

Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Universitätsstraße 38
70569 Stuttgart

Diplomarbeit Nr. 3698

**Entwurf einer zentralen und generischen
Schnittstelle für das intuitive Auffinden
verteilter Funktionalität in föderierten
Informationssystemen**

Philipp Pirrung

Matrikel-Nr.: 2526016
Studiengang: Softwaretechnik
Prüfer: Prof. Dr. Albrecht Schmidt
Betreuer: Doris Janssen, M. Sc.
Dipl.-Inf. (FH) Jörg Metzler
Dipl.-Inf. Uli Held
Beginn: 10.11.2014
Ende: 12.05.2015
CR-Klassifikation: H.5.2

Kurzfassung

Bei der Dr. Ing. h.c. F. Porsche AG wird seit vielen Jahren ein föderiertes Informationssystem verwendet, welches verschiedene, verteilte und autonome Datenquellen zusammenführt und an einer zentralen Stelle verfügbar macht. Das System wird täglich von einer Vielzahl an Benutzern verwendet um auf die bereitgestellten Daten zuzugreifen und diese für die tägliche Arbeit aufzubereiten und auszuwerten. Die Funktionalitäten sind dabei im ganzen System verteilt und an den zugehörigen Datenobjekten bereitgestellt. Da die Benutzer in den meisten Fällen wissen, welche Funktionalitäten sie benötigen, ist eventuell eine zentrale Anlaufstelle zur Bereitstellung dieser sinnvoll.

Im Rahmen einer Neuentwicklung des föderierten Informationssystems sollen in dieser Diplomarbeit Teile der Anwendung im Hinblick auf ihre Benutzbarkeit untersucht und verbessert werden. Hierfür sollen zunächst Probleme der Benutzbarkeit analysiert und Lösungsansätze zu ihnen erstellt und bewertet werden. Die vielversprechendsten Lösungsansätze sollen im Anschluss umgesetzt und das entstandene System evaluiert werden.

Abstract

For many years, the Dr. Ing. h.c. F. Porsche AG has been using a federated database system to consolidate numerous, distributed and autonomous data sources and make them available in one centralized place. The system is in daily use by many users to access, edit and evaluate the provided data. All functionality is spread out over the whole system and is only available for use at the site of the associated data objects. Since users typically know which functionality they're looking for when using the system, a centralized interface for accessing that functionality may be more fitting.

The goal of this diploma thesis is to analyse and improve the usability of parts within the scope of a re-development of the system. For this purpose, problems in the usability must first be analysed and suitable approaches to resolving them must be found and evaluated. The most promising approaches shall then be implemented and the usability of the resulting system evaluated.

Die vorliegende Abschlussarbeit enthält zum Teil Informationen, die nicht für die Öffentlichkeit bestimmt sind. Alle Rechte an der Abschlussarbeit, einschließlich der Verbreitung auf elektronischen Medien, liegen bei der Softwarehelden GmbH & Co. KG.

Abweichend hiervon darf der Inhalt der Abschlussarbeit während einer Sperrzeit von 5 Jahren ab dem Abgabedatum nur mit der ausdrücklichen schriftlichen Genehmigung der Softwarehelden GmbH & Co. KG an Dritte weitergegeben werden. Nach Ablauf der Sperrzeit ist diese Genehmigung nicht mehr erforderlich.

Inhaltsverzeichnis

1 Einleitung.....	1
1.1 Problemdarstellung, Ziele und Aufgaben.....	1
1.2 Gliederung	2
2 Grundlagen.....	4
2.1 Föderierte Informationssysteme	4
2.1.1 Architektur	4
2.1.2 Benutzeroberfläche	6
2.2 Usability Engineering	9
2.2.1 Ermittlung von Anforderungen.....	9
2.2.2 Konzeptentwurf	12
2.2.3 Prototyping	13
2.2.4 Usability Evaluation	15
2.2.5 Evaluation des konzeptionellen Entwurfs	20
2.2.6 Screen Design Standards.....	21
2.2.7 Detaillierter Entwurf der Benutzeroberfläche	22
2.2.8 Semantisches Differential.....	23
2.2.9 NASA Task Load Index	24
2.3 Natural Language Processing	26
2.3.1 Damerau-Levenshtein-Distanz zur Einstufung der Qualität von Resultaten	26
2.3.2 N-Gramme.....	27
2.3.3 Part of Speech Tagging.....	31
3 Analyse	38
3.1 Usability Spezifikation.....	38
3.1.1 Anforderungen der Nutzer	38
3.1.2 Anforderungen der Entwickler	42
3.2 Anwendungsfälle	45
4 Lösungsansätze/Konzepte.....	50
4.1 Filtern von Objekten.....	50
4.2 Navigation und Darstellungen innerhalb des Systems.....	50
4.3 Schnellnavigation innerhalb des Systems	51
4.4 Navigieren zu Objekten	52

