. Für die graphische Darstellung der Steuerelemente wird von WPF die *Direct3D-API* angesprochen [55] rechenintensive Aufgaben wie z.B. *Animationen* und *Effekte* werden von der Graphikkarte übernommen. Zudem kann WPF *OpenType*- und *TrueType*-Schriftarten darstellen, wodurch eine typographische Anpassung möglich ist. Durch die Unterstützung des *aRGB-Farbraums*, welcher zusätzlich zu dem *Rot*-, *Grün*- und *Blau*-Kanal einen eigenen *Transparenz-Kanal (Alpha)* anbietet, kann das visuelle Design vollständig umgesetzt werden.

7.1 SOFTWAREARCHITEKTUR

Process-Touch gliedert sich in zwei eigenständige Programme. Zum einen ist eine Anwendung für den Microsoft Pixel Sense notwendig, welche auch als Serverinstanz für die Kommunikation mit den mobilen Geräten dient. Diese Anwendung basiert auf dem Pixel Sense-SDK und ist nur unter Windows 7 lauffähig. Einzuordnen ist diese Anwendung in Abbildung 7-1 auf dem Pfad *Desktop Apps* → *C#* → *.NET*. Die zweite Anwendung ist eine *Windows-Store-App* [56], welche auf mobilen Geräten mit Microsoft WindowsRT ausgeführt wird und ist in Abbildung 7-1 auf dem Pfad *Windows style Apps* → *XAML* → *C#* → *WinRT APIs* zu finden.

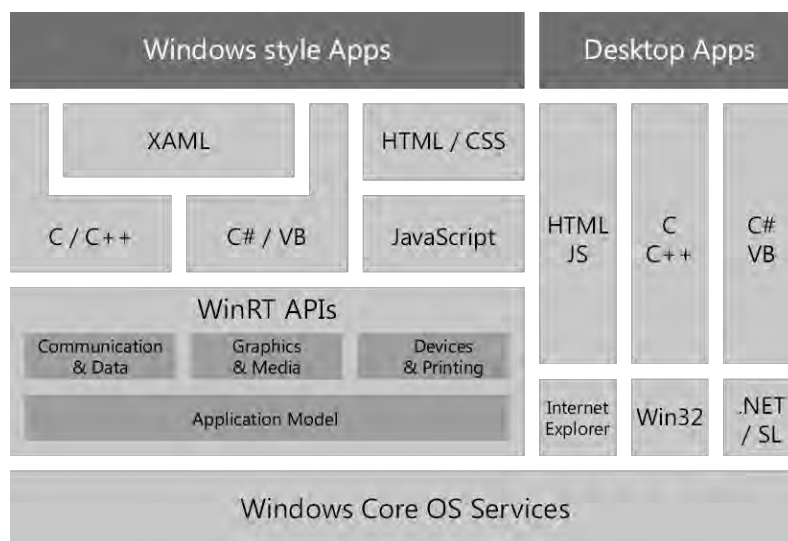


Abbildung 7-1: Technologien von Microsoft Windows [57]

Bei der Einordnung der Programme in das in Abbildung 7-1 dargestellte Schema, zeigt sich, dass beide Programme über unterschiedliche APIs auf unterschiedlichen Windows Cores (das von Windows 7 und das von Windows 8) zugreifen.

Um redundanten Code bezüglich der zwei unterschiedlichen Plattformen (Windows8 App und Windows7 Anwendung) zu minimieren und eine hoch interaktive GUI aufzubauen, ist der Aufbau der Programme bezüglich der Softwarearchitektur entscheidend. Das *Model-View-ViewModel (MVVM)* Design-Pattern (siehe Kapitel 7.1.1) liefert die Grundlage dafür, redundanten Code zu vermeiden [56]. Teile des Softwareprojektes werden anschließend zu einer *Portable Class Library* (siehe Kapitel 7.1.2) kompiliert, um sie auf beiden Plattformen ausführen zu können [57].

Ein *Design Pattern* ist ein Regelsatz, welcher einer Implementierungsempfehlung gleichkommt. Die Implementierung richtet sich stark an das vorgestellte Pattern, bricht dieses jedoch an einigen Stellen um flexibler zu sein und Overhead einzusparen [52].

7.1.1 DAS DESIGN-PATTERN MODEL-VIEW-VIEWMODEL

MVVM, steht für *Model-View-ViewModel* und ist ein weit verbreitetes Design-Pattern für die Entwicklung von Programmen unter C# mit WPF / XAML [52, 56]. Das MVVM Design-Pattern ist eine Weiterentwicklung des *MVC* (*Model-View-Controller*) bzw. des *MVP* (*Model-View-Presenter*) Design-Patterns [58]. Es gliedert das System in die drei Bereiche *Model*, *View* und *ViewModel* (siehe Abbildung 7-2). Diese haben folgende Aufgaben:

- Das *Model* hält die Daten und implementiert die *Geschäftslogik*.
- Das *ViewModel* (VM) abstrahiert das *Model* es macht es der *View* zugänglich. Zusätzlich bietet das VM *Commands* an, welche von den *EventHandlern* der *View* aufgerufen werden. VM Objekte halten nur *Referenzen* der *Models* bzw. von anderen VM, nicht aber der *View*. Daraus folgt eine *lose* Kopplung der *View*.
- Die *View* wird mittels *Datenbindungen* (engl.: *DataBinding*) an die korrespondierenden VMs gekoppelt. Dadurch ist ersichtlich, dass eine *Benutzerschnittstelle* (*View*) eine visuelle Repräsentation der VMs ist.

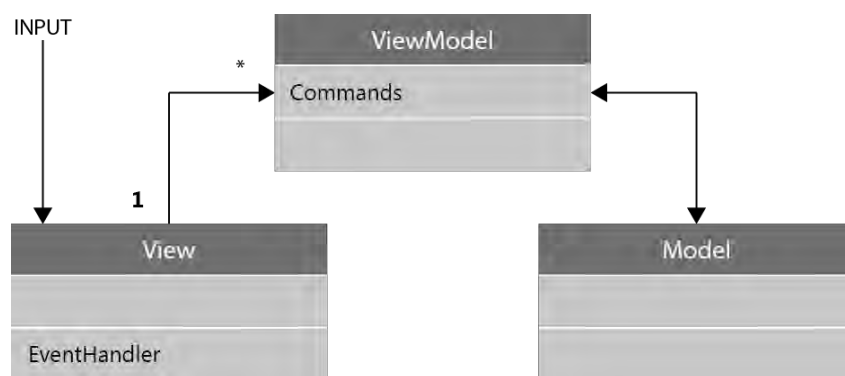


Abbildung 7-2: Prinzip des MVVM-Patterns

Durch eine Datenbindung wird eine Verbindung der Geschäftslogik (d.h. VM und Model) und der Benutzeroberfläche hergestellt (d.h. View) [59]. Durch die Datenbindung ändern sich die an die Daten gebundenen Steuerelemente automatisch

bei jeder Änderung des Werts der Daten. Bei einer Änderung der Werte durch die Steuerelemente, werden die zugrunde liegenden Daten automatisch aktualisiert. Beispielsweise ändert ein Benutzer den Inhalt eines `TextBox`-Steuerelements und die Änderung wird automatisch in der darunterliegenden Variablen aktualisiert. Eine Aktualisierung der Darstellung des Wertes durch ein Steuerelement erfolgt ähnlich dem Observer-Pattern [58]. So schreibt sich das Steuerelement in den `PropertyChangedEventHandler`, welcher Änderungen von Variablenwerten allen eingeschriebenen Objekten mitteilt, des darunterliegenden VMs ein und wird über ein `PropertyChangedEvent` auf die Änderung aufmerksam gemacht.

Da das VM und Model auf keine spezielle View angewiesen sind um ausgeführt zu werden, können auch unterschiedliche Views oder *UnitTests* an ein VM gekoppelt werden [52]. Die beiden Anwendungen des Gesamtsystems (Windows-Store-App und PixelSense-Anwendung) bilden hierbei eine eigene View, welche an die gleichen VMs gekoppelt sind. Die VMs und Models müssen folglich auf beiden Plattenformen ausgeführt werden können, um eine doppelte und somit redundante Implementierung zu verhindern. Dies kann mittels dem Konzept der *Portable Class Library* umgesetzt werden und wird im folgenden Kapitel 7.1.2 erläutert.

7.1.2 PORTABLE CLASS LIBRARY

Für das VM und Model (d.h. die *Geschäftslogik*) wird ein eigenes Software-Projekt erstellt und in einer eigenen Datei als *Portable Class Library* (PCL) ausgeliefert [57]. Eine PCL ist eine dynamische Programm-Bibliothek (DLL), welche auf unterschiedlichen Plattformen ausgeführt werden kann. Dafür werden nur Pakete, Klassen und Methoden verwendet, welche von den beiden Plattformen Windows8 und Windows7, ausgeführt werden können. Somit muss die Logik des Programms nur einmal implementiert werden und unterschiedliche Plattformen verwenden dieselbe Codebasis. Dies erleichtert zusätzlich den Datenaustausch zwischen den Plattformen, da die Daten auf beiden Seiten mittels denselben Klassen serialisiert bzw. deserialisiert werden können (siehe Kapitel 7.3).

7.2 UMSETZUNG DES PROTOTYPEN

Dieses Kapitel beschreibt den generellen Aufbau der Anwendungen anhand des MVVM Design-Patterns. In Kapitel 7.2.1 und Kapitel 7.2.2 wird die konkrete Implementierung der gemeinsamen Datenbasis beschrieben (d.h. Model und VM). Kapitel 7.2.3 und Kapitel 7.2.4 zeigen die konkrete Implementierung der jeweiligen Benutzeroberfläche. Beide Benutzeroberflächen bauen auf einer gemeinsamen Datenbasis auf und bieten eine plattformspezifische, visuelle Repräsentation der Codebasis.

7.2.1 MODEL-KLASSEN

Die Basis aller Interaktionselemente, welche sich auf der Benutzeroberfläche befinden, bildet die abstrakte Klasse `ElementBase`. Diese Klasse implementiert den `PropertyChangedEventHandler`, um die darüber liegenden `ViewModel`-Klassen auf Änderungen der Werte der Datenbasis aufmerksam zu machen. Grundlegende Eigenschaften wie z.B. die *Position*, die *minimale*, *maximale* und *aktuelle Höhe* und *Breite* des Interaktionselements und die *Beschriftung* werden in dieser Klasse zusammengefasst. Es werden Methoden angeboten, um den Layouting Vorgang anzustoßen, rudimentäre Syntaxüberprüfungen bzgl. BPMN durchzuführen und um Prozesselemente miteinander zu verbinden.

Von der Basisklasse `ElementBase` erben alle *Model-Implementierungen*, wie z.B. `ActivityModel`, `ConnectionModel`, `GatewayModel`, `KeyboardModel` und `MainMenuModel`. Auch wenn z.B. ein Hauptmenü nicht mit anderen Steuerelementen verbunden werden kann, reduziert sich die Komplexität der gesamten Anwendung, wenn allen, auf der Benutzeroberfläche angezeigten Interaktionselementen, eine einheitliche Datenbasis zugrunde liegt. Innerhalb dieser *Model-Implementierungen* werden die Regeln der Syntaxüberprüfung für jedes Prozesselement separat vervollständigt und spezifische Ausprägungen dieser implementiert. Zum Beispiel enthält die Klasse `ActivityModel` das Enum `ActivityDimension`, um die Dimension einer Aktivität (keine, Benutzer, Skript, Prozess) zu speichern. Das Enum wird über das zugehörige VM mittels Datenbindung der Benutzeroberfläche zugänglich gemacht, welche je nach Wert der Variable, ein zugehöriges Symbol anzeigt. Ändert sich der Wert dieses Enums, wird das Symbol

automatisch angepasst. Das *Mapping* des Enum-Werts auf ein bestimmtes Symbol wird prinzipiell über `BindingConverter` implementiert, welche Teil der plattformabhängigen `View` sind. Ein `BindingConverter` ist eine Klasse, welche das Interface `IValueConverter` implementiert. Dabei kann für jede Bindung ein eigener `BindingConverter` angegeben werden. Allgemein gesprochen, bildet ein `BindingConverter` einen bestimmten Datentypen A auf einen anderen Datentypen B ab (und umgekehrt) [59].

7.2.2 VIEWMODEL-KLASSEN

Die Klasse `ViewModel` bildet die Basisklasse des gleichnamigen Bereichs des MVVM-Patterns. Diese Klasse referenziert das Model (`ElementBase`), welches der `View` dadurch zugänglich gemacht wird. Benutzeraktionen, welche die `View` empfängt, werden über das *Command Pattern* [56] der Klasse `ViewModel`, und somit auch der Geschäftslogik (d.h. VM und Model) zugänglich gemacht. Hierfür bietet die Klasse `ViewModel` *Commands* an, um auf `TouchUp`- und `TouchDown`-Aktionen des Benutzers reagieren zu können und die Befehle des Kontextmenüs anzunehmen (d.h. *Löschen*, *Duplizieren*, *Subprozess entpacken*, *Subprozess erstellen*, *Tastatur anzeigen*). Änderungen von Eigenschaften der Interaktionselemente, die über Datenbindungen gekoppelt sind, wie z.B. die *Position* oder *Beschriftung*, werden nicht von *Commands*, sondern von *Datenbindungen* geregelt.

Von der Basisklasse `ViewModel` erben alle *spezifischen* VM-Implementierungen, wie z.B. `ActivityViewModel`, `ConnectionViewModel`, `GatewayViewModel`, `KeyboardViewModel` und `MainMenuViewModel`. So implementiert z.B. die Klasse `ActivityViewModel` die Eigenschaft `CornerRadius`, welche den Radius der abgerundeten Ecken des Prozesselements, in Abhängigkeit der Größe des Prozesselements, angibt.

Eine Sonderstellung nimmt das VM `MyScatterViewViewModel` ein, da es das korrespondierende VM der obersten visuellen Ebene der Anwendung ist. Dieses VM hat folgende Aufgaben:

- Speichern aller Instanzen anderer VMs.
- Selektion und Deselektion der Prozesselemente.

- Verschieben von Selektionsgruppen von Prozesselementen.
- Einfügen neuer Prozesselemente zwischen zwei existierende Prozesselemente.
- Erzeugen von Menüs und Tastaturen.
- Aufbau der *horizontalen* Kommunikation zwischen VMs mittels dem *Observer Pattern*.
- Erstellen und Löschen von Interaktionselementen mittels Skizzeneingabe.
- Erstellen von Subprozessen.

Innerhalb der PCL befindet sich der Namespace *Sketch*, welcher die Verarbeitung der Skizzen übernimmt. Die Verfahren für die Skizzenerkennung werden in Kapitel 7.4 aufgezeigt. Da die Klasse `SketchViewModel` *nicht* von der abstrakten Klasse `ViewModel` erbt und auch kein korrespondierendes *Model* aufweist, bildet dieses VM eine Ausnahme. Die Aufgabe der Klasse `SketchViewModel` ist, die einzelnen Punkte des Pfades der Gesteneingabe zu speichern. Dieser Pfad bildet eine Visualisierung ohne Interaktionsmöglichkeiten und wird daher separat behandelt.

Das *Layoutingverfahren* (siehe Kapitel 5.5) wird in der Klasse `LayoutManager` implementiert und ist ebenfalls Teil der PCL, da es auf allen Plattformen zum Einsatz kommt (siehe Kapitel 7.5).

7.2.3 VIEW DES TABLETOP-SYSTEMS

Dieser Abschnitt zeigt die Aufgabe der View und die Implementierung für das Tabletop-System. Da die Konzepte der Implementierung für das Tabletop-System und der Tablets gleich sind, werden die Konzepte in diesem Kapitel erklärt und im nächsten Kapitel 7.2.4 wird auf die spezifischen Merkmale der Tablet-Implementierung eingegangen.

Die *View* bildet das ausführbare Programm der Anwendung und referenziert dabei die gemeinsame *PCL*. Das Gesamtsystem besteht aus zwei eigenständigen Anwendungen, welche ähnlich aufgebaut sind. Neben den teils unterschiedlichen Steuerelementen (vgl. Kapitel 6), unterscheiden sie sich hauptsächlich dadurch, dass sie auf verschiedenen Plattformen aufbauen und folglich verschiedene *Software-Development-Kits* (SDK) verwenden. Der komplette Aufbau der Anwendung ist anhand von Aktivitäten in Abbildung 7-3 dargestellt.

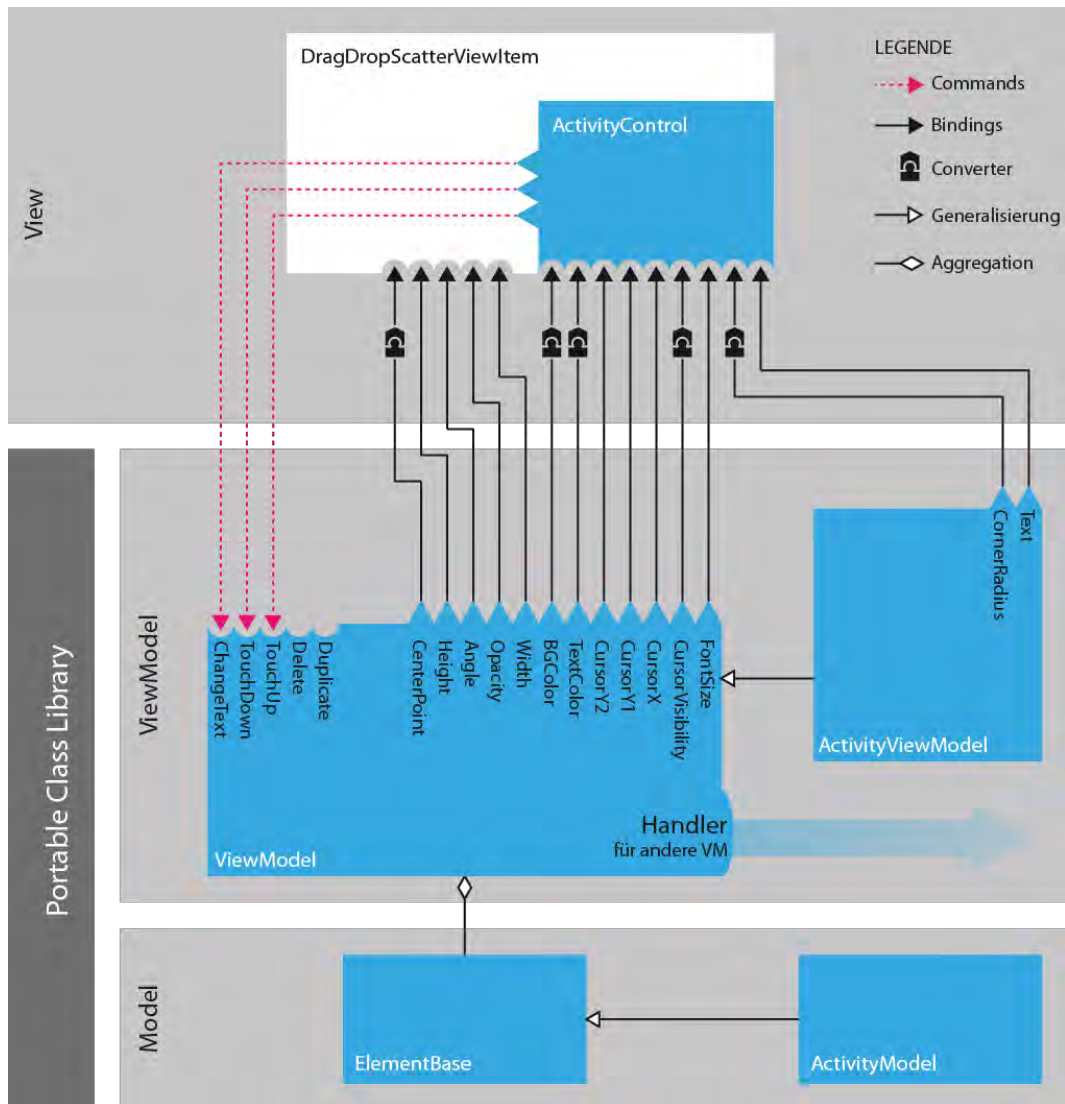


Abbildung 7-3: Aufbau der Anwendung

Die Anwendung für das Tapletop-System (*PixelSenseView*) basiert auf Windows 7 und referenziert das *Surface SDK* von Microsoft [60]. Das Surface SDK bietet Steuerelemente, welche sich für Tapletop-Systeme besonders eignen. Speziell die beiden Steuerelemente `ScatterView` und `ScatterViewItem`. Ein `ScatterViewItem` bildet den Container für Steuerelemente, welche sich dadurch mit einfachen Gesten frei *drehen*, *verschieben* und *skalieren* lassen. Die Klasse `DragDropScatterViewItem` erweitert die Klasse `ScatterViewItem`, indem spezielle Drag&Drop-Aktionen implementiert werden. Innerhalb des Steuerelements `DragDropScatterViewItem` befinden sich *spezifische Steuerelemente*, welche die einzelnen VMs visuell repräsentieren. Für jedes VM existiert ein korrespondierendes *Steuerelement*, wie z.B. `ActivityControl`, `ConnectionPoint`, Gate-

wayControl, Keyboard und MainMenu. Das Steuerelement ActivityControl, das das Steuerelement für Aktivitäten darstellt, ist, wie alle anderen Steuerelemente auch, lose an das VM gekoppelt und kommuniziert (hauptsächlich) über *Commands* mit diesem.

Ein Steuerelement `DragDropScatterViewItem` kann dabei nur das `ScatterView`-Steuerelement als visuelles *Elternelement* besitzen. Das `ScatterView`-Steuerelement dient nur als Container. Um die `ScatterView` mit Verwaltungsaufgaben anzureichern, erweitert die Klasse `SebaScatterView` das Steuerelement `ScatterView`. Die `SebaScatterView` erstreckt sich über das komplette Anwendungsfenster und bildet somit die Grundlage der Anwendung. Als `DataContext` des `SebaScatterView`-Steuerelements wird die Klasse `MyScatterViewViewModel` verwendet.

Die Verbindungspfeile der Prozesselemente dürfen sich nicht innerhalb der `SebaScatterView` befinden, da sie sonst in ein `ScatterViewItem` eingebettet werden müssen. Durch die Einbettung könnte der Benutzer diese Interaktionselemente verschieben und mit ihnen interagieren. Die Verbindungspfeile zeigen jedoch nur Verbindungen an und eine Benutzerinteraktion mit diesen ist nicht möglich. Die Kreise, welche sich in der Mitte eines Verbindungspfeils befinden, können als Interaktionselemente genutzt werden (Steuerelement `ConnectionPoint`) und werden innerhalb des Steuerelements `SebaScatterView` in ein Steuerelement `DragDropScatterViewItem` eingebettet. Um die Pfeile darstellen zu können, wird hinter der transparenten Steuerelement `SebaScatterView` ein `ItemsControl`-Steuerelement definiert, welches sich ebenfalls über das komplette Fenster der Anwendung erstreckt. Die Klasse `MyScatterViewViewModel` ist ebenfalls der `DataContext` des `ItemsControl`-Steuerelements. Somit kann auf die Eigenschaften aller dargestellten Interaktionselemente zugegriffen werden. Über das *Templating-System* von XAML werden die Verbindungspfeile (d.h. ein Strich und ein rotiertes Dreieck als Pfeilspitze) innerhalb eines `Canvas` dargestellt. Programmcode 7-1 zeigt exemplarisch das Template der Verbindungspfeile der Prozesselemente auf, um die verwendeten Konzepte zu veranschaulichen. Die Daten, welche von diesem Template darge-

stellt werden, müssen als DataContext des Elternelements gesetzt sein, um darauf zugreifen zu können. In Zeile 1 wird die `ItemSource` ausgewählt, welche die Quelle der darzustellenden Interaktionselemente bildet. Die Quelle wird von der Liste `Connections` repräsentiert, welche ein Attribut der Klasse `MyScatterViewViewModel` ist. Darin sind die Start- (d.h. `FromX` und `FromY`) und Endkoordinaten (d.h. `ArrowHeadX` und `ArrowHeadY`) der Verbindungspfeile der Prozesselemente gespeichert. Die Koordinaten werden verwendet, um eine Linie zu zeichnen (siehe Zeile 5) und ein rotiertes Dreieck (siehe Zeile 15). Dafür wird in Zeile 5 (und den folgenden) eine *Datenbindung* für die Farbe der Linie, für den Linienstil (d.h. *durchgezogen*, oder *gepunktet*), die Linienbreite und für den Start- und Endpunkt der Linie aufgebaut und teils *BindingConverter* verwendet. Die *BindingConverter* müssen in der *View* definiert werden, da diese für die Übersetzung der abstrakten Parameter auf plattformspezifische Parameter zuständig sind.

```

1  <ItemsControl ItemsSource="{Binding Path=Connections}">
2  <ItemsControl.ItemTemplate>
3  <DataTemplate>
4  <Canvas>
5  <Line Stroke="{Binding StrokeColor,
6      Converter={StaticResource ColorConverter}}"
7      StrokeDashArray="{Binding Strokes,
8      Converter={StaticResource StrokeConverter}}"
9      StrokeThickness="1"
10     X1="{Binding FromX}"
11     Y1="{Binding FromY}"
12     X2="{Binding ArrowHeadX}"
13     Y2="{Binding ArrowHeadY}"
14 </>
15 <Polygon Points="{Binding Arrow,
16     Converter={StaticResource PointConverter}}"
17     Fill="{Binding StrokeColor,
18     Converter={StaticResource ColorConverter}}"
19     Stroke="{Binding StrokeColor,
20     Converter={StaticResource ColorConverter}}"
21     StrokeThickness="1"
22 </>
23 </Canvas>
24 </DataTemplate>
25 </ItemsControl.ItemTemplate>
26 </ItemsControl>

```

Programmcode 7-1: *ItemTemplate* der Verbindungspfeile

Die Anzeige der *Pfade der Gesteneingabe* wird äquivalent zu den Verbindungspfeilen, innerhalb eines eigenen `ItemControl`-Steuerelements im Hauptfenster der Anwendung, implementiert.

7.2.4 VIEW DER WINDOWS-STORE-APP

Dieses Kapitel zeigt die Implementierung der View für Windows-Store-App, welche auf den Tablets zum Einsatz kommt. Da diese ähnlich wie die Implementierung der View Tabletop-Systems ist, wird auf die Unterschiede zwischen der Implementierungen der Anwendung für das Tabletop-System und der Windows-Store-App für die Tablets, eingegangen.

Aus der View der Windows-Store-App [56], kann das *Microsoft Surface SDK* nicht referenziert werden, da dieses Windows 8 nicht unterstützt [60]. Daher werden in der Windows-Store-App die beiden Steuerelemente `SebaScatterView` und `DragDropScatterView` mit ähnlicher Funktionalität programmiert. Durch die Bereitstellung der gleichen Steuerelemente für beide Anwendungen, wurde die Entwicklung deutlich vereinfacht, und analog zu Kapitel 7.2.3 umgesetzt.

Das Hauptmenü der Windows-Store-App wird global am unteren Rand des Displays angezeigt. Hierfür wird im Hauptfenster der Anwendung mittels *XAML* ein `BottomAppBar`-Steuerelement definiert, welche die Funktionalität (d.h. Anzeige mittels Geste, Animationen) übernimmt und mittels *Events* die Benutzeraktionen an die korrespondierende Klasse des Hauptfensters delegiert [56].