4.5 Generische Oberfläche	52
4.6 Generische Aktionen	53
4.7 Schnittstelle zur Lokalisierung von Funktionalität	53
4.8 Bewertung	54
5 Evaluation anhand einer prototypischen Umsetzung	55
5.1 Vorgehen	55
5.2 Aufbau der Nutzergruppen	57
5.3 Prototypische Umsetzung der Lösungsansätze	58
5.3.1 Generische Oberfläche	59
5.3.2 Navigation innerhalb des Systems	62
5.3.3 Schnellnavigation innerhalb des Systems	63
5.3.4 Generische Aktionen	64
5.3.5 Schnittstelle zur Lokalisierung von Funktionalitäten	65
5.4 Ergebnisse von informellen Evaluationen der ersten Iteration	66
5.4.1 Navigation/Schnellnavigation und Darstellung innerhalb des Systems	66
5.4.2 TellMe-Auswahl	67
5.5 Umsetzung der Ergebnisse der ersten informellen Evaluation	67
5.5.1 Navigation/Schnellnavigation und Darstellung innerhalb des Systems	67
5.5.2 TellMe-Eingabeleiste	69
5.6 Ergebnisse von informellen Evaluationen der zweiten Iteration	70
5.6.1 Navigation/Schnellnavigation und Darstellung innerhalb des Systems	70
5.6.2 TellMe-Eingabeleiste	71
5.7 Umsetzung der Ergebnisse der zweiten informellen Evaluation	72
5.7.1 Schnellnavigation innerhalb des Systems	72
5.7.2 TellMe-Eingabeleiste	72
5.8 Ergebnisse der abschließenden Evaluation	74
5.8.1 Allgemeine Ergebnisse	74
5.8.2 Erlernbarkeit der Anwendung	76
5.8.3 Belastung der Anwendung nach NASA-TLX	76
5.8.4 Eindruck des getesteten Systems	77
5.8.5 Ergebnisse der Evaluation mit Entwicklern	79
6 Fazit	81
7 Anhang	83
8 Literaturverzeichnis	143

Abbildungsverzeichnis

Abbildung 1 Architektur von Cleverle (nach (Held, 2009))	5
Abbildung 2 Wireframe der Cleverle-Navigation (Teil 1)	7
Abbildung 3 Wireframe der Cleverle-Navigation (Teil 2)	8
Abbildung 4 Damerau-Levenshtein-Distanz am Beispiel der Wörter "Fahrzeug" und "Fhareuzfg"	27
Abbildung 5 Zeile eines Fahrzeugs in Cleverle (Vereinfacht)	41
Abbildung 6 Visualisierte Block-Metapher.....	51
Abbildung 7 Altersverteilung von aktiven Cleverle-Benutzern.....	58
Abbildung 9 Datenmodell der Navigation mit Anwendungen und Navigationsknoten	62
Abbildung 10 Datenmodell der Favoriten.....	63
Abbildung 11 Datenmodell der Aktionen.....	64
Abbildung 12 Objektansicht in Cluu als Wireframe	68
Abbildung 13 Navigationselemente in Cluu als Wireframe	69
Abbildung 14 Klassenmodell der TellMe-Textevalution	73
Abbildung 15 TellMe-Popup für Eingabetext "eine Abteilung".....	75
Abbildung 16 NASA-TLX von Benutzern ohne vorherige Erfahrungen mit Cleverle	77
Abbildung 17 Bewertung von Cleverle (links) und Cluu (rechts) aller Testnutzer	78
Abbildung 18 Bewertung von Cluu von Benutzern ohne vorherige Erfahrungen mit Cleverle	78
Abbildung 19 NASA-TLX von Entwicklern	80