Das Kontextmenü wird mithilfe des `Flyout`-Steuerelements implementiert. Hierfür erweitert das Steuerelement `SebaFlyout` das Steuerelement `SettingsFlyout` und zeigt, abhängig des `DataContext` (d.h. aktuelles Prozesselement), spezifische Steuerelemente auf der rechten Seite des Displays an [56]. Die Anzeige mittels einer Wischgeste, das animierte Erscheinen und Verschwinden wird von dieser Klasse übernommen. Da das Menü global angezeigt wird, kann es innerhalb jeder Klasse erstellt werden und bedarf keinem korrespondierenden VM-Objekt.

7.3 KOMMUNIKATION UND DATENAUSTAUSCH

Das Tabletop-System dient für die Kommunikation mit den Tablets als Server. Das bedeutet, dass die Kommunikation von den Tablets initiiert werden muss. Für den Datenaustausch wird die Kommunikationsplattform *Windows Communication Foundation* (WCF) verwendet [61] und im Folgenden detailliert aufgezeigt.

Dabei gibt es drei verschiedene Kommunikationsszenarien zwischen den Tablets und dem Tabletop-System, welche in *Kommunikation 1*, *Kommunikation 2* und *Kommunikation 3* beschrieben werden.

Kommunikation 1 (Tablet tapt auf Prozesselement):

Unabhängig vom aktuellen Zustand des Tablets, wird auf dem Tablet das Prozesselement angezeigt auf das getapt wurde. In der Ansicht können die Eigenschaften dieses Prozesselements geändert werden. Änderungen werden sofort auf die Anwendung des Tabletop-Systems übertragen (siehe Abbildung 6-16-d).

Kommunikation 2 (Tablet tapt auf freie Fläche auf den Hintergrund):

Dieses Kommunikationsszenario hängt vom Zustand des Tablets ab. Es tritt ein, wenn auf dem Tablet Prozesselemente vorhanden sind. Diese werden auf das Tabletop-System übertragen und relativ zum Touchpunkt des Tablets auf dem Tabletop-System positioniert (siehe Abbildung 6-16-b).

Kommunikation 3 (Tablet selektiert mittels einer Lasso-Geste Prozesselemente):

Dieses Kommunikationsszenario hängt ebenfalls vom Zustand des Tablets ab. Befinden sich auf dem Tablet bereits Prozesselemente, wird bei einer Touch-Aktion auf den Hintergrund Kommunikation 2 ausgeführt. Sind keine Prozesselemente auf dem Tablet vorhanden und der Benutzer tapt auf den Hintergrund, kann er mittels einer Lasso-Geste Elemente selektieren, die anschließend auf das Tablet übertragen werden.

Das Konzept von WCF basiert auf einer nachrichtenbasierten Kommunikation und unterstützt dabei verschiedene Transportmechanismen wie z.B. *HTTP* oder *TCP*. Dabei unterscheidet WCF zwischen *Clients*, die die Verbindung initiieren, und *Diensten*, welche auf Anfragen der Clients warten. Die bereitgestellte Kommunika-

tion wird mittels *Verträgen* (engl.: *Contracts*) geregelt. Ein *Contract* ist ein Interface, welches die aufzurufenden *Methoden*, die *Über-* und *Rückgabeparameter* definiert (siehe Programmcode 7-2). Ein Contract kann dabei mehrere Operationen unterstützen (siehe Programmcode 7-2, Zeile 5, Zeile 8, Zeile 11).

```

1  [ServiceContract(CallbackContract= typeof(PixelSenseToTablet))]
2  public interface TabletToPixelSense
3  {
4      [OperationContract]
5      ClickConfirmation ClickDetected
        (
            long timeStamp,
            List<ElementBase> data,
            List<SubprocessTransferHelper> spths
        );
6
7      [OperationContract]
8      long Sync(long currentTicks);
9
10     [OperationContract]
11     bool Update(ElementBase updatedElement);
12 }

```

Programmcode 7-2: Contract TabletToPixelSense

Programmcode 7-3 erstellt, konfiguriert und startet einen WCF-Dienst (d.h. ServiceHost). Der ServiceHost wird in Zeile 1 erstellt, indem der Typ der Klasse übergeben wird, welche den *Contract* implementiert (d.h. TabletRequest). Der zu verwendende *Contract* ist in Zeile 3 angegeben (d.h. TabletToPixelSense). Dort ist auch angegeben, dass die Datentransport mittels TCP vonstatten geht (Zeile 2 bzw. Zeile 3). In Zeile 4 startet der *Dienst*. Die Ausführung dieses Programmcodes erfolgt unmittelbar nach dem Start der Anwendung. Dabei bleibt der Dienst während der gesamten Lebenszeit der Anwendung geöffnet.

```

1  fileServiceHost = new ServiceHost(
        typeof(TabletRequest),
        new Uri[]
        {
            new Uri("net.tcp://localhost")
        }
    );
2  NetTcpBinding tcpBinding = new NetTcpBinding(SecurityMode.None);
3  fileServiceHost.AddServiceEndpoint(
        typeof(PortableViewModelModel.Contracts.TabletToPixelSense),
        tcpBinding,
        "Seba"
    );
4  fileServiceHost.Open();

```

Programmcode 7-3: Erstellen eines ServiceHost

Die Übergabe- und Rückgabeparameter der Methoden des *Contracts*, stellen dabei *serialisierbare* Datentypen dar. Als Standardserialisierungsprogramm verwendet WCF den *Data Contract Serializer* [62].

Die zu übertragenen Objekte (z.B. *ElementBase*) zeichnen sich dadurch aus, dass sie eine Graphstruktur aufweisen. Dies bedeutet, dass jedes Objekt mehrere Nachfolger, aber auch Vorgänger haben kann. Daher muss bei der Übertragung darauf geachtet werden, dass die Objekte mittels *Referenzen* serialisiert werden, um *Zyklen* zu vermeiden. Um Objekte mittels dem *Data Contract Serializer* zu de-/serialisieren, werden die Klassenvariablen mit einem *DataContractAttribute* versehen, um anzugeben, ob und wie eine Klassenvariable serialisiert werden soll.

Abbildung 7-4 zeigt ein Sequenzdiagramm, um die Vorbereitung und Durchführung der Kommunikation zwischen dem Tabletop-System und den Tablets zu veranschaulichen (siehe Kommunikation 1). Auf der linken (Tabletop-Implementierung) und rechten Seite (Tablet-Implementierung) sind jeweils die drei Kommunikationspartner der beiden Anwendungen dargestellt, welche jeweils einen eigenen Thread bilden.

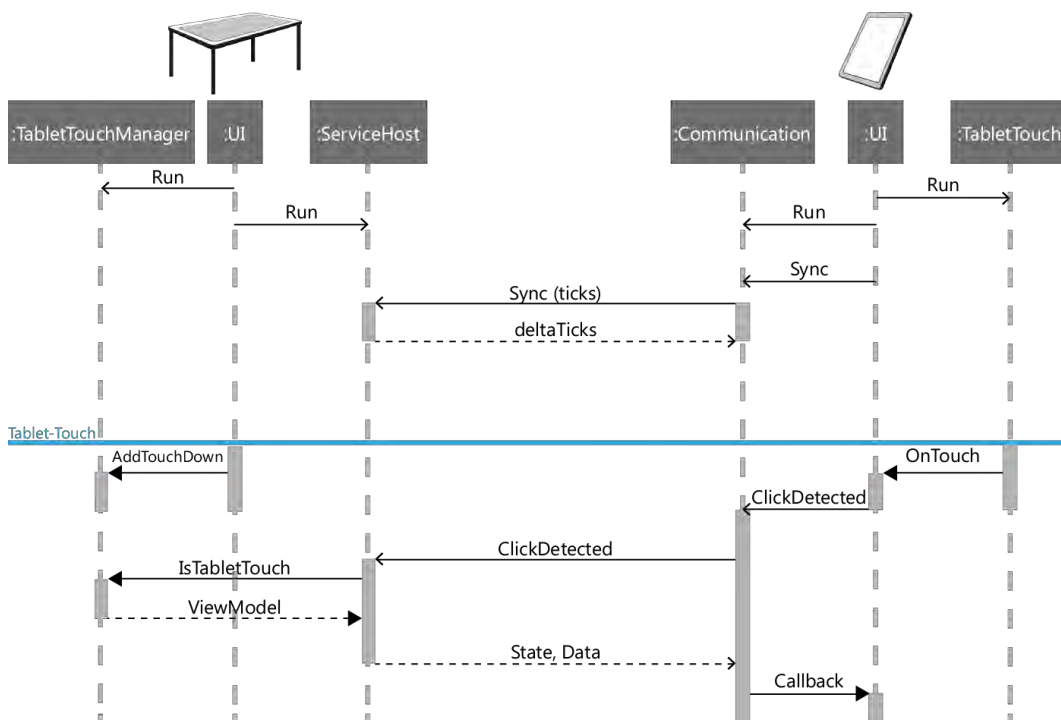


Abbildung 7-4: UML-Sequenzdiagramm – Tablet tapt auf Prozesselement des Tabletops

Beim dem Start der *PixelSenseView*-Anwendung (linke Seite), wird neben dem *UI*-Thread, der *TabletTouchManager*-Thread gestartet. Die Aufgabe des *TabletTouchManager* ist, alle *TouchDown*-Events der Benutzer, zusammen mit dem berührten Element und dem aktuellen Timecode, global zu speichern. Aufgabe des *UI*-Threads ist es, die Benutzeroberfläche darzustellen. Nach dem Programmstart, startet der *UI*-Thread den *ServiceHost*-Thread (siehe Programmcode 7-3).

Auf dem Tablet (siehe Abbildung 7-4, rechte Seite) wird beim Start der Windows-Store-App der *UI*-Thread erzeugt, welcher für die Darstellung der Benutzeroberfläche auf dem Tablet zuständig ist. Der *UI*-Thread erzeugt den *TabletTouch*-Thread sowie den *Communication*-Thread, welcher für die asynchrone Kommunikation mit dem Tabletop-System zuständig ist und nutzt dessen *ServiceHost*-Schnittstelle. Nach der Erstellung schickt der *Communication*-Thread eine *Sync*-Anfrage an das Tabletop-System, um die Uhrzeit zu synchronisieren.

Aufgabe des *TabletTouch-Thread* ist es, die drei Beschleunigungssensoren des Tablets zu überwachen. Führt der Benutzer eine Tablet-Touch-Aktion aus (siehe Abbildung 7-4), befindet sich das Tablet in einer Position, welche eine solche Aktion zulässt. Zusätzlich zeigt der Beschleunigungssensor der Y-Dimension einen *kurzen* und *starken* Ausschlag an. Daraus interpretiert der *TabletTouch*-Thread, dass eine Tablet-Touch-Aktion ausgeführt wurde und folglich wird der *UI*-Thread benachrichtigt. Dieser delegiert die Aktion an den *Communication*-Thread, welcher die *ClickDetected*-Methode des *ServiceHosts* aufruft.

Der Tap des Benutzers mit dem Tablet (siehe Abbildung 7-4, Linie „Tablet-Touch“) wurde vom Tabletop-System als *TouchDownEvent* registriert und folglich vom *TabletTouchManager*-Thread zusammen mit dem aktuellen Timecode gespeichert. Wird nun die *ClickDetected*-Methode des *ServiceHost-Threads* aufgerufen, leitet dieser die Anfrage an die *isTabletTouch*-Methode des *TabletTouchManger*-Threads weiter. Als Übergabeparameter wird der synchronisierte Zeitpunkt mitgeliefert, an welchem das Tablet die Tablet-Touch-Aktion erkannte. Wenn der *TableTouchManger* an einem ähnlichen Zeitpunkt ebenfalls einen Touch registriert hat, gibt die Methode das *VM*-Objekt des Interaktionselements, von dem das *TouchDownEvent* ausgelöst wurde, an den *ServiceHost* zurück.

Dieser überträgt das VM-Objekt zum aufrufenden Tablet und kann vom UI-Thread dargestellt werden.

Das in Kommunikation 3 beschriebene Szenario hebt sich durch die zeitliche Verzögerung des Rückgabewertes deutlich von den anderen Szenarien ab. Löst der Benutzer eine Tablet-Touch-Aktion aus, tauschen beide Geräte Informationen über das Netzwerk aus. Anschließend selektiert der Benutzer mit einer Lasso-Geste die gewünschten Prozesselemente. Dies kann einige Zeit in Anspruch nehmen und unter Umständen zu lange für den Rückruf dauern, da nach einer gewissen Dauer, eine Timeout-Exception geworfen wird. Hebt der Benutzer nun das Tablet vom Tabletop-System (d.h. `TouchUpEvent`), werden die selektierten Prozesselemente vom Tabletop-System zum Tablet übertragen. Somit muss der Tabletop eine Verbindung hin zum Tablet initiieren. Dies ist denkbar, wenn er sich zuvor die IP-Adresse des Tablets merkt. Durch die Verwendung eines WinRT-Tablets ist dies jedoch nicht möglich, da eine Windows-Store-App aus Sicherheitsgründen nicht als „Server“ (d.h. `ServiceHost` in WCF) fungieren kann. Um das beschriebene Interaktionsszenario dennoch anbieten zu können, werden *Callback-Operationen* von WCF verwendet. WCF unterstützt dabei die Funktion, seine Clients kontaktieren zu können. Während einer Callback-Operation werden die Rollen getauscht: Ein Dienst fungiert dabei als Client und der Client wird zum Dienst [63]. Die Callback-Operation wird als eigener *Contract* definiert und aus dem ersten *Contract* referenziert (siehe Programmcode 7-2, Zeile 1). Durch Verwendung der Callback-Operationen kann der Rückruf der Methode beliebig lang nach dem Aufruf dieser stattfinden, um die selektierten Prozesselemente zum Tablet zu transportieren.

7.4 SKIZZENBASIERTE BENUTZEREINGABE

Dieses Kapitel zeigt, wie Skizzeneingaben der Benutzer verarbeitet werden, wenn diese auf den Hintergrund der Anwendung Eingaben tätigen. Skizzeneingaben werden von Process Touch interpretiert, wenn der Benutzer den Finger hebt und dem System somit signalisiert, dass die Eingabe abgeschlossen ist. Die Skizzeneingabe (siehe Abbildung 7-5-a) ist jedoch nicht eindeutig, da der Benutzer auch frei mittels einer Lasso-Geste mehrere Prozesselemente markieren kann (siehe Abbildung 7-5-b). Die Unterscheidung der Aktionen, die Erkennung der Skizzen

und die Zuordnung von Prozesselementen zu einem Selektionspfad, werden in diesem Kapitel aufgezeigt.

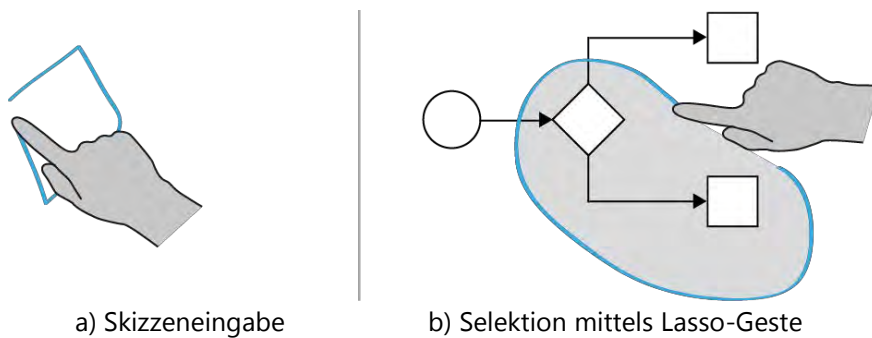


Abbildung 7-5: Dualität der Skizzeneingabe

Da es sich um eine Multi-Touch-Anwendung handelt, können mehrere Eingaben und folglich mehrere Skizzeneingaben zur selben Zeit vorhanden sein. Bei jedem `TouchMoveEvent` wird über die mitgesendete `TouchID` der jeweiligen Berührung ein neuer Touch-Punkt der Auflistung hinzugefügt. Zusätzlich wird für alle Prozesselemente die sich auf der Benutzeroberfläche befinden, überprüft, ob sie sich innerhalb des gezeichneten Pfades befinden. Dieses Kriterium wird für die Unterscheidung der mehrdeutigen Eingabe verwendet. Liegt ein Prozesselement innerhalb des Pfades, handelt es sich bei der Eingabe folglich um eine Selektion und die Fläche des Pfades wird gefüllt, um dem Benutzer mitzuteilen, dass das System eine Selektion erkannt hat. Sonst wird versucht die Eingabe als Skizze zu interpretieren.

Je langsamer der Benutzer die Eingabe tätigt, desto mehr Touch-Punkte werden vom Tabletop-System detektiert und das daraus resultierende Polygon wird durch viele Punkte repräsentiert. Da die Algorithmen als Laufzeitkomplexität sich auf die Anzahl der Punkte beziehen, ist es ratsam diese zu minimieren. Zusätzlich werden die verwendeten Algorithmen dadurch robuster und liefern bessere Ergebnisse.

7.4.1 VEREINFACHUNG EINES POLYGONS

Um unnötige Punkte entlang eines Pfades zu minimieren, wird die Eigenschaft ausgenutzt, dass Punkte bereits in der richtigen Reihenfolge gespeichert sind. Folgendes Verfahren wird angewendet um Punkte zu eliminieren [64]:

Seien Punkt A , B , C aufeinanderfolgend. Der Betrag des Kreuzprodukts $|\overrightarrow{AB} \times \overrightarrow{AC}|$ gibt den Flächeninhalt, des von den Punkten A , B , C aufgespannten Parallelogramms, an. Liegt B auf der Linie zwischen A und C ist der Flächeninhalt 0 und Punkt B kann folglich gelöscht werden. Um den Pfad annähernd zu repräsentieren wird der Flächeninhalt nicht mit 0 verglichen, sondern Punkt B wird gelöscht, wenn der Flächeninhalt kleiner als die Fehlerkonstante EPS ist. Dadurch werden Geraden geglättet, jedoch bleiben Eckpunkte erhalten. Als Algorithmus kann das Verfahren, wie in Programmcode 7-4 gezeigt, formuliert werden.

```

1 void Simplify (List<Point> Path)
2 {
3     //Declaration
4     Point A, B, C;
5     double Xab, Yab, Xac, Yac, cross;
6     double EPS = 20;
7
8     for(int i = 0; i < Path.Count - 2; i++)
9     {
10        //Get the points
11        A = Path[i];
12        B = Path[i+1];
13        C = Path[i+2];
14
15        //Calculate X and Y of the vectors AB and AC
16        Xab = B.X - A.X;
17        Yab = B.Y - A.Y;
18        Xac = C.X - A.X;
19        Yac = C.Y - A.Y;
20
21        //Calculate the cross product
22        cross = Math.Abs(Yab * Xac - Xab * Yac);
23        if(cross < EPS) Path.remove(B);
24    }
25 }

```

Programmcode 7-4: Vereinfachung eines Polygons

In Abbildung 7-6 ist das Ergebnis des Verfahrens exemplarisch aufgeführt. Abbildung 7-6-a zeigt die vom Benutzer skizzierte Form, wobei die einzelnen Eingabepunkte eingezeichnet sind. Dabei wird jeder mit seinem Nachfolger durch eine gerade Linie verbunden. In Abbildung 7-6-b zeigt den Benutzereingabe, nachdem der Pfad vereinfacht wurde, indem unnötige Punkte gelöscht wurden. Dabei ist zu erkennen, dass trotz deutlich weniger Punkte, die Form der Skizzeneingabe erhalten bleibt.

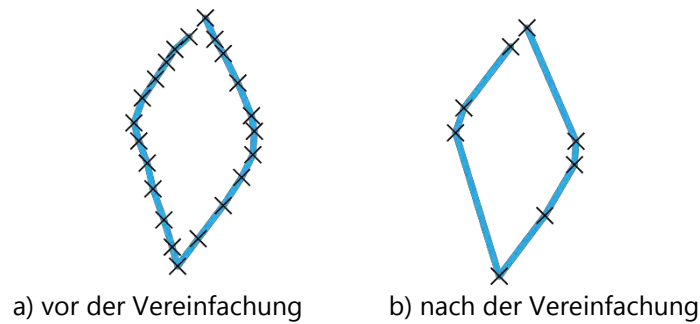


Abbildung 7-6: Vereinfachung eines Polygons

7.4.2 SYMBOLERKENNUNG

Die Erkennung der Symbole ist essentiell für eine skizzenbasierte Eingabe. Dafür gibt es verschiedene Techniken, um Symbole zu erkennen [65]. Der \$1-Recognizer [66] wurde durch seine einfache Implementierung und hohe Genauigkeit populär. Aus diesem und im Folgenden aufgeführten Gründen wird die Diskussion mit diesem Verfahren geführt.

Der \$1-Recognizer erkennt *Single-Stroke-Symbole*, welche aus einem einzelnen Pfad bestehen. Der Vorteil von Single-Stroke-Symbolen besteht darin, dass die Komplexität der Symbole geringer ist als bei *Multi-Stroke-Gesten*, welche aus mehreren Strichen zusammengesetzt werden. Vor allem in einem Multi-User-System es schwer ist, mehrere getrennte Eingaben einer einzelnen Geste zuzuordnen. Dies begründet sich aus der Tatsache, dass mehrere Benutzer gleichzeitig Skizzieren können.

Um ein bestimmtes Symbol zu erkennen, werden vom \$1-Recognizer die Eingaben *vereinfacht*, *skaliert* und *rotiert* und als *unbekannte Vorlage* gespeichert. Die daraus folgende unbekannte Vorlage wird mit zuvor *gespeicherten Vorlagen* verglichen. Dies geschieht, indem die Distanz der korrespondierenden Punkte berechnet wird. Die Genauigkeit des Verfahrens hängt daher von der Anzahl der Vorlagen pro Geste und der Anzahl an Gesten ab. Je mehr Vorlagen geladen werden, desto höher ist die Genauigkeit, aber auch die Laufzeit des Algorithmus [66].

Durch die Skalierung der Eingaben ist es nicht möglich Seitenverhältnisse von Formen zu bestimmen. Folglich können Rechtecke nicht von Quadraten, und Ellipsen nicht von Kreisen unterschieden werden. Weiter ist es nicht möglich eindimensionale Symbole (z.B. Striche) zu erkennen [66]. Es sollen jedoch die Benut-

zereingaben aus Kapitel 5.2 (Kreis, Rechteck, Quadrat, Strich) erkannt werden. Dabei müssen *Striche* erkannt und *Rechtecke* von *Quadraten* unterschieden werden.

Durch die abgeschlossene und überschaubare Menge der angebotenen Skizzen-eingaben, kann ein eigenes Verfahren verwendet werden. Der Vorteil hierbei ist, dass Gesten mittels *Deskriptoren* beschrieben werden können. Dies erhöht die Robustheit und Performanz des Verfahrens.

Ausgehend von dem vereinfachten Polygon wird, ein *Orientierungshistogramm* berechnet. Dies ist ein Array der Länge 180, welches die Winkel von 0 bis 179 repräsentiert. Dazu wird der Winkel zwischen zwei aufeinanderfolgenden Punkten des Polygons berechnet, welcher die Stelle im Array repräsentiert. Der Abstand der beiden Punkte wird auf den Wert der Stelle, des korrespondierenden Winkels, addiert.

Die Stelle des maximalen Betrags im *Orientierungshistogramm* repräsentiert die *Hauptorientierung*. Um jedoch mehr Toleranz und die gesamte Orientierung zu erhalten, wird ein normalisierter Boxfilter der Breite 9 mit den Werten

$$[1, 1, 2, 3, 5, 3, 2, 1, 1] \times \frac{1}{19}$$

verwendet. Dieser berechnet eine gewichtete Summe einer bestimmten Umgebung. Durch die gewählte Gewichtung, wird das Zentrum, im Vergleich zu den Rändern, stärker berücksichtigt. Die Summe der Werte des Boxfilters ergibt genau 1. Dies wird als normalisiert bezeichnet. Folglich wird der gewichtete Mittelwert einer bestimmten Umgebung gebildet. Abbildung 7-7 zeigt beispielhaft die Verwendung eines Boxfilters. Das Maximum der Ausgangsmatrix und der Ergebnismatrix sind hervorgehoben.

Durch die Verwendung eines normalisierten Boxfilters wird mehr Toleranz geschaffen und eine *Region*, im Vergleich zu einer *einzelnen Orientierung*, wird betrachtet.

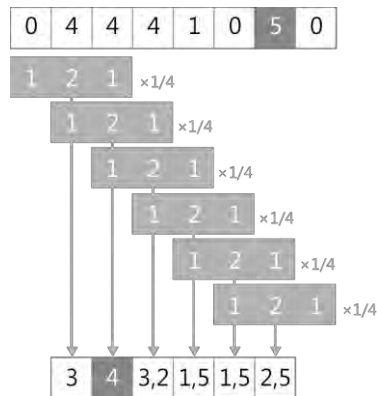


Abbildung 7-7: Normalisierter Boxfilter ohne Randbehandlung

Abbildung 7-8 zeigt zwei Eingaben mit den dazugehörigen Orientierungshistogrammen.

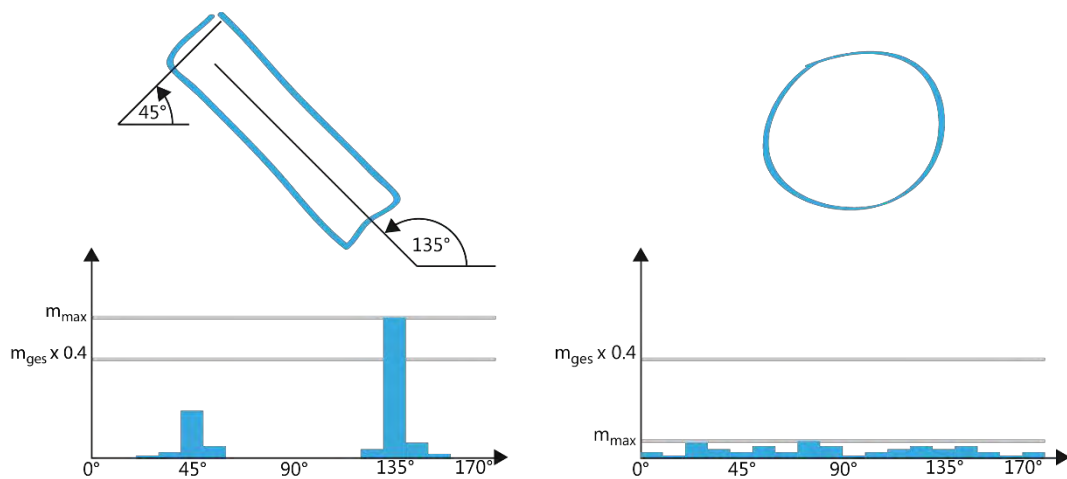


Abbildung 7-8: Orientierungshistogramme samt korrespondierender Skizzeneingabe

Anschließend wird das Polygon um den Betrag der Hauptorientierung, welche mittels dem Boxfilter berechnet wurde, gedreht und die *Höhe* (d.h. BB_{Height}) und *Breite* (d.h. BB_{Width}) der umgebenden *Bounding Box* (d.h. das kleinste umschließende Rechteck) wird berechnet. BB_{Height} und BB_{Width} werden als Merkmale verwendet (siehe Abbildung 7-9).