Tabellenverzeichnis

Tabelle 1 Für semantisches Differential verwendete Gegensatzpaare	24
Tabelle 2 Bewertungskriterien des NASA-TLX	25
Tabelle 3 Anwendungsfall "Funktionalität ändern"	46
Tabelle 4 Anwendungsfall „Personalliste exportieren“	47
Tabelle 5 Anwendungsfall "Fahrzeug reservieren"	48
Tabelle 6 Anwendungsfall "Anwendungsstruktur erstellen"	49
Tabelle 7 Bewertung von Lösungsansätzen	54
Tabelle 8 Verwendete Hauptansichten	60
Tabelle 9 Verwendete Unteransichten	61

1 Einleitung

Anwendungslandschaften in Unternehmen stellen heute das Resultat jahrelang gewachsener Strukturen dar. Die eingesetzten Anwendungen sind dabei häufig ohne eine architektonische Gesamtplanung im Sinne eines Bottom-Up-Prozesses auf Abteilungsebene entstanden. Das Resultat sind heterogene und meist isolierte Systeme, die unterschiedliche Bedienoberflächen haben und zwischen denen keine standardisierte Kommunikation existiert. Die Zusammenführung von Anwendungen zur optimalen Unterstützung der Firmenprozesse stellt darum heutzutage eine wesentliche Herausforderung in größeren Unternehmen dar.

1.1 Problemdarstellung, Ziele und Aufgaben

Durch die Vereinheitlichung und Zusammenführung der unterschiedlichen Teilsysteme entsteht jedoch ein neues Problem. Dem Benutzer stehen riesige Mengen an Daten zur Verfügung, die alle ihre eigenen Interaktionsmöglichkeiten bieten. Da diese Interaktionsmöglichkeiten über die einzelnen Teilsysteme verteilt sind, ist es für den Benutzer nur schwer möglich, alle für ihn relevanten Funktionalitäten des Systems zu kennen und folglich zu nutzen. Auch wenn der Benutzer zu Beginn seiner Sitzung schon weiß, welche Funktionalitäten des Systems er verwenden möchte, so muss er diese dennoch jedes Mal lokalisieren und an der entsprechenden Stelle ausführen. Durch dieses Vorgehen entstehen dem Benutzer ein deutlich höherer zeitlicher Aufwand, sowie eine höhere mentale Belastung bei der Verwendung des Systems. Wäre es für den Benutzer jedoch möglich, die gesuchte Funktionalität an einer zentralen Stelle gebündelt abzurufen, so könnte er die Anwendung möglicherweise effizienter und mit weniger Aufwand benutzen.

Für eine Untersuchung der beschriebenen Probleme wird in dieser Diplomarbeit das bei der Dr. Ing. h.c. F. Porsche AG eingesetzte föderierte Informationssystem Cleverle verwendet. Bei Porsche wurde die Notwendigkeit einer Integrationsplattform schon frühzeitig erkannt. Bereits vor mehreren Jahren wurde ein

serviceorientiertes Integrationssystem entwickelt, mit welchem verteilte und heterogene Datenbanken und Anwendungen strukturiert zusammengeführt werden können.

Im Rahmen einer grundlegenden Neuentwicklung des föderierten Informationssystems soll auch die Benutzbarkeit analysiert und überarbeitet werden. Das bietet die Möglichkeit, in einer neuen, generischen Benutzeroberfläche die beschriebene zentrale Schnittstelle zur Auffindung von verteilten Funktionalitäten zu integrieren.

Ziel der Diplomarbeit ist es, eine weitgehend generische Benutzerschnittstelle zu entwerfen, welche sowohl den zentralen, als auch den klassischen, verteilten Ansatz unterstützt.

Die besondere Herausforderung besteht darin, diese Schnittstelle möglichst intuitiv, evtl. sogar mit natürlich sprachlichen Elementen zu gestalten, Benutzeingaben zu interpretieren und auf vorhandene Businessfunktionalität abzubilden.