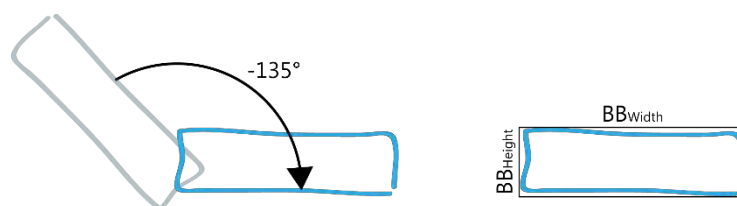


Abbildung 7-9: Drehen des Polygons und Berechnung der Bounding Box

Als zusätzliches Merkmal für die Interpretation der Benutzereingabe eignet sich die *Pfadlänge*. Diese ergibt sich durch die Summierung aller Werte im Orientierungshistogramm. Ein weiteres entscheidendes Merkmal für die Interpretation der Benutzereingabe ist die *Pfadöffnung*. Diese ergibt sich aus dem Abstand zwischen dem ersten und dem letzten Punkt des Polygons.

Anhand des Orientierungshistogramms in Kombination mit den Werten BB_{Height} , BB_{Width} , der *Pfadlänge* und der *Pfadöffnung* lassen sich die in Tabelle 7-1 gezeigten Gesten unterscheiden.

Geste	Beschreibung
<i>Kreis</i>	<p>Befindet sich der maximale Wert des Orientierungshistogramms unterhalb der Schwelle $Pfadlänge \times 0,4$ handelt es sich nicht um ein Rechteck. Folgende Kriterien überprüfen dabei, ob es sich um einen Kreis handelt und ob anschließend die Eingabe als <i>Event</i> interpretiert werden kann:</p> $BB_{Height} \approx BB_{Width}$ $Pfadlänge < \pi \cdot (BB_{Height} + BB_{Width})$ $Pfadöffnung < \text{Min}(BB_{Height}, BB_{Width})$
<i>Strich</i>	<p>Befindet sich der maximale Wert des Orientierungshistogramms über der Schwelle $Pfadlänge \times 0,4$ und gilt zusätzlich die Forderung, dass</p> $Pfadöffnung \approx Pfadlänge$ <p>gilt, so ist die Eingabe als Strich (d.h. <i>Löschgeste</i>) zu interpretiert.</p>
<i>Rechteck</i>	<p>Befindet sich der maximale Wert des Orientierungshistogramms über der Schwelle $Pfadlänge \times 0,4$ und gilt zusätzlich die Forderung, dass</p> $BB_{Height} + BB_{Width} \approx Pfadlänge / 2 - Pfadöffnung$ <p>gilt, so ist die Eingabe ein Rechteck. Dies ist ein Zwischenzustand, von dem ausgehend folgende drei Gesten unterschieden werden:</p>
<i>Gateway</i>	<p>Ist die Eingabe ein <i>Rechteck</i>, und zusätzlich gilt die Eigenschaft</p> $BB_{Height} \approx BB_{Width}$ <p>handelt es sich hierbei um ein Quadrat (d.h. <i>Gateway</i>).</p>

<i>Anmerkung</i>	Ist die Eingabe ein <i>Rechteck</i> , und zusätzlich gilt die Eigenschaft $\text{Pfadöffnung} \approx \text{Max}(BB_{\text{Height}}, BB_{\text{Width}})$ handelt es sich um eine eckige Klammer (d.h. <i>Anmerkung</i>).
<i>Aktivität</i>	Ist die Eingabe ein <i>Rechteck</i> , und zusätzlich gilt die Eigenschaft $\text{Pfadöffnung} < \text{Min}(BB_{\text{Height}}, BB_{\text{Width}}) / 2$ handelt es sich um ein geschlossenes Rechteck (d.h. <i>Aktivität</i>).

Tabelle 7-1: Erkennung von Gesten

7.4.3 MULTISELEKTION VON PROZESSELEMENTEN

Wird eine Gesteneingabe nicht dazu verwendet, Prozesselemente zu erstellen oder zu löschen, sollen die Prozesselemente innerhalb des Eingabepfades, selektiert werden. Um zu erkennen, ob sich ein Prozesselement innerhalb eines Eingabepfades befindet, wird das *Jordan Curve Theorem* in Anlehnung an Frankins [67] Implementierung verwendet. Dieses besagt, dass sich ein Punkt innerhalb eines Polygons befindet, wenn jeder aus dem Punkt ausgehende Strahl die Ränder des Polygons *ungerade* oft schneidet (siehe Abbildung 7-10). Somit liefert das Verfahren auch bei nicht rein konvexen Polygonen eine korrekte Antwort für Regionen, die zwar von einem Polygon teils umschlossen sind, jedoch nicht innerhalb des Polygons sind. Die Komplexität dieses Algorithmus bei einem Polygon mit n Punkten ist $O(n)$.

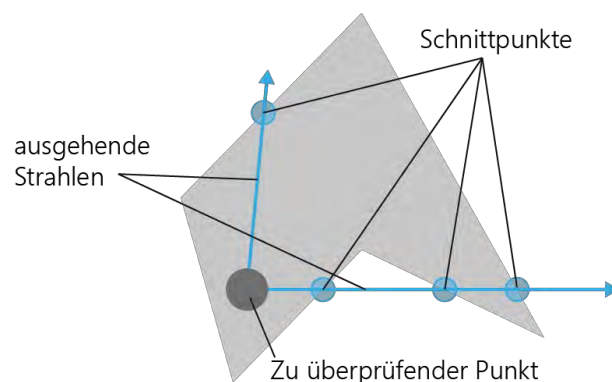


Abbildung 7-10: Jordan Curve Theorem

Als Optimierung wird vor der Berechnung der Schnittpunkte überprüft, ob sich das Prozesselement innerhalb der Bounding Box des Polygons befindet. Die

Bounding Box eines Polygons mit n Punkten kann mit einer Komplexität von $O(n)$ berechnet werden. Jedoch muss diese nur bei einer Änderung der Punkte berechnet werden und nicht für jedes Prozesselement neu. Die Überprüfung selbst braucht wenige Abfragen, mit einer Komplexität $O(1)$. Somit stellt diese Vorgehensweise eine Optimierung dar.

7.5 LAYOUTING DER PROZESSELEMENTE

Für das Layouting der Prozesselemente wird ein *Force-Directed-Graph-Algorithmus* verwendet (vgl. Kapitel 5.5), welcher ein physikalisches System simuliert. Auf jeden *Knoten* (d.h. Prozesselement) wirkt eine *Kraft* F (siehe Abbildung 7-11), welche sich aus verschiedenen Kräften zusammensetzt. Übersteigt der Betrag der Kraft $|F|$ einen bestimmten *Schwellwert* (Haftreibung), bewegt sich der Knoten mit einer bestimmten Geschwindigkeit in Richtung der Kraft.

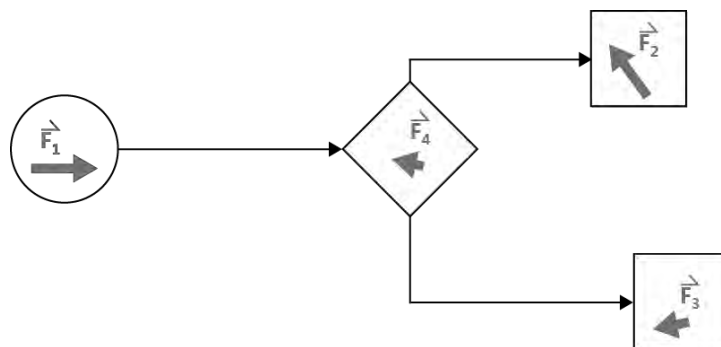


Abbildung 7-11: Kräfte einzelner Knoten

Zum einen wirkt auf jeden Knoten, der mit einem anderen Knoten verbunden ist eine Anziehung in Richtung des verbundenen Knotens mit dem Betrag

$$|F_{Attraction}| = ATTRACTION * (Proximity - SPRINGLENGTH)$$

wobei *ATTRACTION* und *SPRINGLENGTH* Konstanten sind. *Proximity* ist der Abstand zwischen den zwei Knoten. Dies kann mit einer Feder die zwischen den beiden Knoten gespannt ist, veranschaulicht werden (siehe blaue Federn in Abbildung 7-11). Die Anziehungskraft muss für jeden Knoten nur mit den verbundenen Knoten berechnet werden.

Zum anderen Stoßen sich Knoten, ähnlich gleich gepolten Magneten, gegenseitig ab. Die Kraft kann mittels dem Gesetz von *Coulomb* berechnet werden [68]:

$$|F_{Repulsion}| = -REPULSION / Proximity^2$$

REPULSION stellt hierbei eine Konstante dar. *Proximity* ist der Abstand zwischen den zwei Knoten. Die abstoßende Kraft muss für jeden Knoten mit allen anderen Knoten innerhalb eines bestimmten Abstands berechnet werden.

Die resultierende Kraft wird bestimmt, indem alle auf einen Knoten wirkenden Kräfte $F_{Attraction}$ und $F_{Repulsion}$ addiert werden.

Die auf jedes Element wirkenden Kräfte werden parallel von einem eigenen Thread berechnet. Die Klasse `LayoutManager` stellt hierfür die statische Methode `DoLayout(MyScatterViewViewModel svvm)` bereit, welche das übergebene VM der `ScatterView` layoutet. Dadurch, dass die Methode als `static` deklariert wird, kann sie global aufgerufen werden. Das Layouting-Verfahren muss immer dann angestoßen werden, wenn sich die *Größe* oder *Position* eines Prozesselements ändert. Damit immer nur ein *Layouting-Thread* zur gleichen Zeit arbeitet, wurden dieser Methode *exklusive Zutrittsrechte* gegeben. Das Layouting-Verfahren wird solange ausgeführt, solange sich in einem Schleifendurchlauf mindestens ein Prozesselement bewegt.

7.6 ZUSAMMENFASSUNG

Dieses Kapitel gibt tiefe Einblicke in den Aufbau des Proof-of-Concept-Implementierung von Process Touch. Auf die Umsetzung wichtiger Bestandteile wird dabei näher eingegangen.

Die verwendete Softwarearchitektur wird beschrieben, und die Einordnung der einzelnen Klassen in das MVVM-Pattern erläutert. Auf die unterschiedliche Implementierung der Benutzeroberflächen (Views) wird gesondert eingegangen.

Durch die Verwendung einer *Portable Class Library* kann die gesamte Geschäftslogik, die Skizzeneingabe und das Layouting-Verfahren auf beiden Plattformen, ohne redundanten Code, verwendet werden.

Die Verwendung von WCF als Kommunikationsschnittstelle bietet eine einfache und zielgerichtete Integration in das Gesamtprojekt. Die Serialisierung mittels

dem *Data Contract Serializer* ermöglicht eine einfache und zuverlässige Art der De-/Serialisierung.

8 EXPERIMENTELLE VALIDATION

Das in der Arbeit beschriebene Konzept des Multi-User-Software-Systems ist anhand von Forschungsergebnissen anderer Veröffentlichungen und Guidelines entwickelt. Dieses Vorgehen kann als *analytische Methode* im Sinne der Usability-Evaluation angesehen werden [69]. Zur Validation des Konzepts, wird dieses im Folgenden *empirisch* evaluiert, indem ein Experiment mit Probanden durchgeführt wird. Dieses Kapitel beschreibt die Definition des Experiments, den detaillierten Ablauf und die Auswertung der Daten.

Kapitel 8.1 gibt einen Überblick über das Experiment, stellt die Forschungsfrage auf und leitet daraus Hypothesen ab. Um diese zu prüfen, wird ein Experiment definiert und im Folgenden diskutiert. In Kapitel 8.2 wird die Vorbereitung und Durchführung des Experiments betrachtet und die Validation der Daten aufgezeigt. Die gewonnenen Daten werden letztlich in Kapitel 8.3 analysiert und interpretiert. Kapitel 8.4 wertet das Experiment qualitativ aus und zeigt mögliche Verbesserungen des Systems auf.

8.1 FORSCHUNGSFRAGE UND EXPERIMENTDEFINITION

Kollaborative Geschäftsprozessmodellierung hat im Vergleich zur traditionellen Geschäftsprozessmodellierung viele Vorteile. Wissensaustausch findet z.B. zwischen den beteiligten Benutzern während der Modellierung statt, Ideen können einfach geteilt und diskutiert werden und das Verständnis des Prozessmodells steigt bei den Beteiligten [7, 70]. Durch Software-Unterstützung steigt zusätzlich die Qualität von Prozessmodellen, die kollaborativ erstellt werden [8]. Diese Arbeit und insbesondere die Validation kommt der Frage nach, wie ein Software-System aufgebaut sein muss, um von diesen Vorteilen profitieren zu können. Dazu wird ein *deduktiver* Usability-Test in einem kontrollierten Experiment mit *Between-Subjects-Design* angewandt [69]. Im Detail bedeutet das, dass das Software-System in verschiedenen Ausprägungen existiert und von verschiedenen Gruppen von Probanden getestet und bewertet wird.

8.1.1 KONTEXTAUSWAHL UND ZIELDEFINITION

Das in dieser Arbeit erstellte Multi-User-Software-System stützt sich auf aktuelle Erkenntnisse der HCI und kommt insbesondere der Forderung nach Tablets als private Displays nach. Dies führt zu der Forschungsfrage:

Kann durch die Kombination von Tabletop- und Tablet-Geräten eine bessere Prozessmodellqualität bei der kollaborativen Prozessmodellierung erreicht werden?

Aus dieser Fragestellung heraus wird ein Experiment durchgeführt, um die Zufriedenheit der Benutzer mit Konzept, sowie die Qualität der Prozessmodelle zu messen.

Aus der Forschungsfrage heraus lässt sich folgende Zieldefinition definieren [71]:

Analysiere das Konzept zur kollaborativen Prozessmodellierung
zum Zweck der Evaluation
in Bezug auf die Prozessmodellqualität bei kollaborativer Prozessmodellierung mittels einer Kombination aus Tabletop-Systemen und Tablets
aus der Sicht eines Forschers
im Kontext einer *Multi-Object Variation Study* [71] an der Universität Ulm

8.1.2 HYPOTHESENDEFINITION

Basierend auf dem Ziel des Experiments, werden im Folgenden entsprechende Hypothesen abgeleitet. Das Experiment untersucht zum einen den Einfluss der Verwendung von Tablets zusätzlich zum Tabletop-System auf die Qualität der Prozessmodelle, zum anderen den Einfluss auf die Benutzerfreundlichkeit des Systems, welche wiederum Auswirkungen auf die Qualität haben kann. Die Qualität der Prozessmodelle lässt sich in die Bereiche *Granularität*, *syntaktische*, *semantische* und *empfundene Qualität* aufteilen [72]. Folglich lassen sich fünf Hypothesen ableiten (siehe Hypothese 1- Hypothese 5).

Hypothese 1 (Granularität):

Erhöht die zusätzliche Verwendung eines Tablets bei kollaborativer Prozessmodellierung an Tabletop-Systemen die Granularität eines Prozessmodells?

H_{0,1}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat keinen signifikanten Einfluss auf die *Granularität* des damit erstellten Prozessmodells.

H_{1,1}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat signifikanten Einfluss auf die *Granularität* des damit erstellten Prozessmodells.

Hypothese 2 (Syntaktische Qualität):

Erhöht die zusätzliche Verwendung eines Tablets bei kollaborativer Prozessmodellierung an Tabletop-Systemen, die syntaktische Qualität eines Prozessmodells?

H_{0,2}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat keinen signifikanten Einfluss auf die *syntaktische Qualität* des damit erstellten Prozessmodells.

H_{1,2}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat signifikanten Einfluss auf die *syntaktische Qualität* des damit erstellten Prozessmodells.

Hypothese 3 (Semantische Qualität):

Erhöht die zusätzliche Verwendung eines Tablets bei kollaborativer Prozessmodellierung an Tabletop-Systemen, die semantische Qualität eines Prozessmodells?

H_{0,3}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat keinen signifikanten Einfluss auf die *semantische Qualität* des damit erstellten Prozessmodells.

H_{1,3}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat signifikanten Einfluss auf die *semantische Qualität* des damit erstellten Prozessmodells.

Hypothese 4 (Empfundene Qualität):

Erhöht die zusätzliche Verwendung eines Tablets bei kollaborativer Prozessmodellierung an Tabletop-Systemen, die von Probanden empfundene Qualität eines Prozessmodells?

H_{0,4}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat keinen signifikanten Einfluss auf die von den Probanden *empfundene Qualität* des damit erstellten Prozessmodells.

H_{1,4}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat signifikanten Einfluss auf die von den Probanden *empfundene Qualität* des damit erstellten Prozessmodells.

Hypothese 5 (Benutzerfreundlichkeit):

Erhöht die zusätzliche Verwendung eines Tablets bei kollaborativer Prozessmodellierung an Tabletop-Systemen, die Benutzerfreundlichkeit des Systems?

H_{0,5}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat keinen signifikanten Einfluss auf die *Benutzerfreundlichkeit* des in dieser Arbeit vorgestellten Konzepts.

H_{1,5}: Ein Tablet, welches in Kombination mit einem Tabletop-System verwendet wird, hat signifikanten Einfluss auf die *Benutzerfreundlichkeit* des in dieser Arbeit vorgestellten Konzepts.

8.1.3 AUFBAU DES EXPERIMENTS

Dieser Abschnitt beschreibt im Detail den Aufbau des Experiments durch die Definition der *Probanden*, des *Faktors*, der *Objekte* und der *Zielvariablen*.

Probanden: Als Probanden dienen Studenten und Doktoranden der Informatik und Medieninformatik. Alle Probanden haben mindestens Grundkenntnisse in der Prozessmodellierung und in BPMN. Die Probanden nehmen freiwillig an dem Experiment teil. Dabei werden sie zufällig in Zweiergruppen zusammengestellt um gemeinschaftlich ein vorgegebenes Szenario zu modellieren.

Objekte: Die Objekte sind zum einen die von den Probanden erstellten Prozessmodelle in BPMN, zum anderen Log-Dateien, Fragebögen und Audio- und Vide-

oaufzeichnungen. Die Prozessmodelle und Fragebögen dienen dazu, die Hypothesen zu testen. Log-Dateien, Audio- und Videoaufzeichnungen dienen, zusätzlich zu den Fragebögen, dazu, im Anschluss an das Experiment Verbesserungen am Konzept erarbeiten zu können.

Um sicherzustellen, dass alle Probanden auf dem gleichen Wissensstand bezüglich des zu modellierenden Szenarios sind, wird eine textuelle Szenarienbeschreibung ausgeteilt. Beide Probanden, die gemeinsam das Szenario bearbeiten, bekommen jeweils eine Beschreibung des zu modellierenden Szenarios ausgehändigt, das jeweils eine unterschiedliche Perspektive auf dasselbe Szenario beschreibt. Dies begründet sich aus der Tatsache, dass das zu testende Konzept explizit für die *kollaborative* Prozessmodellierung konzipiert wurde und Szenarien dem typischen Aufgabenkontext der zukünftigen Endbenutzer entsprechen sollen [69]. Daraus ergibt sich die Notwendigkeit, die Modellierung in Zweiergruppen von Probanden durchzuführen. Der unterschiedliche Wissensstand bzw. die unterschiedlichen Perspektiven auf ein und denselben Geschäftsprozess, werden durch die verschiedenen Beschreibungen des Szenarios simuliert. Im Folgenden sind beide Perspektiven des Szenarios dargestellt (siehe Szenario 1 und Szenario 2). Die den Probanden textuell vorgelegte Szenarienbeschreibung befindet sich in Anhang F

Szenario 1 (Perspektive des Bewerbers):

Ein Bewerber stellt seine Bewerbungsunterlagen zusammen. Hierfür schreibt er einen Lebenslauf sowie ein entsprechendes Anschreiben. Anschließend schickt er die Bewerbungsunterlagen an das Unternehmen.

Als Antwort bekommt er von dem Unternehmen entweder eine Absage oder eine Einladung zum Vorstellungsgespräch. Im Falle einer Einladung, wird das Vorstellungsgespräch geführt.

Im Anschluss an das Vorstellungsgespräch erhält der Bewerber wiederum eine Antwort welche eine Absage, oder einen Gehaltsvorschlag enthält.

Im Falle des Gehaltsvorschlags, prüft er diesen und nimmt ihn an, wenn er seinen Vorstellungen entspricht. Sonst teilt er dem Unternehmen mit, dass der Gehaltsvorschlag zu niedrig ist. Folglich bekommt er einen neuen Vorschlag von dem Unternehmen. Dies wiederholt sich so lange, bis der Bewerber mit dem Gehaltsvorschlag einverstanden ist, oder er eine Absage des Unternehmens bekommt.

Im Falle einer Einigung bekommt er den Arbeitsvertrag zugeschickt, welchen er ausfüllt und zurücksendet.

Szenario 2 (Perspektive des Unternehmens):

Das Unternehmen bekommt eine Bewerbung samt Bewerbungsunterlagen zugeschickt. Nach einer Überprüfung und Bewertung der Bewerbung wird entweder eine Absage oder eine Einladung zum Vorstellungsgespräch als Antwort verschickt.

Nach dem Vorstellungsgespräch beraten sich alle Beteiligten in dem Unternehmen und das Unternehmen schickt das entsprechende Resultat an den Bewerber. Diese Antwort kann eine Absage oder einen Gehaltsvorschlag enthalten.

Nimmt der Bewerber diese nicht an, wird über einen höheren Gehaltsvorschlag beraten. Nachdem das Unternehmen eine Zusage bekommen hat, verschickt es einen entsprechenden Arbeitsvertrag.

Ist der Vertrag ausgefüllt eingegangen, wird der Bewerber eingestellt. Hierfür muss die Personalabteilung informiert werden, die Daten des Arbeitsvertrags in die Mitarbeiterdatenbank eingetragen werden und ein Arbeitsplatz für den neuen Mitarbeiter eingerichtet werden.

Faktor: Der Faktor ist die Verwendung von Tablets als privates Interaktions-Gerät, welches zusammen mit dem Tabletop-System zur kollaborativen Prozessmodellierung verwendet wird. Die Faktorstufen sind dabei *verwenden* und *nicht verwenden*.

Zielvariable: Als Zielvariable betrachten wir die Angemessenheit der Verwendung von Tablets, zusätzlich zum Tabletop-System. Da sich diese nicht direkt messen lässt, wurde bereits definiert, dass das Konzept adäquat oder angemessen ist, wenn eine höhere Usability des Systems und eine höhere Qualität der Prozessmodelle gemessen werden kann.

Der Grad der Usability wird mittels zwei verschiedener Fragebögen ermittelt, die nachfolgende vorgestellt werden.

Um eine normgerechte Messung nach DIN EN ISO 9241, welche die maßgebliche Normenreihe für eine Gestaltung von Systemen mit hoher Usability [69] darstellt, zu gewährleisten, wird der Fragebogen ISO-Norm 9241/10 verwendet [73] (siehe Anhang A). Dieser bezieht sich nicht auf konkrete Funktionen oder Handlungsfol-

gen und hat daher nur Hinweischarakter. Der Fragebogen zeigt nicht auf, *wie* und an *welchen Stellen* Mängel behoben werden können [69]. Der Fragebogen besteht aus den Kategorien *Aufgabenangemessenheit*, *Selbstbeschreibungsfähigkeit*, *Steuerbarkeit*, *Erwartungskonformität*, *Fehlertoleranz*, *Individualisierbarkeit* und *Lernförderlichkeit*, welche die Faktoren der *User Perceived Quality* der DIN EN ISO 9241/110 bilden [74]. Dieser Fragebogen stellt einen reflektierten Ansatz der Bewertung dar. Die Probanden werden dabei aufgefordert „den Beurteilungsbogen äußerst sorgfältig“ auszufüllen. Ein Auszug aus der Kategorie Aufgabenangemessenheit ist in Abbildung 8-1 zu sehen.

<i>Das System...</i>	--	--	-	-/+	+	++	+++	<i>Das System...</i>
ist kompliziert zu bedienen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist unkompliziert zu bedienen.
bietet nicht alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bietet alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.
bietet schlechte Möglichkeiten, sich häufig wiederholende Bearbeitungsvorgänge zu automatisieren.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bietet gute Möglichkeiten, sich häufig wiederholende Bearbeitungsvorgänge zu automatisieren.
erfordert überflüssige Eingaben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erfordert keine überflüssigen Eingaben.

Abbildung 8-1: Beispiel-Items des ISO-Norm 9241/10

Ergänzend hierzu wird der Fragebogen *User Experience Questionnaire (UEQ)* verwendet (siehe Anhang B). Dieser misst den Gesamteindruck der Probanden in Bezug auf das System [75]. Er besteht aus 26 bipolaren Adjektiven mit jeweils einer 7-Punkt-Likert-Skala. Die Probanden werden angehalten möglichst spontan zu antworten, damit die unmittelbare Einschätzung der Probanden zum Tragen kommt [75]. Der UEQ besteht aus den Kategorien *Attraktivität*, *Durchschaubarkeit*, *Effizienz*, *Steuerbarkeit*, *Stimulation* und *Originalität*. Abbildung 8-2 zeigt einen Auszug aus dem UEQ.

	1	2	3	4	5	6	7	
unerfreulich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erfreulich
unverständlich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	verständlich
kreativ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	phantasielos
leicht zu lernen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	schwer zu lernen
wertvoll	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	minderwertig
langweilig	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	spannend
uninteressant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	interessant

Abbildung 8-2: Beispiel-Items des UEQ

Der Grad der Usability leitet sich aus den Medianen und Standardabweichungen der einzelnen Kategorien beider Fragebögen ab.

Um unter anderem die Homogenität der Probandengruppe zu prüfen, wird ein selbst erstellter *Demographie-Fragebogen* verwendet, welcher am Ende der Evaluation von den Probanden ausgefüllt wird (siehe Anhang C). Dieser beinhaltet die im folgenden Absatz definierten Fragen zur *empfundene Prozessmodellqualität*, Fragen bezüglich der *Prozessmodellierungs-Erfahrung*, *persönliche Angaben* (Alter, Geschlecht, Tätigkeit, Bildung) und die Möglichkeit für Probanden, textuelles *Feedback* als Freitext zu geben.

Die Qualität der Prozessmodelle wird durch die drei Dimensionen *syntaktische*, *semantische* und *empfundene Qualität* charakterisiert [72]. Die *syntaktische Qualität* wird gemessen, indem syntaktische Fehler (z.B. syntaktische Regelverletzungen) der Prozessmodellierungssprache des Prozessmodells (z.B. BPMN) gezählt werden. Die *semantische Qualität* besteht aus den in Tabelle 8-1 aufgeführten Dimensionen.

Dimension	Beschreibung
<i>Korrektheit</i>	Alle Elemente eines Prozessmodells sind <i>korrekt</i> und <i>relevant</i> .
<i>Vollständigkeit</i>	Keine relevanten Aspekte der Domäne (Szenario) fehlen.
<i>Relevanz</i>	Alle Elemente im Prozessmodell sind für den Prozess relevant.
<i>Authentizität</i>	Die gegebene Repräsentation spiegelt einen korrekten Eindruck der Domäne wider.

Tabelle 8-1: Dimensionen semantischer Qualität

Die vier Unterdimensionen der semantischen Qualität werden durch zwei Modellierungsexperten unabhängig voneinander bewertet. Jede Dimension wird auf einer 7-Punkt-Likert-Skala einzeln bewertet und dient der Konsensbildung. Die Skala geht von 1 (starke Ablehnung) bis 7 (starke Zustimmung).

Die *empfundene Qualität* basiert auf dem Grad der Zustimmung des Probanden mit dem Prozessmodell [76]. Hierfür bewerten Probanden unabhängig voneinander nach der Modellierung das Prozessmodell in den von Tabelle 8-2 gezeigten Dimensionen.

Dimension	Beschreibung
<i>Zustimmung</i>	Die <i>Zustimmung</i> beschreibt den Grad, indem das Prozessmodell mit dem Geschäftsprozess der realen Welt übereinstimmt.
<i>Fehlende Aspekte</i>	Die <i>Fehlenden Aspekte</i> beschreiben signifikante Aspekte, welche in dem Prozessmodell fehlen.
<i>Akkurate Beschreibung</i>	Die <i>Akkurate Beschreibung</i> misst den Grad der Genauigkeit der Übereinstimmung zwischen dem Prozessmodell und dem Geschäftsprozess der realen Welt
<i>Fehler</i>	Die Dimension <i>Fehler</i> beschreibt den Grad der ernsthaften Fehler in dem Prozessmodell.
<i>Zufriedenheit</i>	Die <i>Zufriedenheit</i> gibt den Grad der Zufriedenheit mit dem Prozessmodell wieder.

Tabelle 8-2: Kategorien empfundener Qualität

Diese Kategorien werden jeweils mittels einer 5-Punkt-Likert-Skala bewertet. Die Skala geht von 1 (starke Ablehnung) bis 5 (starke Zustimmung).

Die *Granularität* eines Prozessmodells wird mittels der Komplexität des Prozessmodells gemessen (d.h. Anzahl an Aktivitäten, Kanten, Gateways und Ausführungspfade) [77].

Für die *qualitative Evaluation* in Hinblick auf mögliche Verbesserungen des in dieser Arbeit vorgestellten Systems, werden die Audio- und Videoaufzeichnungen und die Log-Dateien herangezogen. Die Log-Dateien zeichnen das Nutzungsverhalten der Probanden während der Prozessmodellierung auf. Die Log-Dateien bestehen aus folgenden Typen von Einträgen:

1. Anzahl an Prozesselementen, welche mittels *Gesten erzeugt* wurden.
2. Anzahl an Prozesselementen, welche mittels *Menüs erzeugt* wurden.
3. Anzahl an Prozesselementen, welche *mittels Gesten gelöscht* wurden.
4. Anzahl an Prozesselementen, welche *mittels Menüs gelöscht* wurden.
5. Anzahl an *Tablet-Touch-Aktionen*.
6. Anzahl an Buchstaben, welche mit der *Tabletop-Tastatur* geschrieben wurden.
7. Anzahl an Buchstaben, welche mit der *Tablet-Tastatur* geschrieben wurden.
8. Anzahl an Prozesselementen auf dem Tablet, bevor diese auf den Tabletop übertragen wurden.
9. Anzahl an Prozesselementen, welche auf dem *Tablet erzeugt* wurden.
10. Anzahl an Prozesselementen, welche auf dem *Tabletop erzeugt* wurden.
11. Anzahl an Verbindungs-Aktionen von Prozesselementen, die auf dem *Tablet* durchgeführt wurden.
12. Anzahl an Verbindungs-Aktionen von Prozesselementen, die auf dem *Tabletop* durchgeführt wurden.

Da bei dem Faktorlevel *nicht-verwenden* einige Einträge der Log-Dateien nicht aussagekräftig sind, können dabei nur die ersten vier Dimensionen ausgewertet werden. Da diese Auswertung rein *qualitative Ziele* verfolgt, stellt dies keine Einschränkung des Experiments dar.

8.1.4 GESTALTUNG DES EXPERIMENTS

Um der Forderung nachzukommen, dass alle gesammelten Daten aus einer unabhängigen Zufallsvariable (Faktor) kommen, verwenden wir den Ansatz der *Randomisierung*. Hierfür werden die Zweiergruppen der Probanden abwechselnd einem Faktorlevel (d.h. verwenden bzw. nicht verwenden) zugeordnet. Die davon abhängigen Variablen sind *Modellqualität* und *Usability*. Diese werden von der Hypothese abgeleitet und von den Objekten repräsentiert [71]. So stellt der Aufbau ein randomisiertes, balanciertes Single-Faktor-Experiment dar. Dabei handelt es sich um ein Single-Faktor-Experiment, da nur ein Faktor (d.h. der Einsatz von Tablets) variiert. Abbildung 8-3 gibt einen Überblick über den Aufbau des Experiments.

Messtechnik und Datenerhebungsmethode: Für die Durchführung des Experiments wird als Tabletop-System der Samsung SUR40 mit Microsoft PixelSense verwendet (ehemals Microsoft Surface 2 bezeichnet [78]). Für die Probanden, die dem Faktorlevel *verwenden* zugeordnet werden, liegt jeweils ein Tablet bereit. Hierbei handelt es sich um ein Microsoft Surface RT der ersten Version, welches ohne externe Tastatur (d.h. TouchCover, bzw. TypeCover) oder Maus verwendet wird.

Auf den Geräten wird das in dieser Arbeit entwickelte System verwendet. Weiter werden Eingaben und Aktionen vom System aufgezeichnet und in Log-Dateien gespeichert.

Für Audio- und Videoaufzeichnungen wird eine Sony HDR-CX570 Kamera samt Stativ verwendet. Der Ton wird mit dem internen Mikrofon aufgenommen.

8.1.5 VALIDITÄT DES EXPERIMENTS

Jedes Experiment birgt Risiken, welche das Resultat beeinflussen bzw. unbrauchbar machen können. Die Validität kann dabei in zwei Dimensionen unterteilt werden. Die *interne Validität* geht der Frage nach, ob die beobachteten Effekte nur aus der Veränderung der unabhängigen Variable abzuleiten sind. Die *externe Validität* betrachtet, ob die Resultate sich generalisieren bzw. auf andere Gruppen übertragen lassen.

Interne Validität: Um sicherzustellen, dass Unterschiede in der Messung nur auf die Veränderung der Kontrollvariable zurückzuführen ist, kontrollieren wir potentielle Einflussfaktoren:

- Ein Ungleichgewicht der zwei Gruppen der Faktorlevel wird durch abwechselnde Zuweisung der Zweiergruppen der Probanden vermieden.
- Probanden werden zufällig in Zweiergruppen eingeteilt, um einer Auswahlverzerrung entgegenzuwirken.
- Um einen Wissenstransfer ausschließen zu können, bearbeiten Probanden nur ein Szenario.
- Damit alle Probanden gleiches Vorwissen bzgl. des Systems mitbringen, hat kein Proband das System im Vorfeld gesehen, ausprobiert bzw. das Konzept gehört.

- Durch die Vorlage des Szenarios (d.h. Domänen-Wissen) wird bei allen Probanden bzw. Teams das gleiche Vorwissen bzgl. der Domäne sichergestellt.
- Aufgrund der starken Marktdurchdringung von Smartphones und Tablets kann ein unterschiedlicher Erfahrungsstand mit Touch-Systemen vernachlässigt werden. Weiter sind alle Probanden Studenten bzw. Doktoranden der Informatik und bringen Standesgemäß eine gewisse Technikaffinität mit. Somit kann davon ausgegangen werden, dass alle Probanden ausreichend Erfahrung mit Touch-Systemen mitbringen.
- Mögliche unterschiedliche Erfahrung mit Prozessmodellierung wird dadurch ausgeschlossen, dass nur Probanden am Experiment teilnehmen, die mindestens Grundkenntnisse in der Prozessmodellierung haben.
- Um negative Einflüsse aufgrund von Müdigkeit oder Hunger auszuschließen, wird das Experiment zu einer Tageszeit durchgeführt, wo dies ausgeschlossen werden kann.
- Das Experiment wird in einem ruhigen, abgeschlossenen Raum durchgeführt, um Ablenkungen der Probanden ausschließen zu können.
- Die erwartete Dauer des Experiments pro Gruppe ist ca. 40 min. Dies schließt negative Auswirkungen aufgrund eines Mangels an Motivation oder Konzentration aus.
- Alle Probanden nehmen freiwillig an dem Experiment teil.

Externe Validität: Alle Probanden haben einen akademischen Hintergrund, was die Generalisierbarkeit einschränkt. Jedoch weisen alle Probanden grundlegende Kenntnisse in der Prozessmodellierung auf. Da sich Modelle von Studenten und Modellierungsexperten nicht unterscheiden lassen [79], können die Probanden dieses Experiments als Stellvertreter für Modellierungsexperten angesehen werden, welche ein grundlegendes Training in der Prozessmodellierung bekommen haben. Die verwendete Hard-/Software und die Implementierung des Konzepts kann die Generalisierbarkeit einschränken.

Jedoch wurde durch das Experiment-Design festgelegt, dass beide Gruppen mit demselben Tabletop-System arbeiten und es nur einen Faktor (d.h. den Einsatz von Tablets) gibt. Somit ist es wichtig auf diesen Faktor zu achten. Bei dem Micro-

soft Surface RT handelt es sich um ein leichtes, gut verarbeitetes Tablet. Durch den Verzicht auf externe Tastaturen, welche für dieses Tablet verfügbar sind, kann die Hardware als neutral betrachtet werden, da keine besonders negativen oder positiven Eigenarten vorhanden sind.

8.2 ABLAUF DES EXPERIMENTS

Dieser Abschnitt basiert auf der Experimentdefinition (siehe Kapitel 8.1) und beschreibt detailliert den vollständigen Ablauf des Experiments.

8.2.1 VORBEREITUNG

Probanden werden für das Experiment eingeladen. Die Probanden werden nicht über die zu untersuchenden Aspekte des Experiments informiert, wissen jedoch, dass das Experiment Teil einer Masterarbeit ist. Allen Probanden wird Anonymität gewährleistet. Vor dem eigentlichen Experiment wird mittels einer Vorstudie geprüft, dass die Hard- und Software zuverlässig läuft, alle technischen Geräte (Kamera, Mikrofon) funktionieren und die Anweisungen und Fragen für die Probanden verständlich sind.

8.2.2 DURCHFÜHRUNG

Das Experiment wird in einem ruhigen Raum der für die Zeit des Experiments reserviert wird, an der Universität Ulm durchgeführt. Es kann immer nur eine Zweiergruppe von Probanden gleichzeitig am Experiment teilnehmen. Das Experiment wird innerhalb von zwei Wochen durchgeführt, wobei jede Sitzung zwischen 60 und 80 Minuten dauert und folgendermaßen abläuft:

Dem Versuchsleiter liegt ein Leitfaden vor (siehe Anhang D), anhand diesem er das Experiment leitet. So gibt er zu Beginn eine kurze Einführung in das System und teilt den Probanden die Szenarienbeschreibung aus. Nachdem diese von den beiden Probanden unabhängig gelesen wurde, bearbeiten diese kollaborativ das Szenario mithilfe des Tabletops und je nach Faktorlevel auch mithilfe des Tablets.

Nach Beendigung des Szenarios teilt der Versuchsleiter die Fragebögen aus. Um die Emotionen und spontanen Gedanken der Probanden festzuhalten füllen die Probanden erst den UEQ und anschließend den ISO-Norm 9241/10 aus. Zuletzt

füllen die Probanden den Demographie-Fragebogen aus. Abbildung 8-3 stellt den Ablauf graphisch dar.

Während der ganzen Zeit werden Audio- und Videoaufzeichnungen erhoben und ausgewählte Aktionen (siehe Kapitel 8.1.4) werden vom System während der Bearbeitung der Szenarien in einer Log-Datei gespeichert.

Am Ende haben Probanden die Möglichkeit auf dem Fragebogen Feedback zu hinterlassen.

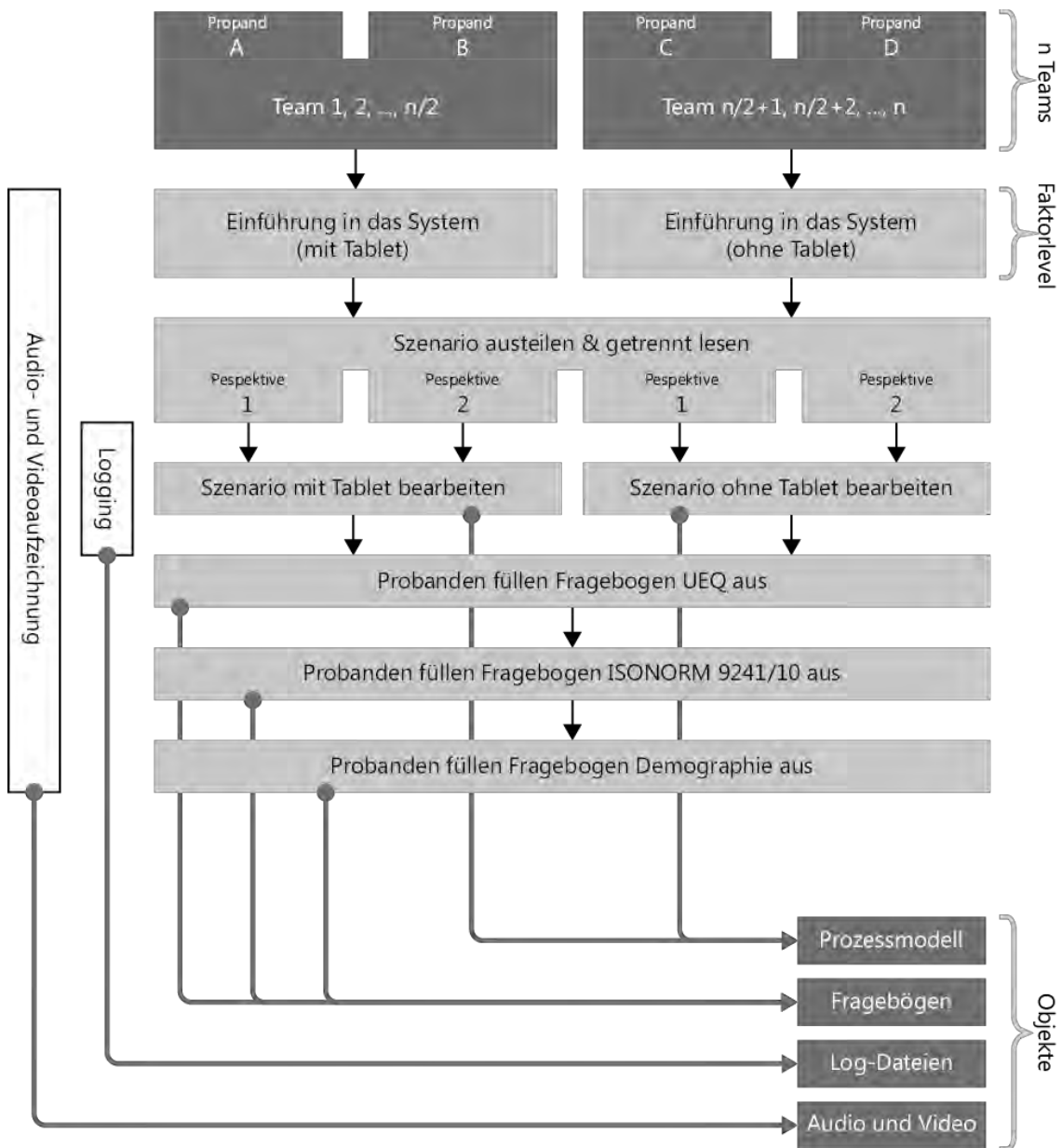


Abbildung 8-3: Ablauf des Experiments

8.2.3 VALIDATION DER DATEN

Insgesamt nahmen 20 Probanden an dem Experiment teil. Alle Durchgänge liefen den Vorstellungen nach ab und somit konnten die Daten aller Probanden verwendet werden. 10 Probanden sind Studenten und 10 Probanden Doktoranden. Von den 20 Probanden sind 3 weiblich und 17 männlich. Die Probanden wurden nach ihrer Erfahrung (Analysieren, Modellieren, Auskennen, System Beherrschen) mit BPMN gefragt, da dies eine Voraussetzung für die Teilnahme an dem Experiment ist. Auf einer Likert-Skala von 1 bis 7 beträgt der Median der Mediane 5,5 mit einer Standardabweichung von 1,6. In den letzten 12 Monaten modellierten die Probanden durchschnittlich 25,6 Modelle und haben im Schnitt 3,5 Jahre Erfahrung mit BPMN. Folglich erfüllen alle Probanden die Voraussetzungen und passen in die gesuchte Probandengruppe. Kritisch zu betrachten sind die α -Fehler der Unterschiede der Probanden, welche einen anderen Faktorlevel zugewiesen bekamen. Diese entstanden durch die zufällige Zuweisung der Probanden in einer der beiden Gruppen. So ist die Gruppe mit dem Faktorlevel *mit Tablet* etwas erfahrener (Median 0,75 Punkte) und haben längere Erfahrung mit BPMN (durchschnittlich 2,1 Jahre).

8.3 DATENANALYSE UND INTERPRETATION

In diesem Abschnitt werden die gesammelten Daten ausgewertet und interpretiert (siehe Kapitel 8.3.1) und die Hypothesen getestet (siehe Kapitel 8.3.2). Die von den Probanden erstellten Prozessmodelle sind in Anhang G abgebildet, die Rohdaten des Experiments sind in Anhang H zu finden.

8.3.1 DATENANALYSE UND BESCHREIBENDE STATISTIK

Abbildung 8-4 zeigt Box Plots (*Median, Minimum und Maximum, als auch das erste und dritte Quartil*) der sieben Kategorien des ISO-Norm 9241/10 Fragebogens, welche die Benutzerfreundlichkeit repräsentieren. Hierbei sind die beiden Faktorlevel *mit Tablet* und *ohne Tablet* gegenübergestellt.

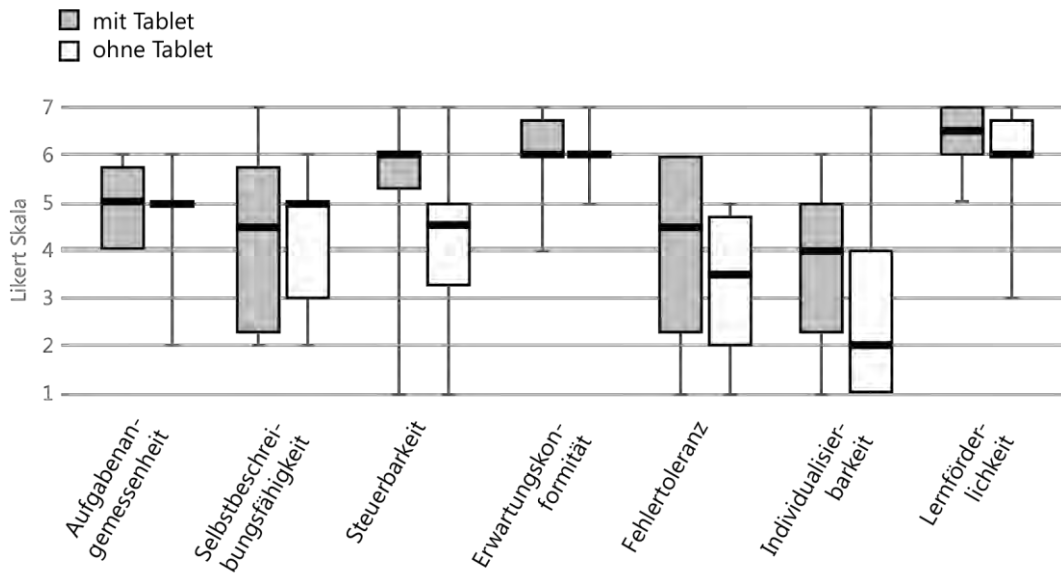


Abbildung 8-4: Box Plots der Kategorien des ISO-Norm 9241/10 Fragebogens

Die Kategorien des ISO-Norm 9241/10 Fragebogens werden im Median um 0,5 Punkte (auf einer Likert-Skala von 1 bis 7) besser Bewertet, wenn der Faktorlevel *mit Tablets* angewendet wird (siehe Abbildung 8-4). Im Besonderen ist die Kategorie *Steuerbarkeit* um 2 Punkte besser, die Kategorie *Selbstbeschreibungsfähigkeit* um 1,5 Punkte und die Kategorie *Fehlertolleranz* um 1 Punkt besser bewertet, wenn die Probanden zusätzlich mit Tablets arbeiteten.

Abbildung 8-5 zeigt die Box Plots (*Median, Minimum* und *Maximum*, als auch das *erste* und *dritte Quartil*) des UEQ. Durchschnittlich sind die einzelnen Dimensionen des Fragebogens gleich gut bewertet. Unterschiede innerhalb der einzelnen Dimensionen weisen keine deutliche Signifikanz auf. Somit kann zusammengefasst werden, dass tendenziell beide Varianten des getesteten Systems eine ähnliche *User Experience* bieten.

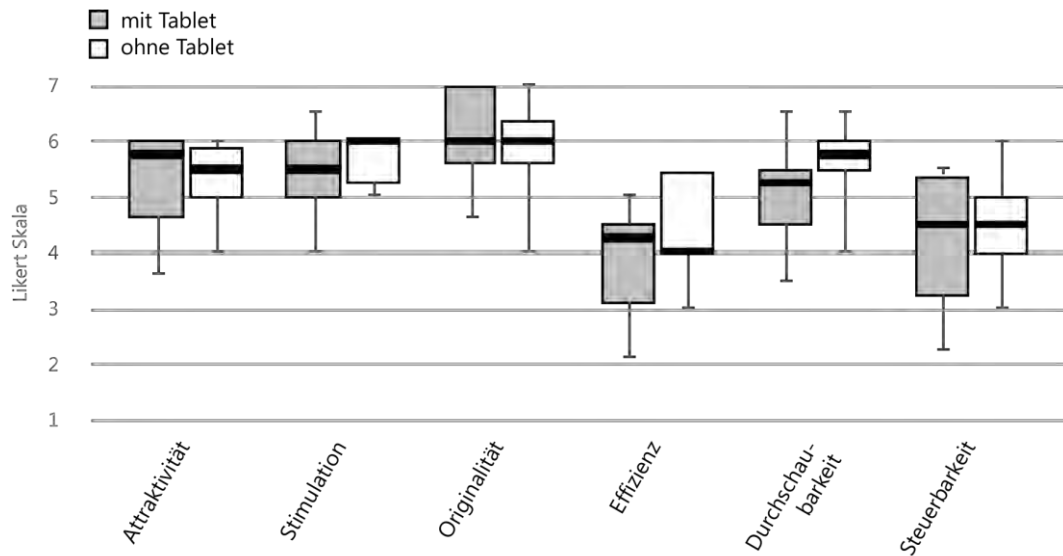


Abbildung 8-5: Box Plots der Dimensionen des UEQ Fragebogens

In Abbildung 8-6 sind die Werte des Loggers als Box Plots (*Median, Minimum* und *Maximum*, als auch das *erste* und *dritte Quartil*) zusammengefasst dargestellt. Darin ist zu sehen, dass bei dem Faktorlevel *mit Tablets*, im Median 28 Prozesselemente weniger erstellt, aber auch 25 Prozesselemente weniger gelöscht werden. Somit kann davon ausgegangen werden, dass die Modellierung von Prozessmodellen mit Tablets tendenziell präziser oder konzentrierter abläuft. Weiter sind die Probanden des Faktorlevels *mit Tablet* im Durchschnitt über 7 Minuten schneller als die Probanden, welche den Faktorlevel *kein Tablet* zugewiesen bekommen. Die Anzahl der getippten Buchstaben unterscheidet sich hingegen bei unterschiedlichen Faktorlevel nicht.

Zusammenfassend kann eine Tendenz hin zur besseren Usability mit Tablets ausgemacht werden, jedoch ist diese nicht signifikant.

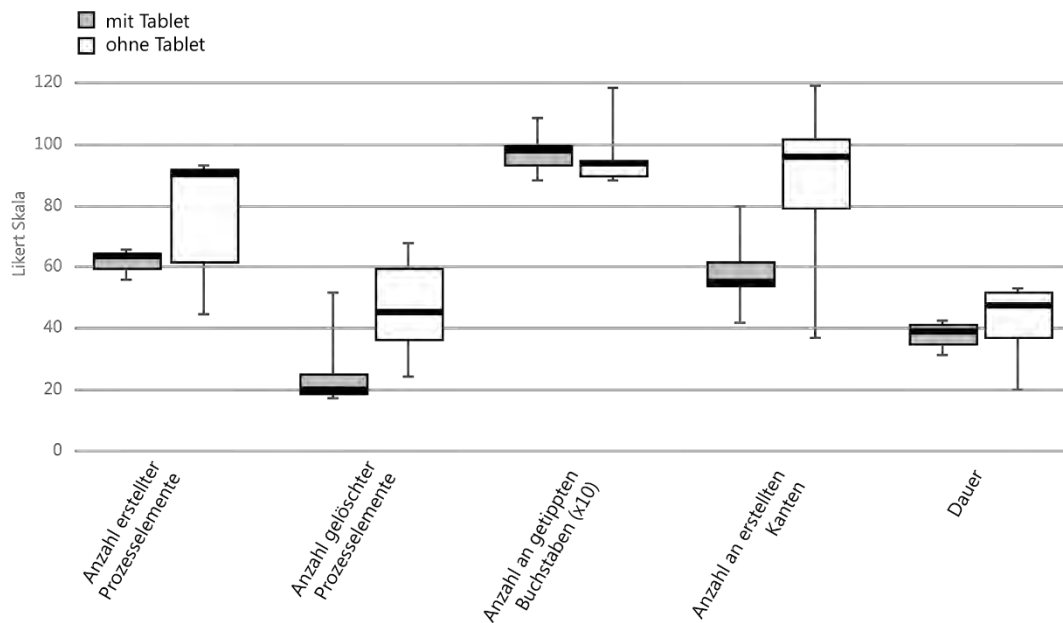


Abbildung 8-6: Box Plots der zusammengefassten Werte des Loggers

8.3.2 HYPOTHESENTEST

Aufgrund der geringen Probandenzahl von 20, wird an dieser Stelle auf einen formellen Hypothesentest verzichtet, da er nicht aussagekräftig ist. Auf Tendenzen soll in diesem Abschnitt dennoch hingewiesen werden.

Die *semantische* und *empfundene Qualität* der Prozessmodelle, bei unterschiedlichen Faktorlevel, zeigen keine Unterschiede auf. Die *syntaktische Qualität* der Prozessmodelle ist bei dem Faktorlevel *mit Tablets* durchschnittlich leicht höher (1,6 Fehler weniger). Durch die implementierte Fehlerprüfung konnten Probanden an vielen Stellen die Syntax jedoch auch nicht verletzen. Als Signifikant kann der Unterschied der syntaktischen Qualität jedoch nicht angesehen werden. Die *Granularität* ist bei beiden Faktorlevels gleich. Die *Benutzerfreundlichkeit* wurde bereits in Kapitel 8.3.1 diskutiert und lässt auch keine deutlichen Unterschiede erkennen. Somit kann keine der fünf Hypothesen, welche in Kapitel 8.1.2 definiert wurden, zurückgewiesen werden.

8.4 QUALITATIVE AUSWERTUNG

Durch die *Beobachtung* des Experiments, die *Audio-* und *Videoaufzeichnungen* und das *Feedback* der Probanden, welches teils Verbal während dem Experiment als auch textuell auf den Fragebögen gegeben wurde, kann eine *qualitative Aus-*

wertung des Systems erfolgen. Die folgenden Aussagen lassen sich nicht anhand der gemessenen Daten belegen, haben ihren Ursprung jedoch in dem Experiment. Die Erkenntnisse können als Verbesserungsvorschläge für zukünftige Systeme verwendet werden.