Zuerst sollen die existierenden Probleme der Benutzer und Entwickler des Systems erfasst und analysiert werden. Auf Basis dieser Ergebnisse werden anschließend Lösungsansätze erarbeitet, die diese Probleme beheben und eine bessere Benutzbarkeit des Systems ermöglichen sollen. Die so entstandenen Konzepte zum Aufbau der Oberfläche sollen iterativ umgesetzt, bewertet und kontinuierlich verbessert werden. Abschließend sollen die umgesetzten Konzepte evaluiert und eine Empfehlung für weitere Verbesserungen der Benutzbarkeit gegeben werden.

1.2 Gliederung

Die Gliederung dieser Diplomarbeit ist an die Vorgehensweise bei deren Durchführung angelehnt.

Das folgende Kapitel befasst sich mit den Grundlagen die einen Einfluss auf die Durchführung und Ergebnisse dieser Diplomarbeit hatten. Zuerst erhält der Leser einen Überblick über die Besonderheiten von föderierten Informationssystemen. Neben der grundlegenden Vorgehensweise im Usability Engineering, werden verschiedene Aspekte des Natural Language Processing erläutert.

Die genauere Analyse von vorhandenen benutzbarkeitsbezogenen Probleme und Wünsche von Benutzern wird in Kapitel 3 beschrieben. Es werden sowohl die Anforderungen von Benutzern, als auch Entwicklern erfasst.

Die erarbeiteten Lösungsansätze zur Optimierung der Benutzbarkeit werden im darauffolgenden Kapitel erläutert und anschließend anhand eines Kosten-Nutzen-Verhältnisses bewertet.

In Kapitel 5 werden die vielversprechendsten Lösungsansätze weiter verfolgt und prototypisch umgesetzt. Der so entstandene Prototyp wird während des Entwicklungsprozesses iterativ evaluiert und mithilfe der so gesammelten Erkenntnisse und Ergebnisse kontinuierlich verbessert. Abschließend wird eine formale Evaluation des resultierenden Systems durchgeführt und die Ergebnisse analysiert.

Das letzte inhaltliche Kapitel zieht ein Fazit aus dem Verlauf der Diplomarbeit. Es werden die Erkenntnisse und Ergebnisse zusammengefasst und bewertet, sowie eine Empfehlung für zukünftige Optimierungsmaßnahmen gegeben.

Nach den Ergebnissen der Evaluation wird die verwendete Literatur aufgeführt.

2 Grundlagen

Diese Diplomarbeit befasst sich im Wesentlichen mit drei Themen. Die Grundlage aller Überlegungen und Ergebnisse sind dabei föderierte Informationssysteme. Für diese soll eine zentrale Schnittstelle zur Auffindung der verteilten Funktionalitäten entworfen und entwickelt werden. Um eine gute Benutzbarkeit zu gewährleisten wird die Entwicklung der Schnittstelle, sowie der restlichen Oberfläche nach den Konzepten des Usability Engineering durchgeführt. Für die Umsetzung der Funktionalität der Schnittstelle werden des Weiteren Vorgehensweisen aus dem Natural Language Processing angewandt. Die Grundlagen der erwähnten Themen sind in diesem Kapitel beschrieben.

2.1 Föderierte Informationssysteme

Ziel und Aufgabe von föderierten Informationssystemen ist die Zusammenführung von verteilten, heterogenen und autonomen Datenquellen um diese an einer einzigen, zentralen Stelle zur Verfügung zu stellen (Busse, et al., 1999). Die Entwickler und Benutzer müssen sich demnach nicht um den Aufbau der angebundenen Teilsysteme kümmern, sondern können auf die verfügbaren Daten über eine vereinheitlichte Schnittstelle zugreifen.

Anhand des bei der Dr. Ing. h.c. F. Porsche AG entwickelten und eingesetzten föderierten Informationssystems Cleverle werden die grundlegenden, für diese Diplomarbeit relevanten Konzepte im Folgenden beschrieben.

2.1.1 Architektur

Cleverle ist ein bei der Dr. Ing. h.c. F. Porsche AG eingesetztes föderiertes Informationssystem. Es wurde seit 1998 von Ulrich Held privat entwickelt um unterschiedliche und verteilte Datenquellen zusammenzuführen und somit an einer zentralen Stelle vereinheitlicht bereitzustellen. Eine vereinfachte Darstellung der Architektur von Cleverle ist in Abbildung 1 veranschaulicht.