8.4.1 TABLETOP-SYSTEM

Das verwendete Tabletop-System Sur40 von Samsung mit Microsofts Pixel Sense Technologie zeigt einige Schwachstellen. So muss der Raum komplett abgedunkelt werden, um die Touch-Erkennung, welche mit Infrarotlicht arbeitet, vor Sonnenlicht zu schützen. Dies schränkte in dem Experiment die Lesbarkeit der gedruckten Szenarien ein, welche die Probanden bekommen. Durch die geringe Höhe von 83cm werden Benutzer beeinflusst die Haltung der Hand bei Touch-Aktionen in einer Weise anzupassen, dass die Finger meist ausgestreckt sind und die Erkennung der Touch-Aktion besser funktioniert. Der Nachteil der geringen Höhe ist, dass Probanden über Rückenschmerzen klagten, teils in die Hocke gingen, oder sich an einer Stuhllehne abstützten. Somit kann verallgemeinert werden, dass langes Stehen und Arbeiten an einem niedrigen Tabletop-System körperlich anstrengend ist.

In vielen Fällen standen Probanden sich an der langen Tischkante gegenüber (siehe auch [30]). Da Prozessmodelle *gerichtet* sind, wurde der Sequenzfluss typischerweise auf der linken Seite begonnen und breitete sich nach rechts hin aus. Modellieren Benutzer kollaborativ unterschiedliche Sichten auf denselben Prozess und stehen sich dabei gegenüber, starten die Teilbereiche auf unterschiedlichen Seiten. Abbildung 8-7 zeigt dies exemplarisch auf. Eine Verbindung (z.B. mit Datenelementen) kreuzt dabei die komplette Arbeitsfläche.

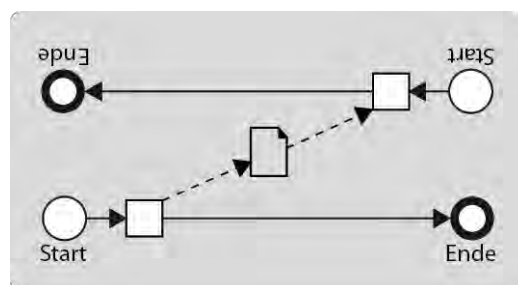


Abbildung 8-7 Modellierung an gegenüberliegenden Seiten des Tabletop-Systems

Da sich BPMN als Standard durchgesetzt hat, wird auch in dem System, welches in dieser Arbeit vorgestellt wird, diese Notationssprache verwendet. Es wurde in dem Experiment beobachtet, dass durch die freie Orientierung der Elemente, XOR- und AND-Gateways öfters *verwechselt* wurden. Die Ähnlichkeit der Symbole der beiden Prozesselemente in rotiertem Zustand wird in Abbildung 8-8 deutlich aufgezeigt.

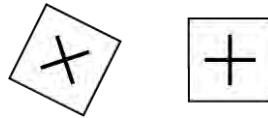


Abbildung 8-8 Ähnlichkeit von AND und XOR bei freier Orientierung

Daraus folgt, dass die Ähnlichkeit reduziert werden muss. Eine Lösung wird in Verbesserung 1 aufgezeigt.

Verbesserung 1 (Ähnlichkeit der Symbolik reduzieren):

Um der Verwechslung frei rotierter Prozesselemente entgegenzuwirken, muss die Ähnlichkeit der Symbolik von AND- und XOR-Elementen reduziert werden. So kann die Symbolik der *Logik* herangezogen werden. Ein AND-Gatter wird mit einem &-Symbol, ein OR-Gatter mit ≥ 1 und ein XOR-Gatter mit $=1$ beschriftet.

8.4.2 GESTEN

Prozesselemente werden dreimal so häufig mit Gesten erstellt als mittels dem Menü. Dies zeigt, dass die Gesten gut gewählt sind und Benutzer gerne mit Skizzen arbeiten. Die Lösch-Geste wurde sogar zehnmal häufiger verwendet als die Löschoperation mittels dem Menü. Dies mag auch an der Einfachheit der Lösch-Geste liegen. Diese Einfachheit wird jedoch bei vielen Probanden zum Problem, da sie versehentlich Prozesselemente löschen. Folglich ergibt sich die Forderung nach einer komplexeren Gestaltung der Lösch-Geste um versehentliche Löschoperationen zu minimieren (siehe Verbesserung 2).

Verbesserung 2 (Erhöhung der Komplexität der Lösch-Geste):

Um versehentliche Löschoptionen zu minimieren, muss die Komplexität der Lösch-Geste erhöht werden. Verschiedene Ansätze sind dazu in der Literatur zu finden [4, 43, 32, 66].

Die Gesten für ein Gateway (*Quadrat*) und eine Aktivität (*Rechteck*) sind einigen Probanden zu ähnlich, bzw. kann beobachtet werden, dass Rechtecke, welche fast Quadrate waren, von Process Touch als Gateway interpretiert werden (siehe Verbesserung 3).

Verbesserung 3 (Anpassung der Grenzwerte für Quadrate):

Rechtecke, die von Benutzern skizziert wurden, wurden falsch kategorisiert. Die Grenzwerte für die Unterscheidung von *Rechteck* und *Quadrat* müssen besser angepasst werden.

8.4.3 LAYOUTING VON PROZESSMODELLEN

Das Layouting-Verfahren (siehe Kapitel 5.5) wird teils als störend empfunden. So schreibt ein Proband, dass die *automatische Anpassung [...] einen Tick zu stark* sei. Einige Probanden fordern eine Funktion, welche das Prozessmodell sauber layoutet, oder dass die Ausrichtung der Prozesselemente auf einem Raster basiert. Diese Forderungen stehen allerdings im Widerspruch zu einem *natürlichen* Interface. Die unterschiedliche Wahrnehmung und Wirkung auf Probanden zeigt jedoch, dass an dieser Stelle Potential für Verbesserungen vorhanden ist (siehe Verbesserung 4).

Verbesserung 4 (Layouting-Verfahren):

Benutzern ist es wichtig, dass das resultierende Prozessmodell übersichtlich dargestellt wird. Der verwendete Layouting-Algorithmus erfüllt hierbei nicht alle Bedürfnisse der Benutzer. Durch widersprüchliche Forderungen (*NUI* vs. *Raster*; *Funktion für sauberes Layout* vs. *Keine Globalen Aktionen* (Anforderung 15)) kann keine konkrete Verbesserung aufgezeigt werden. Jedoch kann beobachtet werden, dass das implementierte Verfahren nicht voll zufriedenstellend ist.

Das Layouting-Verfahren arbeitet in der Weise, dass zwei Prozesselemente die nicht miteinander verbunden werden können, sich gegenseitig abstoßen, auch wenn der Benutzer ein Prozesselement gerade verschiebt und auf ein anderes legen will. Der Hintergrund hierfür ist, dass es auch bei Verschiebe-Operationen zu keiner Verdeckung kommt. Es wird jedoch beobachtet, dass es für Probanden nicht immer nachvollziehbar ist, wann Prozesselemente verbunden werden können und wann nicht. Die Verdrängung von Prozesselement, mit denen Probanden ein anderes Prozesselement verbinden wollten, führt zu Frustration. Aus dieser Beobachtung heraus folgt Verbesserung 5.

Verbesserung 5 (Klares Feedback):

Benutzer brauchen ein klares Feedback seitens des Systems, wenn Aktionen nicht durchführbar sind. Verschiebt ein Benutzer ein Prozesselement, darf kein anderes von diesem abgestoßen werden, auch wenn die beiden nicht miteinander verbunden werden können. Um den Grundsatz der Fehlervermeidung beizubehalten, ist es ratsam die Prozesselemente z.B. in der Farbe Rot einzufärben wenn eine Verbindungsaktion nicht möglich ist, um dies dem Benutzer zu verdeutlichen.

8.4.4 MENÜS

Bei der Erstellung, des in dieser Arbeit vorgestellten Systems, wird davon ausgegangen, dass Menüs nur geöffnet werden, wenn Benutzer sie auch verwenden wollen. Durch eine komplexe Geste (Pinch-Geste) wird eine versehentliche Öff-

nung ausgeschlossen. Sollte dies dennoch vorkommen, schließen sich Menüs nach einer bestimmten Zeit automatisch. Es wird jedoch beobachtet, dass Probanden nicht verwendete Menüs auch selbst schließen wollen. Die automatische Schließung der Menüs wurde weder positiv hervorgehoben, noch negativ bemängelt (siehe Verbesserung 6).

Verbesserung 6 (Menüs Schließen):

Benutzer wollen, *zusätzlich* zur automatischen Schließung der Menüs, die Möglichkeit haben, Menüs manuell zu schließen.

Eine Möglichkeit, diese Verbesserung umzusetzen ist, einen Schließen-Button im Hauptmenü mittig anzuordnen. Aus dem Mittelpunkt heraus entsteht das Menü durch eine Animation und lässt sich folglich auch an dieser Stelle wieder schließen (siehe Abbildung 8-9).



Abbildung 8-9: Verbessertes Menü mit Schließen-Button

8.4.5 VOLLSTÄNDIGKEIT VON BPMN

Die vorhandenen Prozesselemente stellen nur einen Teil der BPMN-Spezifikation dar. Aufgrund zeitlicher Probleme werden nicht mehr Prozesselemente implementiert. Einige Probanden kritisieren vor allem das Fehlen von *Messages*, *Pools* und *Lanes*.

8.4.6 IMPLEMENTIERUNGSFEHLER

Trotz vorhandener Fehler im System, musste dieses jedoch nicht neu gestartet werden und Probanden konnten im Experiment die Arbeit zu jederzeit fortsetzen. Da alle Probandengruppen der beiden Faktorlevel *mit Tablet* und *ohne Tablet* dasselbe System verwendeten, haben die Fehler eventuell das Ergebnis *absolut* beeinflusst, jedoch nicht *relativ* auf die Unterschiede der beiden Gruppen bezo-

gen. Tabelle 8-3 geht auf die im System vorhandenen Fehler bezüglich der Implementierung ein.

Fehler	Beschreibung
<i>Menü öffnen</i>	Wird auf dem Tabletop ein Touch erkannt, kann das Menü nicht mittels einer Pinch-Geste geöffnet werden.
<i>Fehlende Kanten bei Subprozess entpacken</i>	Wird ein Subprozess ausgepackt (engl.: <i>unwrap</i>) wird das Start- und Ende-Prozesselement des Subprozesses nicht automatisch mit den umliegenden Elementen verbunden.
<i>Fehlende Kanten bei Undo</i>	Wird auf dem Tablet die rückgängig-Funktion (engl.: <i>undo</i>) verwendet, erscheint zwar das gelöschte Prozesselement wieder, jedoch die Verbindungen dieses Prozesselements nicht.
<i>Zu lange Antwortzeiten</i>	Bei komplexen Berechnungen bezüglich des Layouts, kann es vor allem auf den Tablets zu Verzögerungen kommen.
<i>Zurück-Taste</i>	Wird die Zurück-Taste der Tastatur länger gedrückt, erwarten Benutzer, dass Buchstaben der Reihe nach gelöscht werden. Dies wurde nicht implementiert.
<i>Cursor per Tap setzen</i>	Bei dem Tabletop-System kann der Cursor nicht per Tap, sondern nur mittels den Pfeiltasten der angezeigten Tastatur verändert werden.
<i>Text von Subprozess automatisch löschen</i>	Wird der Text von neuen Prozesselementen geändert, löscht sich dieser automatisch, wenn die Tastatur angezeigt wird. Bei Subprozessen wurde dieses Verhalten versehentlich nicht implementiert.

Tabelle 8-3: Implementierungsfehler

8.5 ZUSAMMENFASSUNG

Zusammenfassend lässt sich sagen, dass die Ergebnisse des Experiments darauf hindeuten, dass es aus Anwendersicht keine Vor- bzw. Nachteile hat, zusätzlich zum Tabletop-System ein Tablet zu verwenden. Hingegen stellt dies aus Sicht eines Interaction-Designers eine wichtige Erkenntnis dar. So kann es vor allem bei

größeren Benutzerzahlen hilfreich sein, wenn Benutzer ein privates Ein-/Ausgabegerät verwenden.

Für die Weiterentwicklung des Systems ist vor allem die qualitative Auswertung von großer Bedeutung, da, wie in Kapitel 8.4 aufgezeigt, viele Verbesserungen abgeleitet werden konnten.

9 ZUSAMMENFASSUNG UND AUSBLICK

Das Ziel dieser Arbeit ist es, ein Tabletop-System für kollaborative Geschäftsprozessmodellierung zu erstellen. Dafür werden Grundlagen bzgl. BPMN, HCI und der Wahrnehmungspsychologie erläutert.

Durch eine Literaturliteraturarbeit können bestehende Konzepte aus der HCI adressiert und generelle Anforderungen an ein Tabletop-System identifiziert werden. Anschließend erfolgt die Übertragung der Konzepte und Anforderungen in die Domäne der Geschäftsprozessmodellierung. Dabei identifiziert diese Arbeit Anforderungen, welche das System erfüllen muss.

Das Interaction-Design führt die einzelnen Anforderungen zusammen, um ein stimmiges Feinkonzept zu erstellen. Dabei dient es als Verbindung der einzelnen Ein- und Ausgabekonzepte, zeigt das verwendete Gestenset auf und arbeitet Detaillösungen aus.

Das Feinkonzept wird visuell ausgearbeitet und im Anschluss als Proof-Of-Concept-Implementierung umgesetzt. Das Konzept, bei einem Tabletop-System zusätzlich Tablets zu verwenden, wird durch ein Experiment validiert. Die Ergebnisse zeigen keine Tendenz auf, dass es im durchgeführten Szenario Vor- oder Nachteile mit sich bringt, Tablets zu verwenden. Mögliche Vorteile des privaten Interaktionsgeräts könnten sich bei größeren Gruppen zeigen. Das Experiment zeigt jedoch, dass es keine Nachteile hat und somit das Tablet stimmig in das Gesamtkonzept eingebunden ist. Durch die Erhebung der Usability während des Experiments kann zusammenfassend gesagt werden, dass das Konzept von den Benutzern angenommen wird.

Mithilfe des Prototyps können in Zukunft noch weitere Konzepte durch Experimente validiert werden. Folgende Fragestellungen können dabei adressiert werden:

- Erhöht die Verwendung eines Layouting-Algorithmus die Qualität der Prozessmodelle?

- Ab welcher Gruppengröße ist bei der kollaborativen Geschäftsprozessmodellierung eine signifikant bessere Usability, durch den zusätzlichen Einsatz von Tablets, zu erwarten?
- Hat die Verwendung einer Syntaxprüfung signifikante Auswirkungen auf die Benutzerzufriedenheit oder die Qualität der Prozessmodelle?
- Schränkt die Verwendung der Syntaxprüfung die Kreativität der Benutzer bei der Erstellung von Prozessmodellen ein?
- Soll die Syntaxprüfung, Regelverletzungen nur aufzeigen oder diese unterbinden?

Neben zusätzlichen Experimenten, kann das System in Zukunft weiter implementiert werden um BPMN voll zu unterstützen, und die bereits aufgezeigten Konzepte im System anzubieten.

A. ISO-NORM 9241/10

ISO Norm 9241/10

Fragebogen – ISO-Norm 9241/10

Im Folgenden geht es um die Beurteilung des Systems Process Touch auf Grundlage der internationalen Norm ISO 9241/10. Das Ziel dieser Beurteilung ist es, Schwachstellen des Systems Process Touch aufzudecken und konkrete Verbesserungsvorschläge zu entwickeln.

Um dies zu bewerkstelligen, ist Ihr Urteil als Kenner des Systems von entscheidender Bedeutung! Grundlage Ihrer Bewertung sind Ihre individuellen Erfahrungen mit dem System.

Dabei geht es nicht um eine Beurteilung Ihrer Person, sondern um *Ihre persönliche Bewertung des Systems* mit der Sie arbeiten.

Die Kategorien des Fragebogens weisen immer folgende Form auf:

Beispiel 1:

Das System...	—	-	-	-/+	+	++	+++	Das System...
ist schlecht	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	ist gut

In Beispiel 1 wird danach gefragt, wie gut, bzw. wie schlecht das System ist. Der Benutzer beurteilt in diesem Fall das System zwar als gut, sieht jedoch noch Verbesserungsmöglichkeiten.

Beispiel 2:

Das System...	—	-	-	-/+	+	++	+++	Das System...
ist langsam	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist schnell

In Beispiel 2 beurteilt der Benutzer das System als ziemlich langsam.

Füllen Sie bitte den Beurteilungsbogen *äußerst sorgfältig* aus und lassen Sie keine der Fragen aus! Die Auswertung der Daten erfolgt anonym.

Abbildung A-1: Seite 1 des ISO-Norm 9241/10

ISO-Norm 9241/10

AUFGABENANGEMESSENHEIT

Unterstützt das System die Erledigung Ihrer Arbeitsaufgaben, ohne Sie als Benutzer unnötig zu belasten?

	Das System...	-	-	-	+/-	+	++	+++	Das System...
F 1.1	ist kompliziert zu bedienen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist unkompliziert zu bedienen.
F 1.2	bietet nicht alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bietet alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen.
F 1.3	bietet schlechte Möglichkeiten, sich häufig wiederholende Bearbeitungsvorgänge zu automatisieren.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bietet gute Möglichkeiten, sich häufig wiederholende Bearbeitungsvorgänge zu automatisieren.
F 1.4	erfordert überflüssige Eingaben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erfordert keine überflüssigen Eingaben.
F 1.5	ist schlecht auf die Anforderungen der Arbeit zugeschnitten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist gut auf die Anforderungen der Arbeit zugeschnitten.

Abbildung A-2: Seite 2 des ISO-Norm 9241/10

ISO-Norm 9241/10

SELBSTBESCHREIBUNGSFÄHIGKEIT

Gibt Ihnen das System genügend Erläuterungen und ist sie in ausreichendem Maße verständlich?

	Das System...	—	-	-/+	+	++	+++	Das System...
F 2.1	bietet einen schlechten Überblick über ihr Funktionsangebot.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bietet einen guten Überblick über ihr Funktionsangebot.
F 2.2	verwendet schlecht verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	verwendet gut verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs.
F 2.3	liefert in unzureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	liefert in zureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.
F 2.4	bietet auf Verlangen keine situationsspezifischen Erklärungen, die konkret weiterhelfen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bietet auf Verlangen situationsspezifische Erklärungen, die konkret weiterhelfen.
F 2.5	bietet von sich aus keine situationsspezifischen Erklärungen, die konkret weiterhelfen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bietet von sich aus situationsspezifische Erklärungen, die konkret weiterhelfen.

Abbildung A-3: Seite 3 des ISO-Norm 9241/10

ISO-Norm 9241/10

STEUERBARKEIT

Können Sie als Benutzer die Art und Weise, wie Sie mit dem System arbeiten, beeinflussen?

	Das System...	—	-	-	-/*	+	++	+++	Das System...
F 3.1	bietet keine Möglichkeit, die Arbeit an jedem Punkt zu unterbrechen und von dort später ohne Verluste wieder weiterzumachen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bietet die Möglichkeit, die Arbeit an jedem Punkt zu unterbrechen und von dort später ohne Verluste wieder weiterzumachen.
F 3.2	erzwingt eine unnötig starre Einhaltung von Bearbeitungsschritten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erzwingt keine unnötig starre Einhaltung von Bearbeitungsschritten.
F 3.3	ermöglicht keinen leichten Wechsel zwischen einzelnen Menüs oder Masken.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ermöglicht einen leichten Wechsel zwischen einzelnen Menüs oder Masken.
F 3.4	ist so gestaltet, dass der Benutzer nicht beeinflussen kann, wie und welche Informationen am Bildschirm dargeboten werden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist so gestaltet, dass der Benutzer beeinflussen kann, wie und welche Informationen am Bildschirm dargeboten werden.
F 3.5	erzwingt unnötige Unterbrechungen der Arbeit.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erzwingt keine unnötigen Unterbrechungen der Arbeit.

Abbildung A-4: Seite 4 des ISO-Norm 9241/10

ISO-Norm 9241/10

ERWARTUNGSKONFORMITÄT

Kommt das System durch eine einheitliche und verständliche Gestaltung Ihren Erwartungen und Gewohnheiten entgegen?

	Das System...	—	-	-/+	+	++	+++	Das System...
F 4.1	erschwert die Orientierung, durch eine uneinheitliche Gestaltung.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erleichtert die Orientierung, durch eine einheitliche Gestaltung.
F 4.2	lässt einen im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lässt einen nicht im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht.
F 4.3	informiert in unzureichendem Maße über das, was sie gerade macht.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	informiert in ausreichendem Maße über das, was sie gerade macht.
F 4.4	reagiert mit schwer vorhersehbaren Bearbeitungszeiten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	reagiert mit gut vorhersehbaren Bearbeitungszeiten.
F 4.5	lässt sich nicht durchgehend nach einem einheitlichen Prinzip bedienen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lässt sich durchgehend nach einem einheitlichen Prinzip bedienen.

Abbildung A-5: Seite 5 des ISO-Norm 9241/10

ISO-Norm 9241/10

FEHLERTOLERANZ

Bietet Ihnen das System die Möglichkeit, trotz fehlerhafter Eingaben das beabsichtigte Arbeitsergebnis ohne oder mit geringem Korrekturaufwand zu erreichen?

	Das System...	—	-	-/+	+	++	+++	Das System...
F 5.1	ist so gestaltet, dass kleine Fehler schwerwiegende Folgen haben können.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist so gestaltet, dass kleine Fehler keine schwerwiegenden Folgen haben können.
F 5.2	informiert zu spät über fehlerhafte Eingaben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	informiert sofort über fehlerhafte Eingaben.
F 5.3	liefert schlecht verständliche Fehlermeldungen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	liefert gut verständliche Fehlermeldungen.
F 5.4	erfordert bei Fehlern im Großen und Ganzen einen hohen Korrekturaufwand.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erfordert bei Fehlern im Großen und Ganzen einen geringen Korrekturaufwand.
F 5.5	gibt keine konkreten Hinweise zur Fehlerbehebung.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	gibt konkrete Hinweise zur Fehlerbehebung.

Abbildung A-6: Seite 6 des ISO-Norm 9241/10

ISO-Norm 9241/10

INDIVIDUALISIERBARKEIT

Können Sie als Benutzer des Systems ohne großen Aufwand auf Ihre individuellen Bedürfnisse und Anforderungen anpassen?

	<i>Das System...</i>	—	-	-	-/+	+	++	+++	<i>Das System...</i>
F 6.1	lässt sich von dem Benutzer schwer erweitern, wenn für ihn neue Aufgaben entstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lässt sich von dem Benutzer leicht erweitern, wenn für ihn neue Aufgaben entstehen.
F 6.2	lässt sich von dem Benutzer schlecht an seine persönliche, individuelle Art der Arbeitserledigung anpassen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lässt sich von dem Benutzer gut an seine persönliche, individuelle Art der Arbeitserledigung anpassen.
F 6.3	eignet sich für Anfänger und Experten nicht gleichermaßen, weil der Benutzer sie nur schwer an seinen Kenntnisstand anpassen kann.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	eignet sich für Anfänger und Experten gleichermaßen, weil der Benutzer sie leicht an seinen Kenntnisstand anpassen kann.
F 6.4	lässt sich – im Rahmen ihres Leistungsumfangs – von dem Benutzer schlecht für unterschiedliche Aufgaben passend einrichten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lässt sich – im Rahmen ihres Leistungsumfangs – von dem Benutzer gut für unterschiedliche Aufgaben passend einrichten.
F 6.5	ist so gestaltet, dass der Benutzer die Bildschirmdarstellung schlecht an seine individuellen Bedürfnisse anpassen kann.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist so gestaltet, dass der Benutzer die Bildschirmdarstellung gut an seine individuellen Bedürfnisse anpassen kann.

Abbildung A-7: Seite 7 des ISO-Norm 9241/10

ISO-Norm 9241/10

LERNFÖRDERLICHKEIT

Ist das System so gestaltet, dass Sie sich ohne großen Aufwand in das System einarbeiten konnten und bietet es auch dann Unterstützung, wenn Sie neue Funktionen lernen möchten?

	Das System...	—	=	-/*	+	**	+++	Das System...
F 7.1	erfordert viel Zeit zum Erlernen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erfordert wenig Zeit zum Erlernen.
F 7.2	ermutigt nicht dazu, auch neue Funktionen auszuprobieren.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ermutigt dazu, auch neue Funktionen auszuprobieren.
F 7.3	erfordert, dass man sich viele Details merken muss.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erfordert nicht, dass man sich viele Details merken muss.
F 7.4	ist so gestaltet, dass sich einmal Geleertes schlecht einprägt.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist so gestaltet, dass sich einmal Geleertes gut einprägt.
F 7.5	ist schlecht ohne fremde Hilfe oder Handbuch erlernbar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist gut ohne fremde Hilfe oder Handbuch erlernbar.

Abbildung A-8: Seite 8 des ISO-Norm 9241/10

B. USER EXPERIENCE QUESTIONNAIRE

UEQ – User Experience Questionnaire

Fragebogen – User Experience Questionnaire

Bitte geben Sie Ihre Beurteilung ab.

Um das System zu bewerten, füllen Sie bitte den nachfolgenden Fragebogen aus. Er besteht aus Gegensatzpaaren von Eigenschaften, die das System haben kann. Abstufungen zwischen den Gegensätzen sind durch Kreise dargestellt. Durch Ankreuzen eines dieser Kreise können Sie Ihre Zustimmung zu einem Begriff äußern.

Beispiel:

attraktiv	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattraktiv
-----------	-----------------------	----------------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-------------

Mit dieser Beurteilung sagen Sie aus, dass Sie das System eher attraktiv als unattraktiv einschätzen.

Entscheiden Sie möglichst *spontan*. Es ist wichtig, dass Sie nicht lange über die Begriffe nachdenken, damit Ihre unmittelbare Einschätzung zum Tragen kommt.

Bitte kreuzen Sie immer eine Antwort an, auch wenn Sie bei der Einschätzung zu einem Begriffspaar unsicher sind oder finden, dass es nicht so gut zum System passt.

Es gibt keine „richtige“ oder „falsche“ Antwort. Ihre persönliche Meinung zählt!

Abbildung B-1: Seite 1 des User Experience Questionnaires

UEQ – User Experience Questionnaire

Bitte geben Sie nun Ihre Einschätzung des Systems ab. Kreuzen Sie nur einen Kreis pro Zeile an.

		1	2	3	4	5	6	7	
F 1	unerfreulich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erfreulich
F 2	unverständlich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	verständlich
F 3	kreativ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	phantasielos
F 4	leicht zu lernen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	schwer zu lernen
F 5	wertvoll	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	minderwertig
F 6	langweilig	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	spannend
F 7	uninteressant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	interessant
F 8	unberechenbar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	voraussagbar
F 9	schnell	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	langsam
F 10	originell	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	konventionell
F 11	behindernd	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unterstützend
F 12	gut	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	schlecht
F 13	kompliziert	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	einfach
F 14	abstoßend	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	anziehend
F 15	herkömmlich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	neuartig
F 16	unangenehm	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	angenehm
F 17	sicher	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unsicher
F 18	aktivierend	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	einschläfernd
F 19	erwartungskonform	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nicht erwartungskonform
F 20	ineffizient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	effizient
F 21	übersichtlich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	verwirrend
F 22	unpragmatisch	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pragmatisch
F 23	aufgeräumt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	überladen
F 24	attraktiv	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattraktiv
F 25	sympathisch	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unsympathisch
F 26	konservativ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	innovativ

Abbildung B-2: Seite 2 des User Experience Questionnaires

C. ALLGEMEINE FRAGEN

Allgemeine Fragen

Fragebogen – Allgemeine Fragen

Zum Schluss bitten wir Sie noch folgende allgemeine Fragen zu beantworten. Am Ende dieses Fragebogens haben Sie die Möglichkeit, Feedback zu geben.

Wie bewerten Sie Ihr Prozessmodell?

	<i>Das Prozessmodell...</i>	--	-	-/+	+	++	<i>Das Prozessmodell...</i>
F 1.1	stimmt mit dem Ablauf der realen Welt nicht überein.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	stimmt mit dem Ablauf der realen Welt überein.
F 1.2	ist nicht vollständig. Signifikante Aspekte fehlen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist vollständig. Signifikante Aspekte fehlen nicht.
F 1.3	ist ungenau.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist genau.
F 1.4	besitzt ernsthafte Fehler.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	besitzt keine ernsthaften Fehler.
F 1.5	ist nicht gut gelungen. Ich bin damit nicht zufrieden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ist gut gelungen. Ich bin damit zufrieden.

Wie viel Erfahrung haben Sie mit Prozessmodellierung?

	<i>Ich...</i>	---	--	-	-/+	+	++	+++	<i>Ich...</i>
F 2.1	analysiere nicht regelmäßig Prozessmodelle.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	analysiere regelmäßig Prozessmodelle.
F 2.2	modelliere nicht regelmäßig Prozessmodelle.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	modelliere regelmäßig Prozessmodelle.
F 2.3	kenne mich nicht gut mit BPMN aus.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	kenne mich gut mit BPMN aus.
F 2.4	beherrsche das beurteilte System nicht gut.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	beherrsche das beurteilte System gut.

Abbildung C-1: Seite 1 des Allgemeinen Fragebogens

Allgemeine Fragen

F 2.5	Vor wie vielen Jahren haben Sie angefangen BPMN zu modellieren?	_____
F 2.6	Wie viele Prozessmodelle haben Sie ungefähr in den letzten 12 Monaten modelliert?	_____

Angaben zu Ihrer Person

F 3.1	Ihr Alter?	_____
F 3.2	Ihr Geschlecht?	<input type="radio"/> Männlich <input type="radio"/> Weiblich
F 3.3	Ihr Beruf/Tätigkeit	_____
F 3.4	Ihr höchster Bildungsabschluss	<input type="radio"/> Hauptschule <input type="radio"/> Realschule <input type="radio"/> Abitur <input type="radio"/> Hochschule <input type="radio"/> Sonstige:

Vielen Dank für Ihre aktive Hilfe! Das letzte Wort haben Sie:

Abbildung C-2: Seite 2 des Allgemeinen Fragebogens

D. LEITFADEN FÜR DEN EXPERIMENTLEITER

Leitfaden für den Experimentleiter

Leitfaden für den Experimentleiter

Vorgang	Beschreibung
Begrüßung	<ul style="list-style-type: none"> • Freundliche Begrüßung • Danken, dass Probanden ihre Zeit zur Verfügung stellen und versichern, dass die Mithilfe geschätzt wird • Andere Teammitglieder vorstellen (Name, Aufgaben, Funktion erläutern)
Einführung	<ul style="list-style-type: none"> • Fragen, ob Probanden mit Ton- und Videoaufnahmen einverstanden sind. Versichern, dass diese nur für die Auswertung verwendet werden • Einverständniserklärung austeilen, erläutern • Probanden lesen in Ruhe alles durch • Nachdem Probanden unterschrieben haben, einsammeln • Technik (Kamera, Mikrofon) im Raum zeigen • Das Experiment dauert ca. 30 Minuten
Ablauf	<ul style="list-style-type: none"> • Betonung, dass nicht Probanden, sondern das System getestet wird • Betonung, dass Probanden nichts falsch machen können • Erläuterung <ul style="list-style-type: none"> ○ Übungsaufgabe ○ Szenario modellieren ○ Fragebögen ausfüllen
Videoaufzeichnung starten	
Übungsaufgabe gemeinsam durchführen	<ul style="list-style-type: none"> • Die Übungsaufgabe von gesondertem Blatt durchführen (Szenario 1 bzw. Szenario 2)
Fragen	<ul style="list-style-type: none"> • „Haben Sie noch Fragen, bevor wir mit dem Experiment beginnen?“
Systemaufzeichnung starten	
Experiment starten	<ul style="list-style-type: none"> • Szenario austeilen (Perspektive Unternehmen und Perspektive Bewerber) • Evtl. Tablets austeilen • Probanden darauf hinweisen, dass es jetzt beginnt • Probanden sollen Bescheid geben, wenn sie fertig sind

Abbildung D-1: Seite 1 des Leitfadens für den Experimentleiter

Leitfaden für den Experimentleiter	
Szenario beendet	<ul style="list-style-type: none">• Probanden danken und loben
Systemaufzeichnung beenden	
Fragebogen UEQ	<ul style="list-style-type: none">• Den Fragebogen UEQ austeilen• Stifte austeilen• Erläuterung des Fragebogens gemeinsam durchgehen• Betonen, dass die Fragen ehrlich und spontan beantwortet werden sollen• Nachdem Probanden fertig sind, Fragebogen einsammeln
Fragebogen ISONORM	<ul style="list-style-type: none">• Den Fragebogen ISONORM austeilen• Erläuterung des Fragebogens gemeinsam durchgehen• Betonen, dass die Fragen genau durchgelesen, und reflektiert beantwortet werden soll• Nachdem Probanden fertig sind, Fragebogen einsammeln
Fragebogen Allgemein	<ul style="list-style-type: none">• Den Fragebogen zu den allgemeinen Fragen austeilen• Erläuterung des Fragebogens gemeinsam durchgehen• Betonen, dass die Fragen genau durchgelesen werden sollen, und reflektiert geantwortet werden soll• Nachdem Probanden fertig sind, Fragebogen einsammeln
Verabschieden	<ul style="list-style-type: none">• Nochmals für die Beteiligung an der Studie danken und für die die Arbeit loben• Freundlich verabschieden
Videoaufzeichnung beenden	

Abbildung D-2: Seite 2 des Leitfadens für den Experimentleiter

E. EINVERSTÄNDNISERKLÄRUNG

EINVERSTÄNDNISERKLÄRUNG

Name: _____

Sehr geehrte/-r Teilnehmer/-in,

vielen Dank für Ihre Bereitschaft an dem von der Universität Ulm durchgeführtem Experiment teilzunehmen. Das Experiment dient der Überprüfung des Konzepts und der Benutzerfreundlichkeit des bereitgestellten Systems.

Vor Beginn des Experiments ist es üblich, den Teilnehmer über die Verwendung der Daten zu informieren.

Weiterverwendung der Daten aus dem Experiment

Die Daten, die Sie bei diesem Experiment angeben, werden von uns anonymisiert und nur im Kontext der Masterarbeit von Sebastian Ronis ausgewertet. Ihre Angaben sind für unsere Arbeit von großer Bedeutung und deshalb äußerst wertvoll.

Um eine optimale Auswertung des Experiments zu gewährleisten, wird die Sitzung mittels einer Videokamera dokumentiert. Die entstandenen Aufnahmen werden nur zum Zwecke der Studienauswertung verwendet und nicht veröffentlicht oder weitergegeben. Aktionen die Sie auf dem Tabletop und Tablet ausführen werden teilweise vom System erfasst und gespeichert.

Ich bin damit einverstanden, dass meine Daten wie oben beschrieben aufgezeichnet und verwendet werden. Zudem versichere ich hiermit, mit niemandem über die Inhalte und den Zweck der getesteten Software zu sprechen und an niemanden Informationen darüber weiterzugeben, solange sie nicht der Öffentlichkeit frei zugänglich ist.

(Ort, Datum)

(Unterschrift des Teilnehmers)

Abbildung E-1: Einverständniserklärung

F. ANWENDUNGSSZENARIEN

Szenario – Perspektive Bewerber

ANWENDUNGSSZENARIO: BEWERBUNGS-MANAGEMENT

Modellieren Sie ein Prozessmodell, das nachfolgenden Bewerbungsablauf in BPMN abbildet. Verwenden Sie dazu das bereitgestellte System.

Ein Bewerber stellt seine Bewerbungsunterlagen zusammen. Hierfür schreibt er einen Lebenslauf sowie ein entsprechendes Anschreiben. Anschließend schickt er die Bewerbungsunterlagen an das Unternehmen.

Als Antwort bekommt er von dem Unternehmen entweder eine Absage oder eine Einladung zum Vorstellungsgespräch. Im Falle einer Einladung, wird das Vorstellungsgespräch geführt.

Im Anschluss an das Vorstellungsgespräch erhält der Bewerber wiederum eine Antwort welche eine Absage oder einen Gehaltsvorschlag enthält.

Im Falle des Gehaltsvorschlags, prüft er diesen und nimmt ihn an, wenn er seinen Vorstellungen entspricht. Sonst teilt er dem Unternehmen mit, dass der Gehaltsvorschlag zu niedrig ist. Folglich bekommt er einen neuen Vorschlag von dem Unternehmen zugesandt. Dies wiederholt sich so lange, bis der Bewerber mit einem Gehaltsvorschlag einverstanden ist, oder er eine Absage des Unternehmens bekommt.

Im Falle einer Einigung bekommt er den Arbeitsvertrag zugeschickt, welchen er ausfüllt und zurückschickt.

Abbildung F-1: Anwendungsszenario – Bewerber

Szenario – Perspektive Unternehmen

ANWENDUNGSSZENARIO: BEWERBUNGS-MANAGEMENT

Modellieren Sie ein Prozessmodell, das nachfolgenden Bewerbungsablauf in BPMN abbildet. Verwenden Sie dazu das bereitgestellte System.

Das Unternehmen bekommt eine Bewerbung samt Bewerbungsunterlagen zugeschickt. Nach einer Überprüfung und Bewertung der Bewerbungsunterlagen wird entweder eine Absage oder eine Einladung zum Vorstellungsgespräch als Antwort verschickt.

Nach dem Vorstellungsgespräch beraten sich alle Beteiligten in dem Unternehmen. Anschließend wird das entsprechende Resultat an den Bewerber verschickt. Diese Antwort kann eine Absage oder einen Gehaltsvorschlag enthalten.

Nimmt der Bewerber diese nicht an, wird über einen höheren Gehaltsvorschlag beraten und an den Bewerber geschickt. Nachdem das Unternehmen eine Zusage bekommen hat, verschickt es einen entsprechenden Arbeitsvertrag.

Ist der Vertrag ausgefüllt eingegangen, wird der Bewerber eingestellt. Hierfür muss die Personalabteilung informiert werden, die Daten des Arbeitsvertrags in die Mitarbeiterdatenbank eingetragen werden und ein Arbeitsplatz für den neuen Mitarbeiter eingerichtet werden.