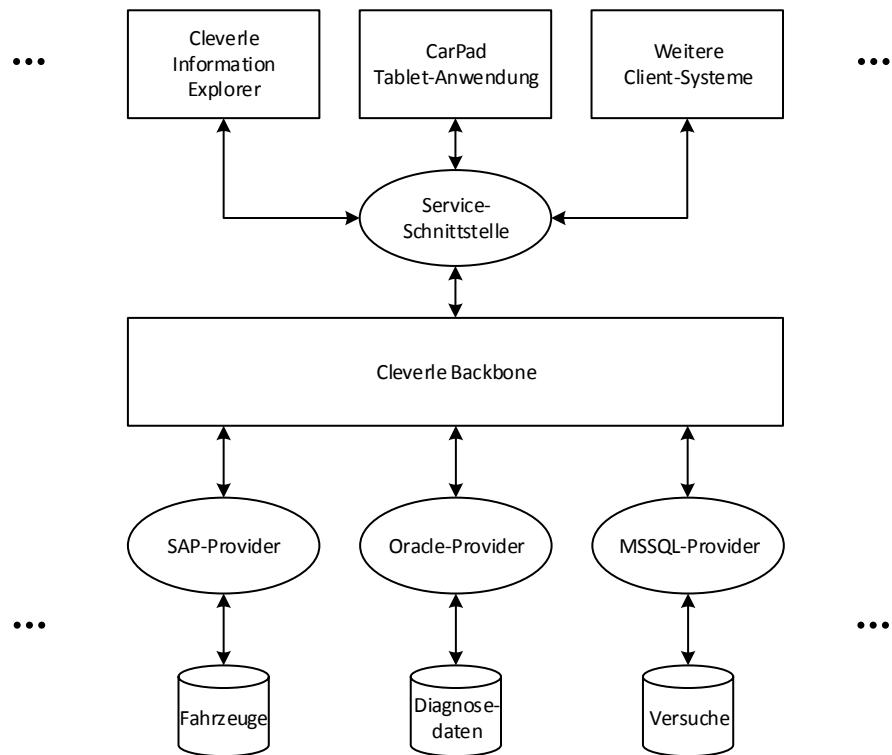


Abbildung 1 Architektur von Cleverle (nach (Held, 2009))

Im Wesentlichen besteht das System Cleverle aus zwei Komponenten (Held, 2009):

Cleverle Information Backbone (CIB)

Das Backbone des föderierten Informationssystems Cleverle bietet Basisoperationen zum Erstellen, Lesen, Bearbeiten und Löschen (CRUD) von Datenobjekten. Über diese Operationen können beliebige Daten, welche über eine Vielzahl von unabhängigen Datenquellen angebunden sind, verwaltet werden. Sie stehen somit für andere Systeme an einer zentralen Stelle zur Verfügung. Erweiterte Funktionalität wie das Erstellen von Berichten kann nur mittels einer Verwendung der CRUD-Operationen umgesetzt werden, was eine separate Implementierung für jedes Client-System bedeutet. Um dieselbe Funktionalität auf unterschiedlichen Client-Systemen zu verwenden, muss diese wegen der Abtrennung der Funktionalität vom Backend separat implementiert oder eingebunden werden.

Cleverle Information Explorer (CIE)

Der CIE ist eine Webanwendung welche die für den Benutzer ersichtliche Schnittstelle zum CIB darstellt. Über sie kann der Benutzer auf alle angebundenen Datenquellen zugreifen, die darin hinterlegten Daten einsehen, bearbeiten und löschen. Die Benutzeroberfläche des CIE ist in 2.1.2 beschrieben.

2.1.2 Benutzeroberfläche

Die in Cleverle dargestellten Daten werden im System als Klassen verwaltet. Dieses Datenmodell wird verwendet um die Einsicht in die verfügbaren Daten zu ermöglichen, indem es vorgibt welche Eigenschaften für die jeweiligen Klassen in einem gewissen Kontext angezeigt werden sollen. Für die Darstellung der angebundenen Daten verwendet Cleverle hierbei zwei grundlegende Ansichten. Die Listenansicht stellt alle zu einer Klasse gehörenden Datenobjekte in tabellarischer Form dar. Durch ein Öffnen der einzelnen Datenobjekte wird die Objektansicht des entsprechenden Datenobjekts geöffnet. In dieser Objektansicht werden sowohl die detaillierten Eigenschaften, als auch Relationen der Datenobjekte dargestellt. Diese Relationen verbinden verschiedene Datenobjekte ähnlich wie in relationalen Datenbanken mittels Primär- und Fremdschlüssel und können als 1:1, 1:n und n:m-Beziehungen angegeben werden. Für die Navigation innerhalb des Systems werden sogenannte SiteMap-Knoten verwendet (siehe Abbildung 2). Diese verweisen mittels URL auf andere SiteMap-Knoten, Listen- oder Objektansichten.

Die Benutzeroberfläche verfügt über verschiedene Kopfzeilen, die jeweils Funktionalitäten unterschiedlicher Art bieten (siehe Abbildung 2). Die oberste Kopfzeile dient als Darstellung der Anwendungsinformationen. Hier werden der Name der Anwendung, die verwendete Version, sowie Benutzername und Abmeldefunktion angezeigt. Unter der ersten Kopfzeile wird der Navigationspfad bis zu den derzeitig dargestellten Daten angezeigt. Mittels dieser Information kann der Benutzer zu einer der aufgelisteten Stellen in der Navigationshierarchie springen. Die nächste Kopfzeile bietet einige Funktionen zur Personalisierung der Anwendung (Speichern, Ändern, Zurücksetzen), sowie eine Hilfs- und eine Suchfunktion. Die nächste Kopfzeile gibt die derzeitige Position des Benutzers in der Navigationshierarchie als Überschrift wieder. Sobald der Benutzer auf eine Instanz eines Datenobjekts oder in eine Listenansicht einer Klasse navigiert (siehe