Abbildung F-2: Anwendungsszenario - Unternehmen

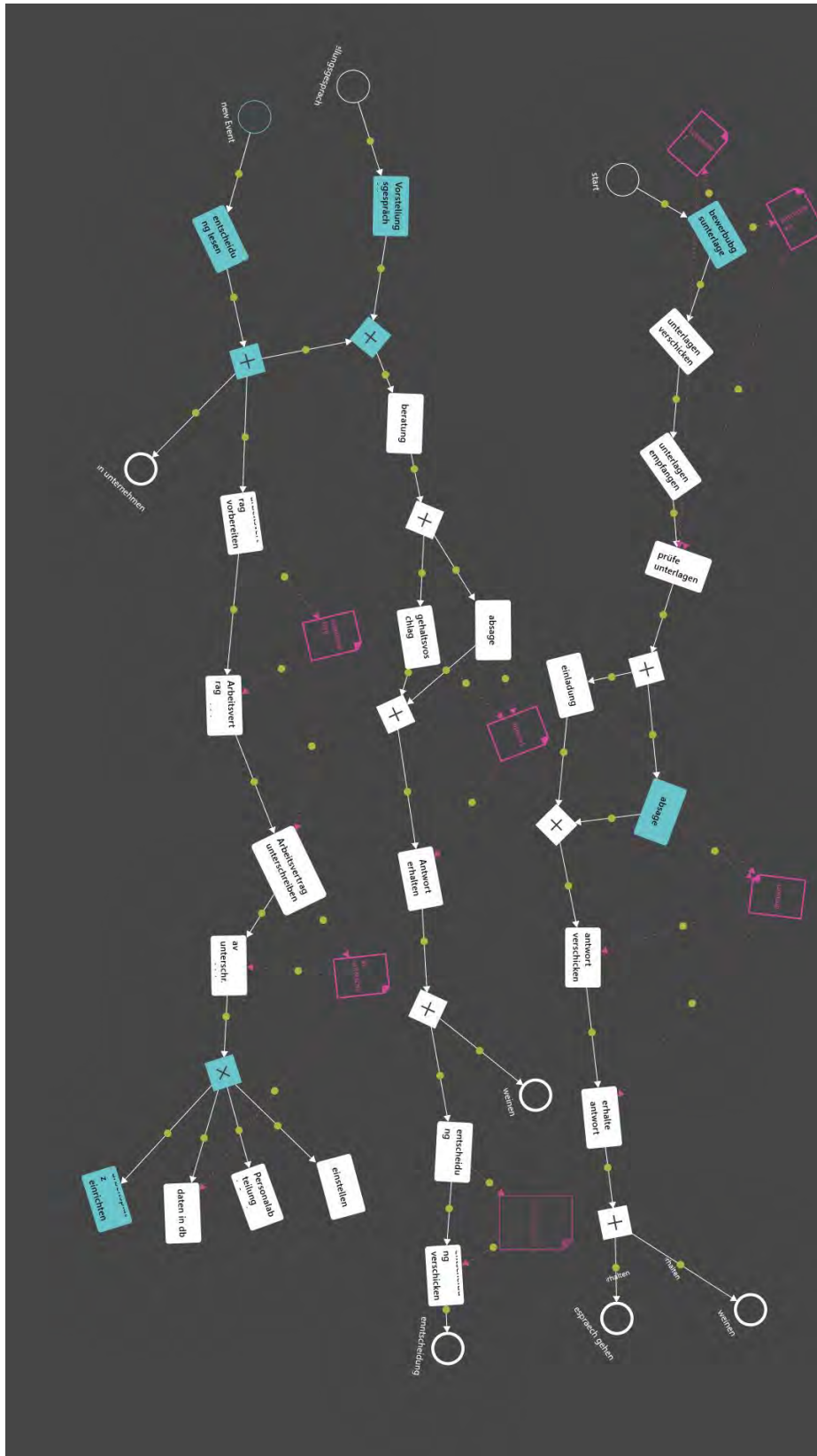


Abbildung G-2: Prozessmodell der Probanden 3 und 4

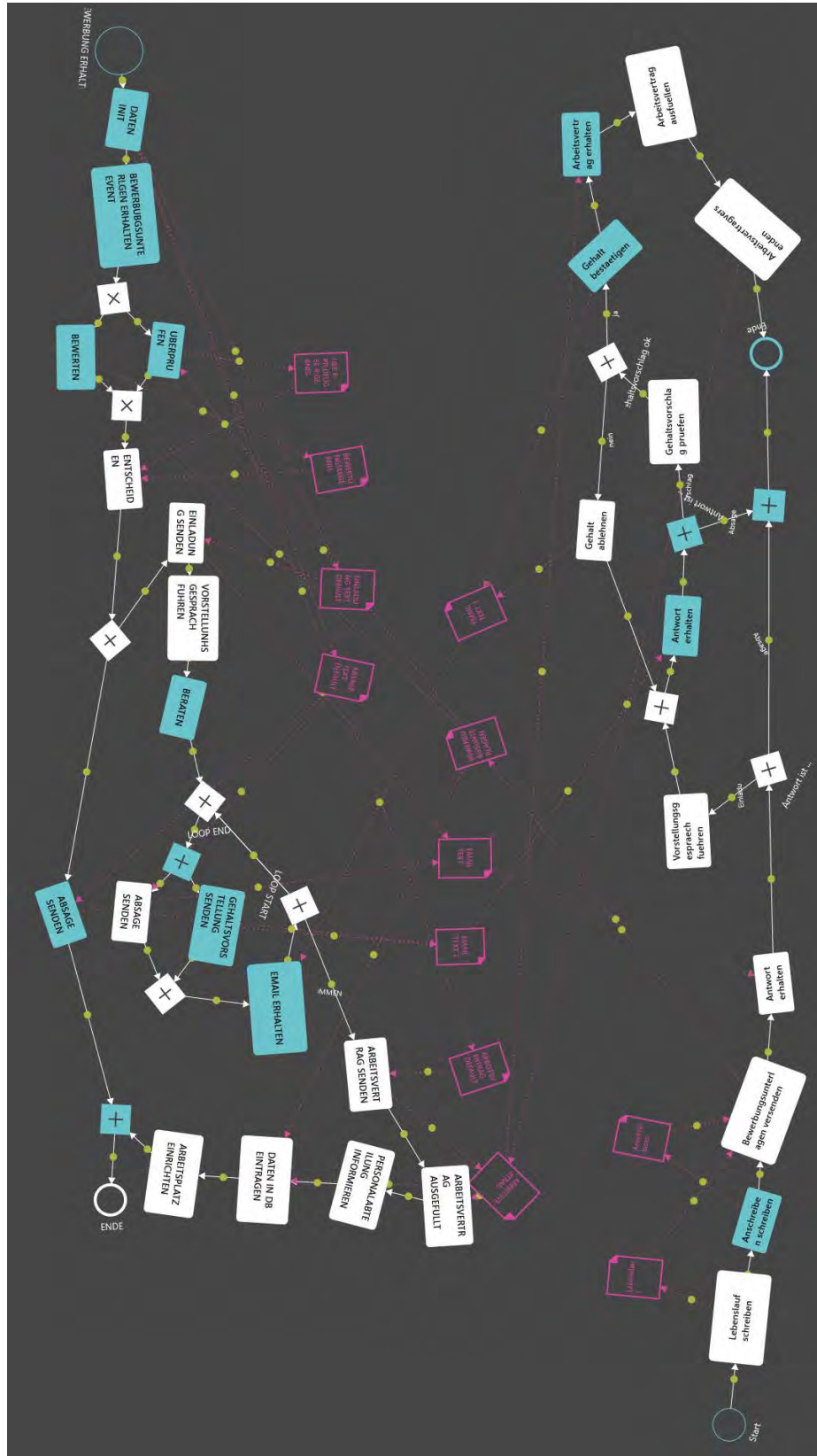


Abbildung G-3: Prozessmodell der Probanden 5 und 6

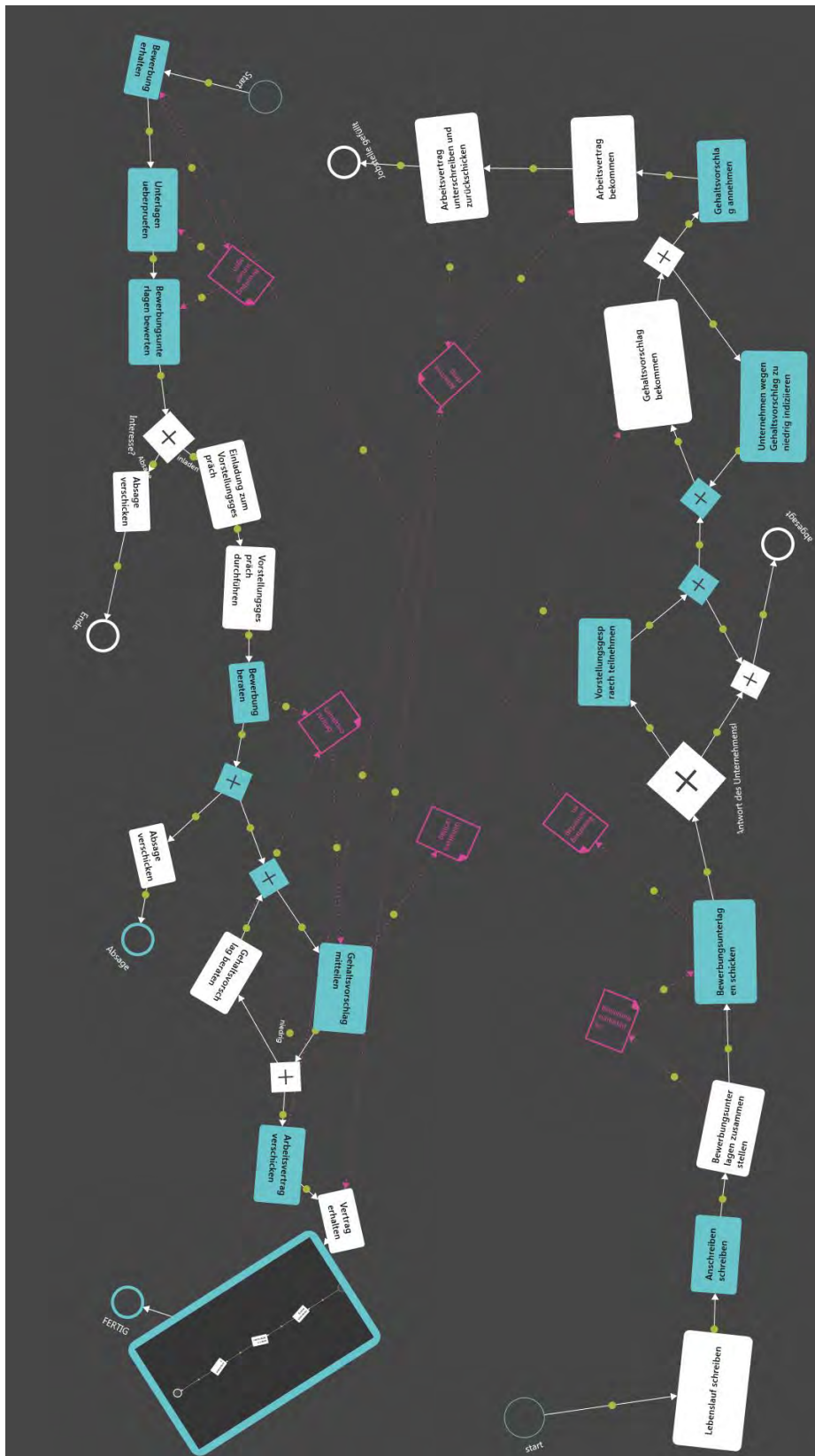


Abbildung G-6: Prozessmodell der Probanden 11 und 12

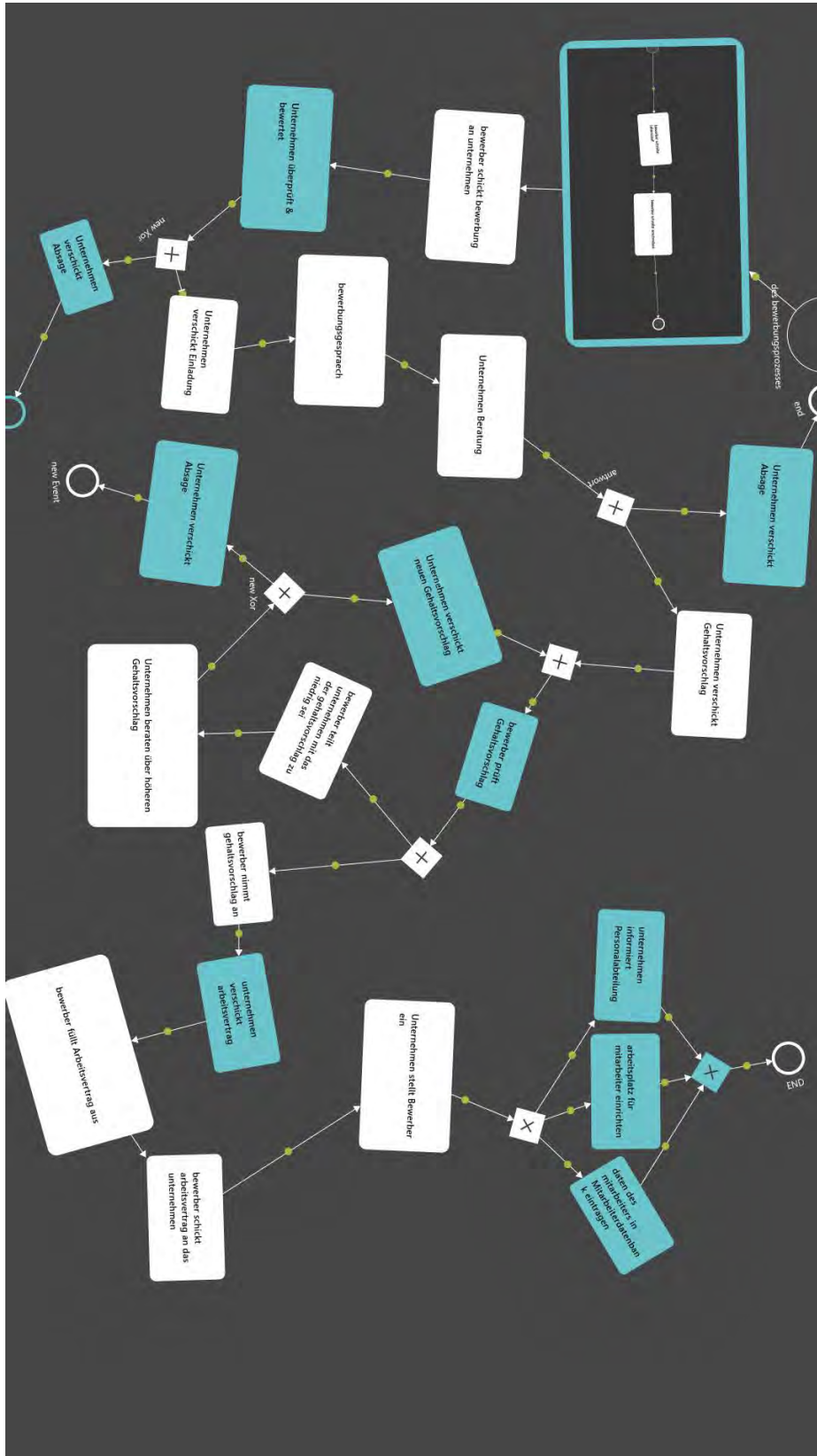


Abbildung G-7: Prozessmodell der Probanden 13 und 14

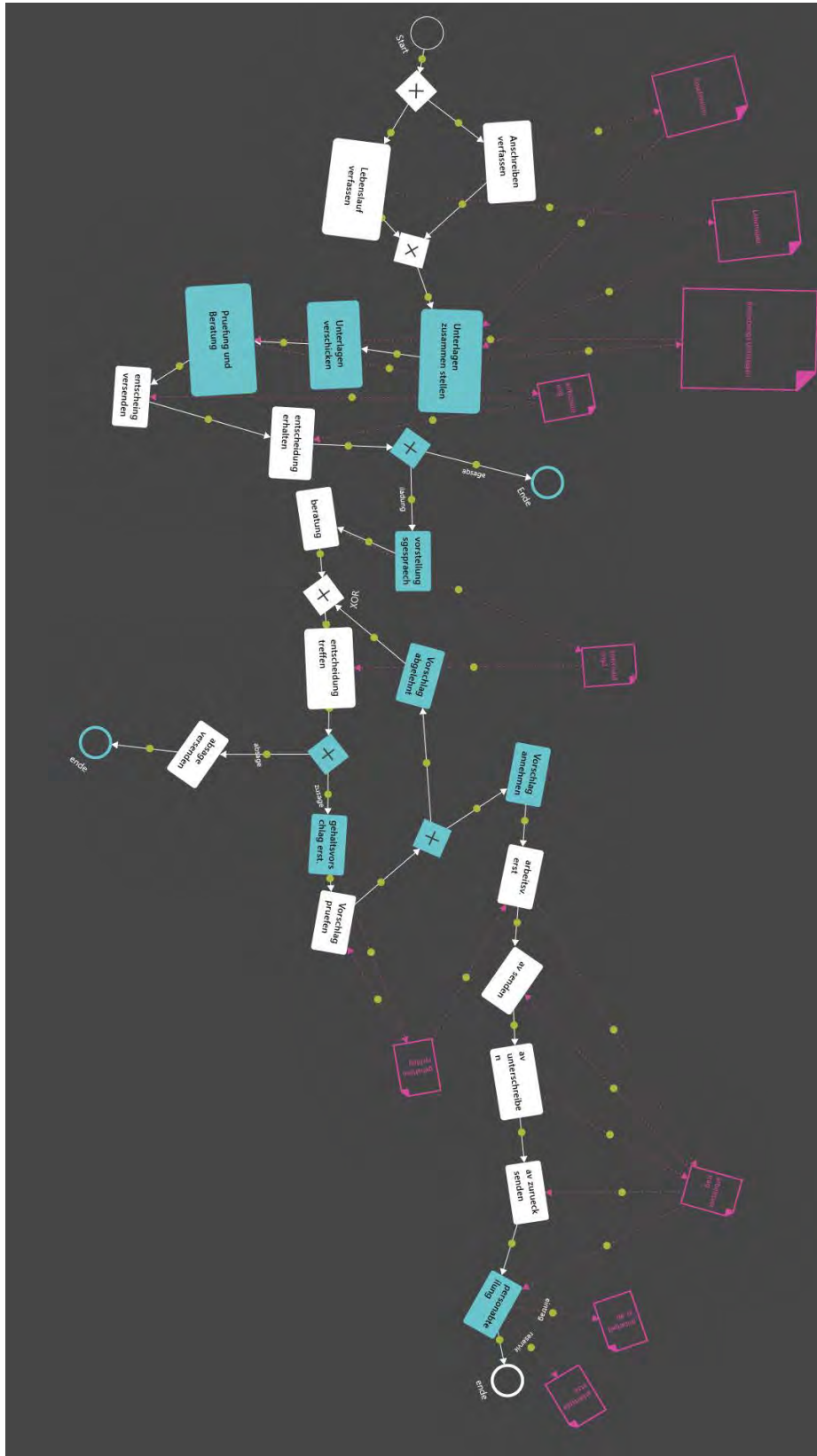


Abbildung G-8: Prozessmodell der Probanden 15 und 16

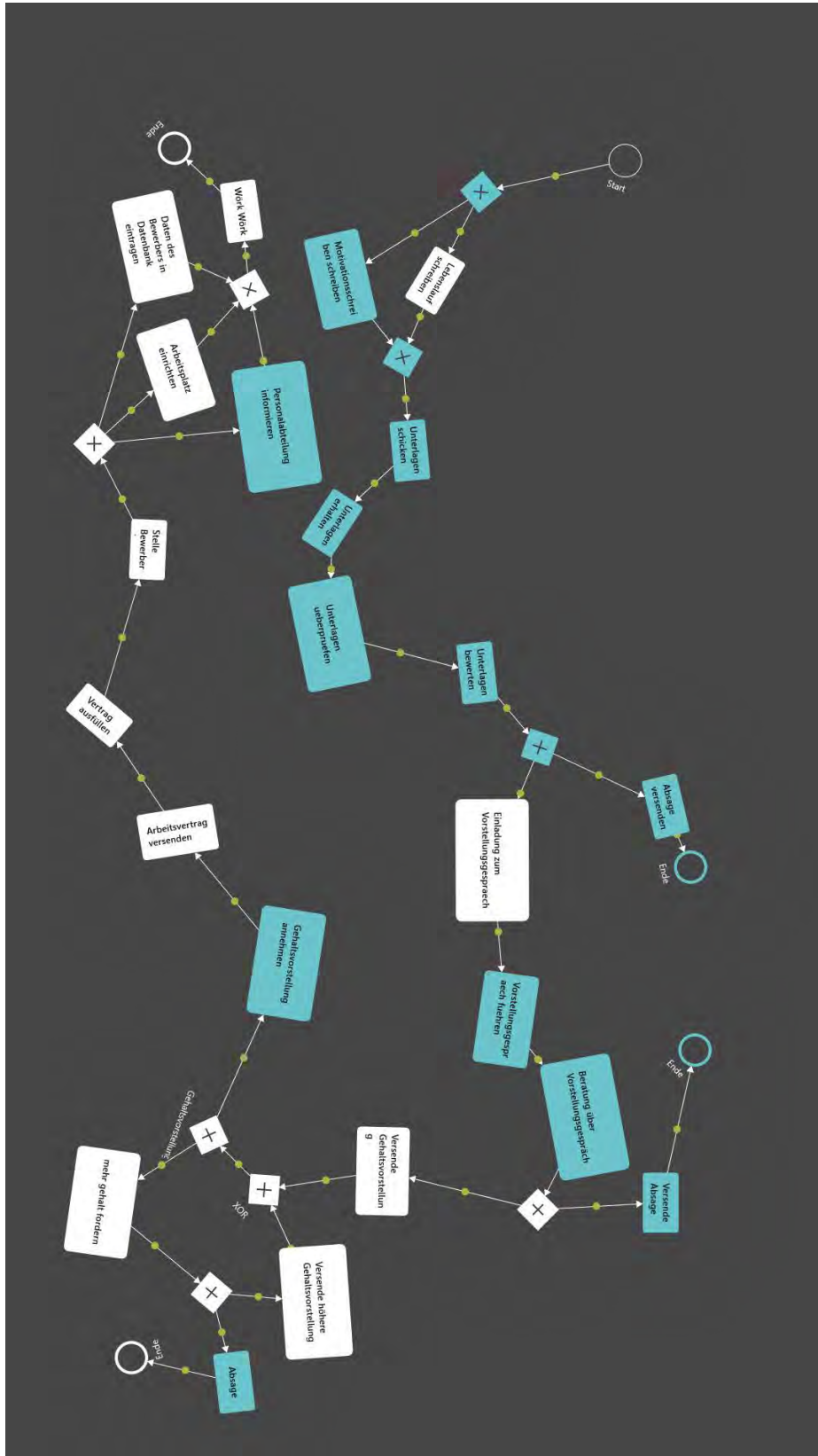


Abbildung G-9: Prozessmodell der Probanden 17 und 18

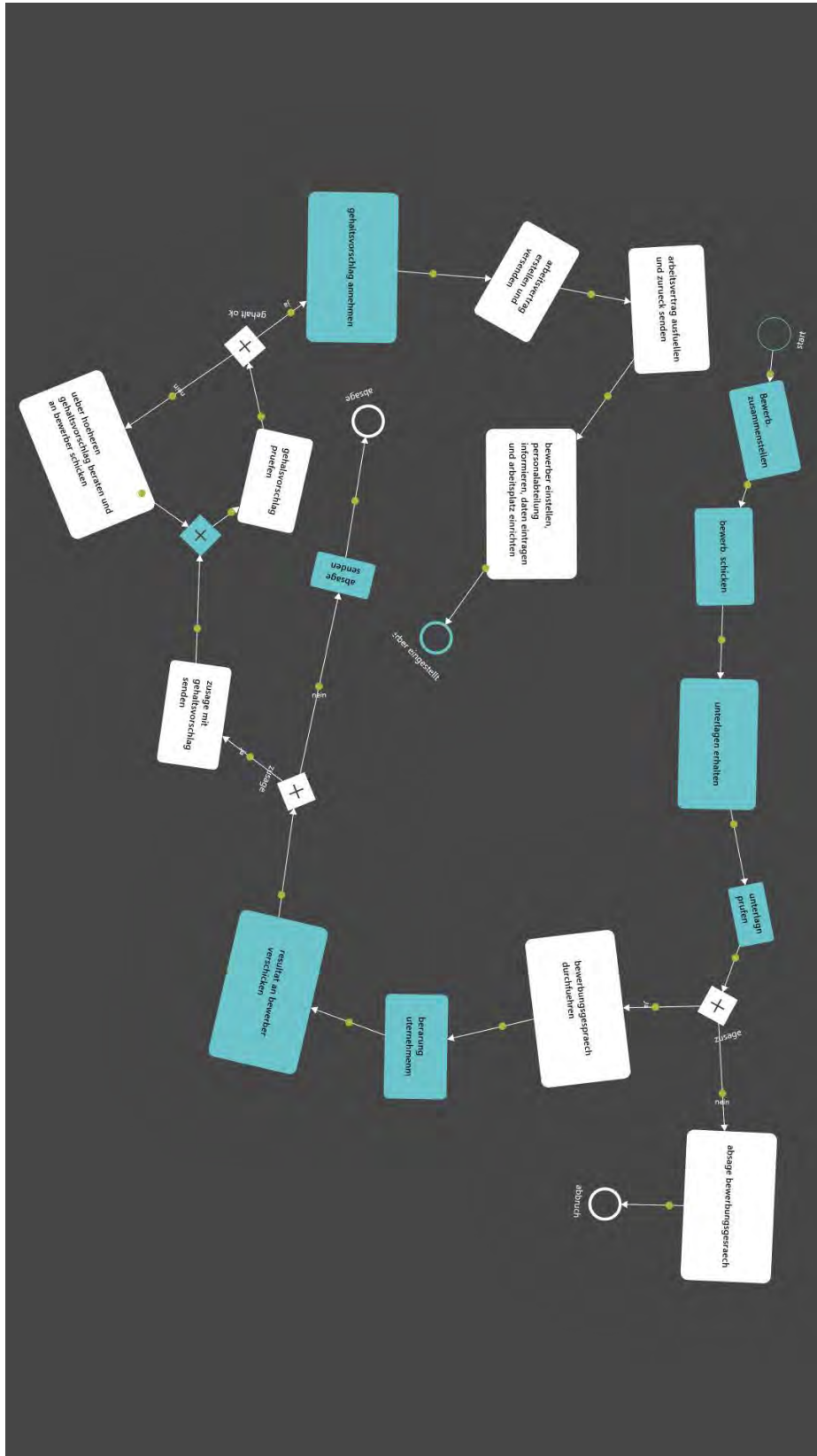


Abbildung G-10: Prozessmodell der Probanden 19 und 20

H. DATEN DES EXPERIMENTS

Faktor Proband	ohne		mit		ohne		mit		ohne		mit		ohne		mit		ohne			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F 1.1	3	2	6	5	6	5	5	4	6	6	6	4	6	6	5	5	3	6	5	6
F 1.2	5	3	3	3	3	2	2	2	4	4	4	2	6	6	6	6	5	4	2	3
F 1.3	2	2	3	5	1	6	6	5	4	4	6	5	2	5	1	4	7	2	4	7
F 1.4	2	2	5	7	7	7	4	5	5	6	5	6	6	6	4	3	6	3	6	6
F 1.5	5	4	5	5	5	3	4	3	6	6	7	3	6	6	6	6	5	6	6	6
F 2.1	3	2	6	6	3	5	5	6	3	6	6	2	7	7	6	1	3	4	5	5
F 2.2	4	3	6	6	5	7	7	6	7	7	5	6	7	7	6	6	6	6	6	6
F 2.3	3	3	7	2	4	5	2	2	4	5	6	5	7	7	6	3	4	6	6	6
F 2.4	2	2	4	4	5	4	2	2	2	4	5	1	6	5	4	1	3	2	4	1
F 2.5	2	2	4	5	5	3	2	2	2	5	5	1	5	6	4	5	3	2	3	1
F 3.1	7	7	1	3	1	1	4	6	4	1	1	1	6	3	4	1	4	7	1	1
F 3.2	5	7	7	6	2	5	2	6	7	5	7	7	7	4	3	7	7	7	7	6
F 3.3	5	4	7	6	1	4	5	6	4	5	1	3	7	7	4	7	6	5	1	7
F 3.4	2	2	6	6	5	2	4	4	5	6	7	6	6	6	2	7	4	6	1	7
F 3.5	7	4	6	5	6	7	4	3	6	6	1	5	7	7	2	2	6	5	7	7
F 4.1	6	2	6	6	5	7	6	6	6	6	6	5	7	7	3	6	5	4	3	6
F 4.2	5	7	7	6	7	7	5	6	6	7	4	2	7	7	5	7	7	6	6	6
F 4.3	5	6	5	5	6	7	4	6	4	6	5	3	7	7	4	6	6	7	6	2
F 4.4	7	6	7	6	5	7	5	1	4	7	6	4	6	6	7	7	6	6	5	4
F 4.5	6	2	7	6	7	6	6	6	6	6	7	6	7	7	7	6	6	6	7	7
F 5.1	3	2	6	6	6	4	2	3	6	5	7	2	6	3	1	2	7	6	6	4
F 5.2	4	6	7	6	5	7	2	4	5	6	1	2	3	6	1	6	6	7	7	7
F 5.3	1	1	1	4	4	7	1	4	4	6	2	6	4	6	5	1	6	7	5	1
F 5.4	4	2	6	5	3	2	1	2	5	5	1	3	5	5	1	4	6	6	5	1
F 5.5	1	3	2	2	1	2	2	2	2	3	1	2	4	6	2	1	3	4	1	1
F 6.1	1	1	1	1	1	1	2	4	4	1	2	4	6	1	4	1	4	4	5	7
F 6.2	1	1	1	2	1	2	2	5	4	1	3	5	6	4	5	1	3	5	3	5
F 6.3	4	1	6	4	2	7	3	6	6	5	6	6	6	7	3	6	7	6	2	7
F 6.4	1	1	1	1	1	2	3	5	4	2	4	5	6	2	6	6	4	5	5	6
F 6.5	5	1	1	3	1	2	2	2	4	2	2	2	7	6	6	1	6	6	4	7
F 7.1	6	7	7	5	7	7	4	6	7	6	7	5	7	7	7	6	7	6	7	7
F 7.2	6	7	7	6	4	2	6	6	6	5	7	2	7	7	6	5	6	6	7	7
F 7.3	4	7	7	4	7	6	5	6	6	6	6	5	7	6	2	6	7	6	7	5
F 7.4	5	7	7	6	7	7	5	6	5	6	7	6	7	7	3	6	7	7	7	6
F 7.5	2	3	7	6	2	5	6	7	3	6	7	2	6	7	2	5	3	6	5	3

1△--- 2△-- 3△- 4△+/- 5△+ 6△++ 7△+++

Tabelle H-1: Daten des ISO-Norm 9241/10

H DATEN DES EXPERIMENTS

Faktor Proband	ohne		mit		ohne		mit		ohne		mit		ohne		mit		ohne			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F 1	5	3	5	5	6	6	5	5	6	6	6	6	4	5	5	3	6	6	4	6
F 2	5	4	6	6	6	6	6	7	6	6	6	5	5	5	5	4	6	7	6	6
F 3	2	2	1	2	1	4	3	4	2	1	1	1	1	1	2	2	2	2	4	
F 4	2	3	1	2	2	2	2	2	1	2	2	3	3	4	2	4	1	2	3	1
F 5	3	4	2	3	2	2	2	4	4	2	2	3	3	3	3	4	2	3	3	3
F 6	5	5	6	6	6	5	6	6	5	6	6	6	6	6	5	6	4	5	5	
F 7	3	6	6	6	7	6	6	6	6	6	7	6	6	6	6	5	7	5	5	5
F 8	4	2	4	4	4	5	3	3	6	4	5	4	3	2	6	3	6	4	2	4
F 9	5	2	6	5	3	2	4	6	2	3	5	3	3	6	4	6	4	5	5	4
F 10	2	2	1	2	2	4	3	5	1	1	1	1	1	1	2	3	3	3	1	4
F 11	4	3	6	5	6	6	5	3	5	4	6	4	5	4	5	2	5	4	3	5
F 12	3	4	2	3	3	2	3	4	2	3	2	2	2	4	1	5	2	2	4	2
F 13	5	4	5	4	6	6	4	7	7	5	5	4	5	5	5	4	5	5	3	7
F 14	5	4	5	5	6	6	5	6	6	5	6	6	5	4	5	4	4	5	5	6
F 15	5	6	7	7	5	5	6	5	6	7	7	7	7	7	6	5	6	6	6	4
F 16	5	4	5	5	6	5	6	4	6	6	6	5	5	5	5	4	6	5	3	6
F 17	4	5	2	4	4	3	4	7	3	2	3	3	4	6	4	6	2	4	3	2
F 18	3	3	1	2	2	2	3	2	3	2	1	2	3	3	2	3	2	3	2	3
F 19	3	5	2	3	4	3	3	3	2	2	2	3	3	6	2	5	3	3	5	3
F 20	2	3	6	3	5	5	5	3	5	6	4	4	2	3	5	2	6	4	2	5
F 21	4	6	2	3	2	3	3	7	5	5	5	4	3	4	2	3	3	3	6	4
F 22	4	2	5	5	6	6	4	5	3	6	5	5	4	3	6	3	4	2	2	5
F 23	4	5	2	3	2	3	4	2	6	4	2	4	2	3	3	2	6	4	1	5
F 24	3	4	2	3	1	3	1	4	2	3	2	2	2	3	1	3	2	3	5	2
F 25	3	4	2	2	2	2	1	3	2	2	2	2	2	4	1	5	3	3	5	2
F 26	6	6	6	7	6	5	6	4	6	7	7	6	7	6	6	5	6	5	6	4

1△--- 2△-- 3△- 4△+/- 5△+ 6△++ 7△+++

Tabelle H-2: Daten des UEQ

Faktor Proband	ohne		mit		ohne		mit		ohne		mit		ohne		mit		ohne			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F 1.1	4	4	4	4	5	4	3	1	4	4	4	2	5	5	4	4	4	6	3	4
F 1.2	3	4	1	5	5	4	4	1	2	4	4	2	5	5	4	4	5	6	2	1
F 1.3	2	4	1	4	5	4	3	2	2	4	4	1	5	4	2	4	4	6	2	4
F 1.4	3	3	4	4	5	5	3	1	4	4	4	2	5	5	5	4	5	5	5	5
F 1.5	3	3	4	4	5	5	2	1	3	3	4	3	5	5	4	4	4	5	4	4
F 2.1	1	1	6	4	5	6	6	7	6	7	7	7	5	6	1	2	5	5	6	7
F 2.2	1	2	6	3	5	5	5	6	5	7	7	7	3	6	1	2	6	5	6	7
F 2.3	2	2	7	5	5	7	6	7	6	7	7	7	5	5	3	5	7	6	6	6
F 2.4	2	2	6	5	5	5	6	4	4	3	6	6	6	6	4	5	5	6	5	3
F 2.5	0	0	6	3	4	3	4	5	5	6	8	6	4	4	2	2	3	3	1,5	1,5
F 2.6	0	0	12	4	5	15	25	30	10	100	5	50	2	10	1	30	18	20	75	100
F 3.1	34	37	27	34	26	25	30	29	28	34	30	31	25	28	25	26	24	25	25	27
F 3.2	m	m	m	m	m	m	m	m	m	m	w	m	w	m	m	m	m	m	m	w
F 3.3	mi	mi	mi	mi	s	s	i	mi	mi	mi	i	s	s	s	s	s	s	s	s	s
F 3.4	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h	H

1△--- 2△-- 3△- 4△+/- 5△+ 6△++ 7△+++

Legende:

m=männlich

w=weiblich

h=Hochschule

s=Student

i=Informatiker

mi=wissenschaftlicher Mitarbeiter

Tabelle H-3: Daten des Fragebogens Allgemeine Fragen

H DATEN DES EXPERIMENTS

Faktor Proband	ohne		mit		ohne		mit		ohne		mit		ohne		mit		ohne		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Analyse der Prozessmodelle																			
Aktivitäten	22		24		29		23		26		25		23		20		23		16
Kanten	34		41		52		22		36		22		36		31		39		24
Gateways	11		6		13		12		7		9		7		6		9		4
Datenelemente	12		7		12		8		6		6		0		9		0		0
Events	1		8		4		2		2		7		5		4		5		4
Datenkanten	24		19		32		14		15		16		0		21		0		0
Anzahl der Modelle	2		2		2		1		3		2		1		1		1		1
Daten des Loggers																			
Elemente mit Geste erzeugt	64		43		27		46		27		56		49		63		38		22
Elemente mit Menü erzeugt	28		16		66		18		34		10		16		29		18		23
Elemente mit Geste gelöscht	56		18		31		24		31		20		42		66		16		24
Elemente mit Menü gelöscht	4		0		14		1		5		0		10		2		1		0
Anzahl Tablet-Touch	0		24		0		1		0		11		13		0		4		0
Buchstaben Tabletop	928		484		1182		791		941		769		49		880		343		897
Buchstaben Tablet	0		512		0		139		0		317		932		0		539		0
Tabletop erzeugt	92		59		93		47		61		28		24		92		25		45
Tablet erzeugt	0		0		0		15		0		37		41		0		31		0
Verbindung Tabletop	102		80		119		50		79		34		24		96		29		37
Verbindung Tablet	0		0		0		12		0		21		29		0		13		0
# vor Tablet-Touch	0		0		0		10		0		3		10,44		0		5,25		0
Dauer	46,25		35,18		52,35		41,72		35,7		39,1		56,62		51,15		26,8		20,3
Expertenbewertung																			
Experte 1																			
Korrektheit	7		5		6		6		5		7		7		7		7		5
Vollständigkeit	7		6		7		5		4		6		4		6		5		3
Relevanz	7		6		7		6		4		6		7		7		6		6
Authentizität	5		3		7		5		4		6		4		7		5		4
Experte 2																			
Korrektheit	6		6		7		5		5		7		7		6		6		6
Vollständigkeit	7		6		7		5		5		6		6		5		6		5
Relevanz	7		6		5		7		6		7		7		7		7		6
Authentizität	6		6		6		6		5		7		6		7		6		6

1△--- 2△-- 3△- 4△+/- 5△+ 6△++ 7△+++

Tabelle H-4: Weitere Daten des Experiments

Proband	Bemerkung/Feedback
20	Rollen integrieren Drehen ist nicht so komfortabel
19	Rollen (Pools und Lanes) wären sehr hilfreich Automatisches Anordnen/Ausrichten zur besseren Übersicht
18	System auf dem Tablet reagiert bei vielen Elementen sehr träge Layoutalgorithmus? Wenn ja, sehr uninnovativ
17	Olal, slap dat table
16	Texteingabe: Alles löschen bei langen drücken auf "<-" evtl. Cursor per Tap setzen Das aufeinanderschieben der Elemente ist etwas umständlich. Schöner wäre es vielleicht mit den Finger zwei Knoten zu verbinden
15	Wichtig: Löschvorgänge müssen erschwert werden & rückgängig gemacht werden können! Das Verschieben von Elementen liefert unvorhersehbare Resultate, da sich andere Elemente teilweise mitbewegen Beim Drehen und Anordnen wäre ein Raster sehr hilfreich und würde deutlich die Übersichtlichkeit erhöhen
14	Text von Subprozess löscht sich auf Pixelsense nicht automatisch Wenn man auf dem Surface mehrmals Undo nach einer Übertragung macht kommen die Elemente wieder, nicht jedoch die Verbindungen zwischen ihnen.
13	Beim übertragen vom Tablet an den Tisch wurde alles laut Tablet erfolgreich übertragen, aber auf dem Tisch war nichts zu sehen. Als ein neuer Task auf dem Tisch erstellt wurde & dessen Beschreibung geändert wurde, war das Übertragene wieder sichtbar und der neu erstellte Task verschwunden Einmal ist es vorgekommen, dass ein Task unabsichtlich gelöscht wurde, als dessen Beschreibung korrigiert wurde. Dabei wurde zu lang auf den Backspace Button gedrückt. Ein Undo-Button auf dem Tisch wäre nützlich gewesen.
12	Die Bedienbarkeit des Tisches lässt zu wünschen übrig...
11	Usability ist noch ein Problem mit dem Tisch. Manche Sachen sind nicht so einfach. Vielleicht statt mit dem Tablet auf den Tisch gehauen, soll man "drag-and-drop" Funktionen nutzen?
10	alignment ist manchmal problematisch Fehlen von bestimmten BPMN Konstrukten macht modellierung anfangs schwer (z.B. Pools & Lanes) insgesamt interessant und motivierend effizient durch gleichzeitiges modellieren
9	Sehr gute Idee, ein paar Verbesserungsvorschläge "User-Token o.Ä.", damit das System weiß, wo der Modellierer am Tisch steht und so neue Elemente korrekt ausrichtet (unnötiges Drehen der Elemente verhindern) Erweiterung um Lanes, Intermediate Events
8	
7	
6	Ich denke die automatische Anpassung des Layouts ist einen Tick zu stark
5	
4	Sehr interessantes und innovatives System. Der spielerische Aspekt macht sehr viel aus. Gefällt mir!!!
3	sehr cooles System - echt innovativ! Weiter so! Pools/Lanes wären wichtige Erweiterungen für das System um besser zu Aktivieren

- 2 nettes System, aber Bedienung an vielen Stellen noch Hakelig. Das AutoLayout und das Drehen ermöglichen es nicht, ein sauberes Layout zu erzeugen bzw. nur mit sehr viel Aufwand
Positiv ist, dass zwei oder mehr Leute gleichzeitig am Modell arbeiten können.
Durch Drehen und AutoLayout ist das Modell sehr unübersichtlich, was meiner Meinung nach die Arbeit erschwert
Die exakte Semantik von BPMN mit diesen 4 Symbolen sollte man lernen, dann ist das Modell am Ende vielleicht korrekter.
- 1 Gelöschte Transitionen können nicht immer wiederhergestellt werden. Insb. Wenn die Richtung verändert wurde
neue Elemente tauchen irgendwo auf ohne, dass man dies sofort meinte
Gestenerkennung für Erzeugung der Basiselemente häufig fehlerhaft
Subprozesse können nicht erstellt werden
Hat Spaß gemacht ;-)

Tabelle H-5: Feedback der Probanden

ABKÜRZUNGSVERZEICHNIS

<i>BPM</i>	Business Process Management
<i>BPMN</i>	Business Process Model and Notation
<i>CBPM</i>	Collaborative BPM
<i>GUI</i>	Graphical User Interface
<i>HCI</i>	Human Computer Interaction
<i>ISO</i>	Internationale Organisation für Normung
<i>MVC</i>	Model View Controller (Software Design Pattern)
<i>MVP</i>	Model View Presenter (Software Design Pattern)
<i>MVVM</i>	Model View ViewModel (Software Design Pattern)
<i>NFC</i>	Near Field Communication
<i>NUI</i>	Natural User Interface
<i>RFID</i>	Radiofrequency Identification
<i>UEQ</i>	User Experience Questionnaire
<i>WCF</i>	Windows Communication Foundation
<i>WinRT</i>	Windows Runtime (Windows 8 Betriebssystem für ARM Prozessoren)
<i>WPF</i>	Windows Presentation Foundation
<i>HTTP</i>	Hypertext Transfer Protocol
<i>TCP</i>	Transmission Control Protocol
<i>NDH</i>	Nicht dominante Hand
<i>DH</i>	Dominante Hand
<i>XAML</i>	Extensible Application Markup Language

ABBILDUNGSVERZEICHNIS

Abbildung 3-1: BPMN-Kategorien	7
Abbildung 3-2: Ereignisse in BPMN.....	8
Abbildung 3-3: Sequenzfluss in BPMN.....	8
Abbildung 3-4: Nachrichtenfluss in BPMN	8
Abbildung 3-5: Assoziation in BPMN.....	9
Abbildung 3-6: Aktivitäten in BPMN	9
Abbildung 3-7: Portale in BPMN.....	9
Abbildung 3-8: Datenobjekt in BPMN	10
Abbildung 3-9: Pool in BPMN.....	10
Abbildung 3-10: Lane in BPMN.....	10
Abbildung 3-11: Anmerkung in BPMN.....	10
Abbildung 3-12: Prozessmodell	11
Abbildung 3-13: Prinzip von Touch&Interact [12]	14
Abbildung 3-14: Aufbau von PhoneTouch [18]	15
Abbildung 3-15: Automatische Extraktion von Modellen aus einem Bild	16
Abbildung 3-16: Skizzenbasierte Eingabe.....	17
Abbildung 3-17: Gesetz der Nähe.....	17
Abbildung 3-18: Gesetz der Ähnlichkeit	18
Abbildung 3-19: Gesetz der Schließung	18
Abbildung 4-1: Tabletop als zentrale Instanz	31
Abbildung 4-2: Alternativer Systemaufbau.....	32
Abbildung 4-3: Skizzenbasierte Eingabe	34
Abbildung 4-4: Auswahl der Prozesselemente.....	34
Abbildung 4-5: Gedrucktes und mit NFC-Tags präpariertes Prozessmodell	35
Abbildung 4-6: Mobile Augmented Reality.....	36
Abbildung 4-7: Projizierte Information.....	36
Abbildung 5-1: Hauptmenü öffnen	40
Abbildung 5-2: Arbeiten mit Verbindungspfeilen.....	40
Abbildung 5-3: Virtuelle Tastatur	41
Abbildung 5-4: Aktivität erstellen.....	43

Abbildung 5-5: Gateway erstellen	43
Abbildung 5-6: Ereignis erstellen.....	44
Abbildung 5-7: Prozesselemente verschieben	44
Abbildung 5-8: Elemente verbinden	44
Abbildung 5-9: Selektierung von Prozesselementen	44
Abbildung 5-10: Tap auf Hintergrund	45
Abbildung 5-11: Zoomen	45
Abbildung 5-12: Löschgeste.....	45
Abbildung 5-13: Subprozess erstellen.....	45
Abbildung 5-14: Subprozess öffnen.....	46
Abbildung 5-15: Lokale Vergrößerung.....	46
Abbildung 5-16: Menübasierte Auswahl auf dem Smartphone	47
Abbildung 5-17: Smartphone als Tastatur	48
Abbildung 5-18: Sprachanmerkung	48
Abbildung 5-19: Anziehung (Blau) und Abstoßung (rot) der Knoten.....	50
Abbildung 5-20: Ablauf des Layouting-Vefahrens.....	51
Abbildung 6-1: Farbtöne für Rückmeldungen auf der Hintergrundfarbe mit RGB- Werten	54
Abbildung 6-2: Informationsfarben auf der Hintergrundfarbe mit RGB-Werten...	55
Abbildung 6-3: Icons des Hauptmenüs.....	56
Abbildung 6-4: Icons des Kontextmenüs.....	56
Abbildung 6-5: Anpassung von Buchstaben bei der Schriftart Segoe360 [30].....	57
Abbildung 6-6: Prozesselemente in Process Touch.....	57
Abbildung 6-7: Hauptmenü.....	58
Abbildung 6-8: Verbindungsmenü	58
Abbildung 6-9: Animation des Kontextmenüs	59
Abbildung 6-10: Kontextmenü eines Prozesselements.....	59
Abbildung 6-11: virtuelle Tastatur auf dem Tabletop-System	60
Abbildung 6-12: Hauptmenü auf dem Tablet.....	61
Abbildung 6-13: Kontextmenü des Tablets	62
Abbildung 6-14: Tastatur auf dem Tablet mit Autovervollständigung	62
Abbildung 6-15: Rückmeldungen bei Verbindungsoperationen	63
Abbildung 6-16: Tablettouch Rückmeldungen	64

Abbildung 7-1: Technologien von Microsoft Windows [57]	68
Abbildung 7-2: Prinzip des MVVM-Patterns	69
Abbildung 7-3: Aufbau der Anwendung	74
Abbildung 7-4: UML-Sequenzdiagramm – Tablet tapt auf Prozesselement des Tabletops	80
Abbildung 7-5: Dualität der Skizzeneingabe.....	83
Abbildung 7-6: Vereinfachung eines Polygons.....	85
Abbildung 7-7: Normalisierter Boxfilter ohne Randbehandlung.....	87
Abbildung 7-8: Orientierungshistogramme samt korrespondierender Skizzeneingabe	87
Abbildung 7-9: Drehen des Polygons und Berechnung der Bounding Box.....	87
Abbildung 7-10: Jordan Curve Theorem.....	89
Abbildung 7-11: Kräfte einzelner Knoten	90
Abbildung 8-1: Beispiel-Items des ISO-Norm 9241/10	99
Abbildung 8-2: Beispiel-Items des UEQ.....	100
Abbildung 8-3: Ablauf des Experiments	106
Abbildung 8-4: Box Plots der Kategorien des ISO-Norm 9241/10 Fragebogens	108
Abbildung 8-5: Box Plots der Dimensionen des UEQ Fragebogens	109
Abbildung 8-6: Box Plots der zusammengefassten Werte des Loggers.....	110
Abbildung 8-7 Modellierung an gegenüberliegenden Seiten des Tabletop- Systems.....	111
Abbildung 8-8 Ähnlichkeit von AND und XOR bei freier Orientierung	112
Abbildung 8-9: Verbessertes Menü mit Schließen-Button	115
Abbildung A-1: Seite 1 des ISO-Norm 9241/10.....	ix
Abbildung A-2: Seite 2 des ISO-Norm 9241/10.....	x
Abbildung A-3: Seite 3 des ISO-Norm 9241/10.....	xi
Abbildung A-4: Seite 4 des ISO-Norm 9241/10.....	xii
Abbildung A-5: Seite 5 des ISO-Norm 9241/10.....	xiii
Abbildung A-6: Seite 6 des ISO-Norm 9241/10.....	xiv
Abbildung A-7: Seite 7 des ISO-Norm 9241/10.....	xv
Abbildung A-8: Seite 8 des ISO-Norm 9241/10.....	xvi
Abbildung B-1: Seite 1 des User Experience Questionnaires.....	xvii
Abbildung B-2: Seite 2 des User Experience Questionnaires.....	xviii

Abbildung C-1: Seite 1 des Allgemeinen Fragebogens	xix
Abbildung C-2: Seite 2 des Allgemeinen Fragebogens	xx
Abbildung D-1: Seite 1 des Leitfadens für den Experimentleiter	xxi
Abbildung D-2: Seite 2 des Leitfadens für den Experimentleiter	xxii
Abbildung E-1: Einverständniserklärung.....	xxiii
Abbildung F-1: Anwendungsszenario – Bewerber	xxiv
Abbildung F-2: Anwendungsszenario - Unternehmen.....	xxv
Abbildung G-1: Prozessmodell der Probanden 1 und 2.....	xxvii
Abbildung G-2: Prozessmodell der Probanden 3 und 4.....	xxviii
Abbildung G-3: Prozessmodell der Probanden 5 und 6.....	xxix
Abbildung G-4: Prozessmodell der Probanden 7 und 8.....	xxx
Abbildung G-5: Prozessmodell der Probanden 9 und 10	xxxi
Abbildung G-6: Prozessmodell der Probanden 11 und 12.....	xxxii
Abbildung G-7: Prozessmodell der Probanden 13 und 14.....	xxxiii
Abbildung G-8: Prozessmodell der Probanden 15 und 16.....	xxxiv
Abbildung G-9: Prozessmodell der Probanden 17 und 18.....	xxxv
Abbildung G-10: Prozessmodell der Probanden 19 und 20.....	xxxvi

TABELLENVERZEICHNIS

Tabelle 5-1: Aktionen des Kontextmenüs.....	41
Tabelle 5-2: Regeln der Syntaxprüfung.....	49
Tabelle 7-1: Erkennung von Gesten.....	89
Tabelle 8-1: Dimensionen semantischer Qualität.....	100
Tabelle 8-2: Kategorien empfundener Qualität.....	101
Tabelle 8-3: Implementierungsfehler	116
Tabelle H-1: Daten des ISO-Norm 9241/10.....	xxxvii
Tabelle H-2: Daten des UEQ.....	xxxviii
Tabelle H-3: Daten des Fragebogens Allgemeine Fragen.....	xxxix
Tabelle H-4: Weitere Daten des Experiments	xl
Tabelle H-5: Feedback der Probanden	xlii

LITERATURVERZEICHNIS

- [1] J. Freund, „Geschäftsprozesse: Die zwölf wichtigsten Fragen zum Business-Process-Management“, IDG Business Media GmbH, 21 Juli 2008. [Online]. Available: <http://www.computerwoche.de/a/die-zwoelf-wichtigsten-fragen-zum-business-process-management,1869128>. [Zugriff am 10 Januar 2014].
- [2] B. Funk, J. Gómez, P. Niemeyer und F. Teuteberg, „Geschäftsprozess-integration mit SAP - Fallstudien zur Steuerung von Wertschöpfungsprozessen entlang der Supply Chain“, Heidelberg: Springer, 2010.
- [3] G. M. Olson und J. S. Olson, „Distance Matters“, in *Journal Human-Computer Interaction, Volume 15, Issue 2, S. 139-178*, 2000.
- [4] J. Kolb, B. Rudner und M. Reichert, „Towards Gesture-based Process Modeling on Multi-Touch Devices“, in *International Journal of Information System Modeling and Design, Volume 4 Issue 4, S. 280-293*, Heidelberg, Germany, 2012.
- [5] H. Wittern, „Empirical Study Evaluating Business Process Modeling on Multi-Touch Devices“, in *Proceedings of the 2012 IEEE International Conference on Software Science, Technology and Engineering (SWSTE '12), S. 20-29*, Washington, DC, USA, 2012.
- [6] J. Rekimoto und M. Saitoh, „Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99), S. 378-385*, New York, NY, USA, 1999.
- [7] B. Scholtz, A. Calitz und I. Snyman, „The Usability of Collaborative Tools: Application to Business Process Modelling“, in *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT '13), S. 347-358*, New York, NY, USA, 2013.

- [8] P. Rittgen, „Collaborative Modeling of Business Processes: A Comparative Case Study“, in *Proceedings of the 2009 ACM symposium on Applied Computing (SAC '09)*, S. 225-230, New York, NY, USA, 2009.
- [9] S. Oppl und C. Stary, „Facilitating Shared Understanding of Work Situations Using a Tangible Tabletop Interface“, in *Behaviour & Information Technology*, Volume 33, Number 6, S. 619-635, Bristol, PA, USA, 2013.
- [10] D. Wachholder und S. Oppl, „Interactive Coupling of Process Models: A Distributed Tabletop Approach to Collaborative Modeling“, in *Proceedings of the 2014 European Conference on Cognitive Ergonomics (ECCE '14)*, Artikel 31, 8 Seiten, Vienna, Austria, 2014.
- [11] J. Edelman, A. Grosskop, M. Weske und L. Leifer, „Tangible Business Process Modeling: A New Approach“, in *Proceedings of the 17th International Conference on Engineering Design, ICED. Volume 9*, S. 153-168, 2009.
- [12] R. Hardy und E. Rukzio, „Touch & Interact: Touch-based Interaction of Mobile Phones with Displays“, in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services (MobileHCI '08)*, S. 245-254, New York, NY, USA, 2008.
- [13] S. Döweling, T. Tahiri, B. Schmidt, A. Nolte und M. Khalibeigi, „Collaborative Business Process Modeling On Interactive Tabletops“, in *Proceedings of the 21st European Conference on Information Systems (ECIS '13), Research in Progress, Paper 29*, 2013.
- [14] J. Burkhardt, „Collaborative Process Modelling with Multi-Touch Tables“, Diploma Thesis, Ulm, Germany, 2013.
- [15] Object Management Group, „Business Process Model and Notation, Version 2.02“, 20 Januar 2014. [Online]. Available: <http://www.omg.org/spec/BPMN/2.0.2/>. [Zugriff am 3 Juli 2014].

- [16] E. Rukzio, „Physical Mobile Interactions: Mobile Devices as Pervasive Mediators for Interactions with the Real World“, München, 2006.
- [17] M. Dettweiler, „Computer „Surface“: Windows 7 ist vom Tisch - Technik & Motor - FAZ“, Frankfurter Allgemeine Zeitung, 27 Januar 2012. [Online]. Available: <http://www.faz.net/aktuell/technik-motor/computer-surface-windows-7-ist-vom-tisch-11627206.html>. [Zugriff am 28 Januar 2014].
- [18] D. Schmidt, F. Chehimi, E. Rukzio und H. Gellersen, „PhoneTouch: A Technique for Direct Phone Interaction on Surfaces“, in *Proceedings of the 23rd annual ACM symposium on User interface software and technology (UIST '10)*, S. 13-16, New York, NY, USA, 2010.
- [19] R. Hardy, E. Rukzio, P. Holleis und M. Wagner, „Mobile Interaction with Static and Dynamic NFC-based Displays“, in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services (MobileHCI '10)*, S. 123-132, New York, NY, USA, 2010.
- [20] B. Pfleging, E. d. C. Valderrama Bahamondez, A. Schmidt, M. Hermes und J. Nolte, „MobiDev: a Mobile Development Kit for Combined Paper-based and In-situ Programming on the Mobile Phone“, in *CHI '10 Extended Abstracts on Human Factors in Computing Systems (CHI EA '10)*, S. 3722-3738, New York, NY, USA, 2010.
- [21] C. H. Damm, K. M. Hansen und M. Thomsen, „Tool Support for Cooperative Object-oriented Design: Gesture-based Modelling on an Electronic Whiteboard“, in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '00)*, S. 518-525, New York, NY, USA, 2000.
- [22] S. Cheema, S. Gulwani und J. LaViola, „QuickDraw: Improving Drawing Experience for Geometric Diagrams“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, S. 1037-1064, New York, NY, USA, 2012.

- [23] R. Malaka, A. Butz und H. Hußmann, „Medieninformatik - Eine Einführung“, Pearson Studium, 2009.
- [24] P. Fitts, „The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement“, in *Journal of Experimental Psychology*, Volume 47, Number 6, S. 381-391, 1954.
- [25] S. D. Scott, K. D. Grant und R. L. Mandryk, „System Guidelines for Co-located, Collaborative Work on a Tabletop Display“, in *Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work (ECSCW'03)*, S. 159-178, Norwell, MA, USA, 2003.
- [26] R. Kruger und S. Carpendale, „Orientation and Gesture on Horizontal Displays“, in *UbiComp 2002 Workshop on Collaboration with Interactive Walls and Tables*, Gotenborg, Sweden, 2002.
- [27] P. Dragicevic und Y. Shi, „Visualizing and Manipulating Automatic Document Orientation Methods Using Vector Fields“, in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*, S. 65-68, New York, NY, USA, 2009.
- [28] L. Barnkow, „Kollaboration an Multitouch-Tischen: Visualisierung von Besitz bei Gruppenarbeit“, in *Master Thesis*, Hamburg, Germany, 2012.
- [29] G. Schiavo, G. Jacucci, T. Ilmonen und L. Gamberini, „Evaluating an Automatic Rotation Feature in Collaborative Tabletop Workspaces“, in *Proceeding CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*, S. 1315-1320, New York, NY, USA, 2011.
- [30] Microsoft Corporation, „Microsoft Surface 2.0 Design and Interaction Guide“, Juli 2011. [Online]. Available: <http://www.microsoft.com/en-us/download/details.aspx?id=26713>. [Zugriff am 14 Januar 2014].

- [31] C. Shen, K. Ryall, C. Forlines, A. Esenther, F. D. Vernier, K. Everitt, M. Wu, D. Wigdor, M. Ringel Morris, M. Hancock und E. Tse, „Informing the Design of Direct-Touch Tabletops“, in *IEEE Computer Graphics and Applications*, Volume 26, Issue 5, S. 36-46, Los Alamitos, CA, USA, 2006.
- [32] J. O. Wobbrock, M. Ringel Morris und A. D. Wilson, „User-defined Gestures for Surface Computing“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, S. 1083-1092, New York, NY, USA, 2009.
- [33] D. Kirk, S. Izadi, O. Hilliges, R. Banks, S. Taylor und A. Sellen, „At Home with Surface Computing?“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, S. 159-168, New York, NY, USA, 2012.
- [34] Y. Rogers und S. Lindley, „Collaborating Around Vertical and Horizontal Large Interactive Displays: Which Way is Best?“, in *Interacting with Computers*, Volume 16, Number 6, S. 1133-1152, 2004.
- [35] A. Fox, B. Johanson, P. Hanrahan und T. Winograd, „Integrating Information Appliances into an Interactive Workspace“, in *Journal IEEE Computer Graphics and Applications*, Volume 20, Issue 3, S. 54-65, Los Alamitos, 2000.
- [36] H. Suzuki und H. Kato, „Interaction-level Support for Collaborative Learning: AlgoBlock-an Open Programming Language“, in *The first international conference on Computer support for collaborative learning (CSCL '95)*, S. 349 - 355, 1995.
- [37] D. Schneider, J. Seifert und E. Rukzio, „MobIES: Extending Mobile Interfaces Using External Screens“, in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia (MUM '12)*, S. 1-2, Ulm, Germany, 2012.

- [38] G. Broll, J. Hamard, M. Paolucci, M. Haarländer, M. Wagner, S. Siorpaes, E. Rukzio, A. Schmidt und K. Wiesner, „Mobile Interaction with Web Services Through Associated Real World Objects“, in *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services (MobileHCI '07)*, S. 319-321, New York, NY, USA, 2007.
- [39] M. de Sá, J. Antin, D. Shamma und E. F. Churchill, „Mobile Augmented Reality: Video Prototyping“, in *Proceedings of CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*, S. 1897-1902, New York, NY, USA, 2011.
- [40] D. Baur, S. Boring und S. Feiner, „Virtual Projection: Exploring Optical Projection as a Metaphor for Multi-device Interaction“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, S. 1693-1702, New York, NY, USA, 2012.
- [41] Samsung , „Samsung GT-I8530“, 2014. [Online]. Available: <http://www.samsung.com/de/consumer/mobile-device/mobilephones/archive-mobile-phones/GT-I8530BAADB>T. [Zugriff am 3 September 2014].
- [42] P. Brandl, C. Forlines, D. Wigdor, M. Haller und C. Shen, „Combining and Measuring the Benefits of Bimanual Pen and Direct-touch Interaction on Horizontal Interfaces“, in *Proceedings of the working conference on Advanced visual interfaces (AVI '08)*, S. 154-161, New York, NY, USA, 2008.
- [43] M. Frisch, J. Heydekorn und R. Dachsel, „Diagram Editing on Interactive Displays using Multi-Touch and Pen Gestures“, in *Proceedings of the 6th international conference on Diagrammatic representation and inference (Diagrams'10)*, S. 182-196, Heidelberg, Germany, 2010.
- [44] K.-P. Yee, „Two-handed Interaction on a Tablet Display“, in *Proceedings of CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI EA '04)*, S. 1493-1496, New York, NY, USA, 2004.

- [45] D. Käser, M. Agrawala und M. Pauly, „FingerGlass: Efficient Multiscale Interaction on Multitouch Screens“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*, S. 1601-1610, New York, NY, USA, 2011.
- [46] Nuance, „Swype | About“, Nuance, [Online]. Available: <http://www.swypeinc.com/category/about/>. [Zugriff am 30 Januar 2014].
- [47] S. MacKenzie, „The Design and Evaluation of a High-performance Soft Keyboard“, in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99)*, S. 25-31, New York, NY, USA, 1999.
- [48] J. Kolb und M. Reichert, „A flexible approach for abstracting and personalizing large business process models“, in *ACM SIGAPP Applied Computing Review, Volume 13, Issue 1*, S. 6-18, New York, NY, USA, 2013.
- [49] T. M. Fruchterman und E. M. Reingold, „Graph-drawing by Force-directed Placement“, in *Software: Practice and experience, Volume 21, Issue 11*, S. 1129-1164, 1991.
- [50] Microsoft, „Windows 8 Design and coding guidelines - Updated for Windows 8.1“, 2014. [Online]. Available: http://download.microsoft.com/download/C/0/A/C0AEF0CC-B969-406D-989A-4CDAFDBB3F3C/W8_1_Guidlines.pdf. [Zugriff am 26 August 2014].
- [51] C. Moser, „User Experience Design“, Springer-Verlag, 2012.
- [52] J. Smith, „WPF Apps With The Model-View-ViewModel Design Pattern“, Microsoft, Februar 2009. [Online]. Available: <http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>. [Zugriff am 31 03 2014].
- [53] Microsoft, „Windows Presentation Foundation“, 2014. [Online]. Available: [http://msdn.microsoft.com/de-de/library/ms754130\(v=vs.110\).aspx](http://msdn.microsoft.com/de-de/library/ms754130(v=vs.110).aspx). [Zugriff am 2 September 2014].

- [54] Microsoft, „XAML Overview (WPF)“, 2014. [Online]. Available: [http://msdn.microsoft.com/de-de/library/ms752059\(v=vs.110\).aspx](http://msdn.microsoft.com/de-de/library/ms752059(v=vs.110).aspx). [Zugriff am 2 September 2014].
- [55] Microsoft, „Introducing Windows Presentation Foundation“, 2014. [Online]. Available: <http://msdn.microsoft.com/en-us/library/aa663364.aspx>. [Zugriff am 2 September 2014].
- [56] B. Dewey, „Getting Started with Metro Style Apps“, O’Reilly Media, Inc., 2012.
- [57] D. Meixner, „Windows 8 + Windows Phone 8 Cross Platform App Development mit C#/XAML“, Microsoft, 5 April 2013. [Online]. Available: <http://blogs.msdn.com/b/dmx/archive/2013/04/05/windows-8-windows-phone-8-cross-platform-app-development-mit-c-xaml-teil-1.aspx>. [Zugriff am 31 März 2014].
- [58] J. Bloch, „Effective Java“, Addison-Wesley, 2008.
- [59] Microsoft, „Data Binding Overview“, 2014. [Online]. Available: [http://msdn.microsoft.com/de-de/library/ms752347\(v=vs.110\).aspx](http://msdn.microsoft.com/de-de/library/ms752347(v=vs.110).aspx). [Zugriff am 2 September 2014].
- [60] Microsoft, „Microsoft PixelSense Design and Development“, 2014. [Online]. Available: <http://msdn.microsoft.com/en-US/windows/desktop/hh241326.aspx>. [Zugriff am 2 September 2014].
- [61] Microsoft, „Windows Communication Foundation“, 2014. [Online]. Available: [http://msdn.microsoft.com/de-de/library/dd456779\(v=vs.110\).aspx](http://msdn.microsoft.com/de-de/library/dd456779(v=vs.110).aspx). [Zugriff am 2 September 2014].
- [62] Microsoft, „Data Contract Serializer“, 2014. [Online]. Available: [http://msdn.microsoft.com/de-de/library/ms731072\(v=vs.110\).aspx](http://msdn.microsoft.com/de-de/library/ms731072(v=vs.110).aspx). [Zugriff am 3 September 2014].

- [63] J. Lowy, „WCF: Working with One-Way Calls, Callbacks, And Events“, Microsoft MSDN Magazine, 2014. [Online]. Available: <http://msdn.microsoft.com/en-us/magazine/cc163537.aspx#S2>. [Zugriff am 3 September 2014].
- [64] Max, „Stack Overflow“, 13 Juni 2010. [Online]. Available: <http://stackoverflow.com/questions/3032630/optimizing-simplifying-a-path/3032686#3032686>. [Zugriff am 14 Mai 2014].
- [65] J. Reaver, T. Stahovich und J. Herold, „How to make a Quick\$: Using Hierarchical Clustering to Improve the Efficiency of the Dollar Recognizer“, in *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling (SBIM '11)*, S. 103-108, New York, NY, USA, 2011.
- [66] J. O. Wobbrock, A. D. Wilson und Y. Li, „Gestures Without Libraries, Toolkits or Training: a \$1 Recognizer for User Interface Prototypes“, in *Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST '07)*, S. 159-168, New York, NY, USA, 2007.
- [67] W. R. Franklin, „PNPOLY - Point Inclusion in Polygon Test“, 21 Januar 2014. [Online]. Available: http://www.ecse.rpi.edu/Homepages/wrf/Research/Short_Notes/pnpoly.html. [Zugriff am 8 Mai 2014].
- [68] Wikipedia contributors, „Coulombsches Gesetz“, 28 Mai 2014. [Online]. Available: http://de.wikipedia.org/wiki/Coulombsches_Gesetz. [Zugriff am 9 September 2014].
- [69] F. Sarodnick und H. Brau, „Methoden der Usability Evaluation - Wissenschaftliche Grundlagen und praktische Anwendung“, Verlag Hans Huber, 2011.
- [70] J. Barjis, „CPI Modeling: Collaborative, Participative, Interactive Modeling“, in *Proceedings of the Winter Simulation Conference (WSC '11)*, S. 3099-3108, 2011.

- [71] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell und A. Wesslén, „Experimentation in Software Engineering - An Introduction“, Kluwer Academic Publishers, Norwell, MA, USA, 2000.
- [72] O. I. Lindland, G. Sindre und A. Sölvberg, „Understanding Quality in Conceptual Modeling“, in *IEEE Software, Volume 11, Number 2, S. 42-49*, 1994.
- [73] J. Prümper, „ergo-online® - Fragebogen ISONORM“, 7 August 2012. [Online]. Available: http://www.ergo-online.de/site.aspx?url=html/software/verfahren_zur_beurteilung_der/fragebogen_isonorm_online.htm. [Zugriff am 1 August 2014].
- [74] International Organization for Standardization, „ISO 9241-11“, 1998. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=16883. [Zugriff am 5 Juli 2014].
- [75] A. Hinderks, „UEQ-Online“, [Online]. Available: <http://www.ueq-online.org/>. [Zugriff am 17 Juli 2014].
- [76] P. Rittgen, „Quality and Perceived Usefulness of Process Models“, in *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10)*, S. 65-72, New York, NY, USA, 2010.
- [77] V. Gruhn und R. Laue, „Complexity Metrics for Business Process Models“, in *In Proceedings of the ninth International Conference on Business Information Systems (BIS'2006)*, S. 1–12, Klagenfurt, Austria, 2006.
- [78] A. Robertson, „Microsoft rebrands original Surface table-based touchscreen as PixelSense“, 19 Juni 2012. [Online]. Available: <http://www.theverge.com/2012/6/19/3096652/microsoft-surface-table-rebranded-as-pixelsense>. [Zugriff am 1 August 2014].

- [79] H. Reijers und J. Mendling, „A Study Into the Factors That Influence the Understandability of Business Process Models“, in *IEEE Trans. on Systems Man and Cybernetics, Volume 41, Number 3, S. 449-462, 2011.*

NAME

Sebastian Ronis

MATRIKELNUMMER

680536

ERKLÄRUNG

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den _____

Datum

Sebastian Ronis