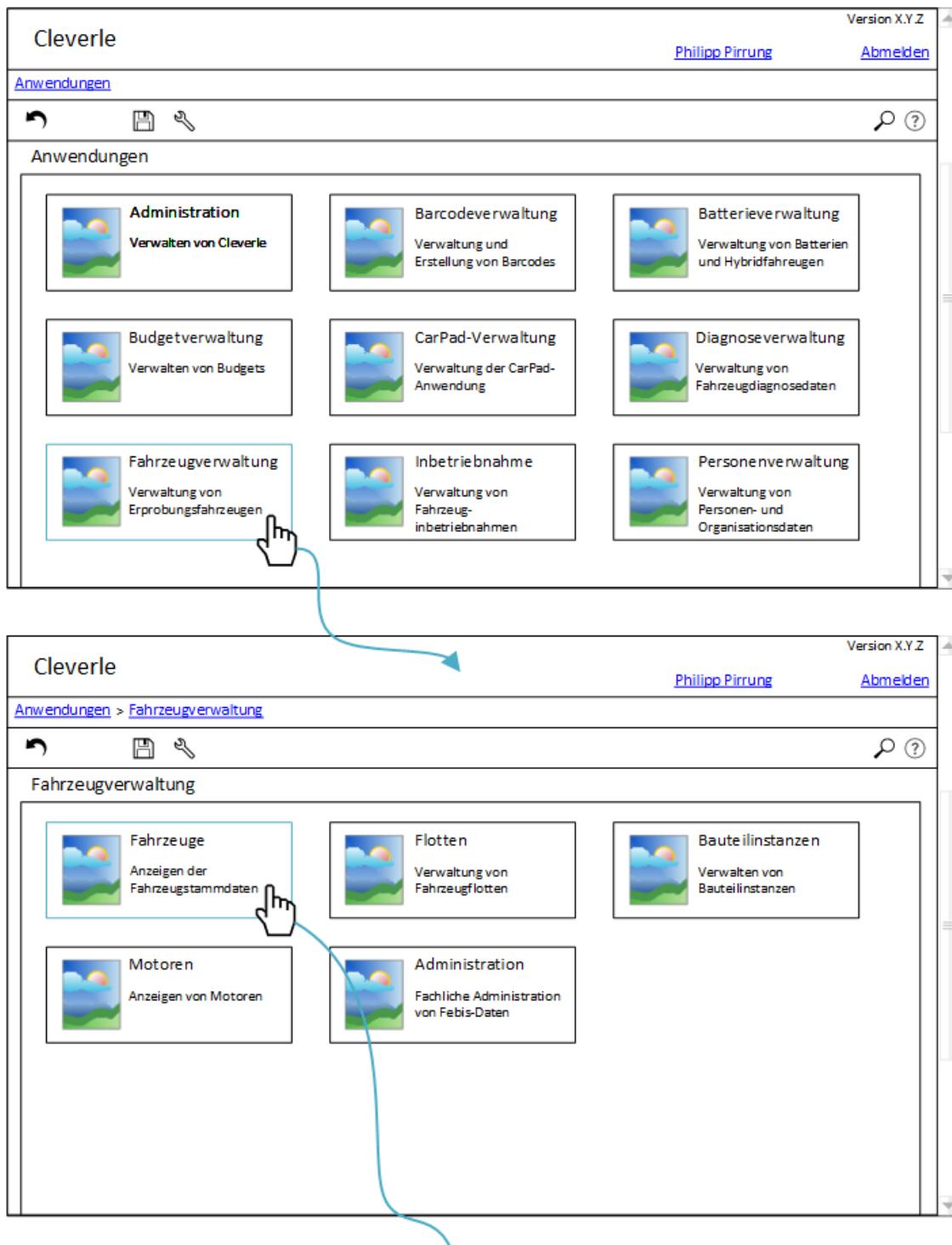


Abbildung 2 Wireframe der Cleverle-Navigation (Teil 1)

Abbildung 3), kommt eine weitere Kopfzeile hinzu. Diese bietet eine Auswahl von Aktionen auf den einzelnen, ausgewählten Datenobjekt-Instanzen an. Zusätzlich wird in der Listenansicht der Klassen eine Liste an möglichen Filtern eingeblendet, die es dem Benutzer ermöglichen, die angezeigten Daten einzuschränken. In der Objektansicht einer Datenobjekt-Instanz wird des Weiteren eine Aufzählung an Registern angezeigt, welche im ersten dargestellten Register die Details

The wireframe illustrates the Cleverle navigation interface, divided into two main sections: a list view and a detail view.

List View (Top):

- Header:** Cleverle, Version X.Y.Z, Philipp Pirrun, Abmeklen.
- Breadcrumbs:** Anwendungen > Fahrzeugverwaltung > Fahrzeuge.
- Toolbar:** Back, Save, Cut, Search, Help.
- Section:** Fahrzeuge.
- Filter:** Bez., Amtl. K., Fahr-Gst-Nr., Modelljahr.
- Actions:** Aktionen, Ansicht.
- Table:** Displays a list of vehicles with columns: Bezeichnung, Amtl. Kennzeichen, Fahrgestellnummer, Modelljahr. A blue arrow points to the row for '991 4S'.

	Bezeichnung	Amtl. Kennzeichen	Fahrgestellnummer	Modelljahr
<input type="checkbox"/>	Panamera Hybrid	LB-EN 2454	ABC123DEF456GHI78	2007
<input type="checkbox"/>	991 4S	LB-IE 5324	ABC123DEF456GHI78	2018
<input type="checkbox"/>	981 GTS	S-KY 8392	ABC123DEF456GHI78	2008
<input type="checkbox"/>	Cayenne Turbo	LB-TE 3729	ABC123DEF456GHI78	2015
<input type="checkbox"/>	Panamera	BB-BG 28	ABC123DEF456GHI78	2007
<input type="checkbox"/>	991 4S	BB-KE 5793	ABC123DEF456GHI78	2007
<input type="checkbox"/>	Macan S	S-GO 213	ABC123DEF456GHI78	2008
<input type="checkbox"/>	Cayenne S	BB-GF 254	ABC123DEF456GHI78	2015
<input type="checkbox"/>	Panamera GTS	S-PO 269	ABC123DEF456GHI78	2008
<input type="checkbox"/>	918S	BB-RF 3798	ABC123DEF456GHI78	2013
<input type="checkbox"/>	981S	BB-TM 201	ABC123DEF456GHI78	2014

Detail View (Bottom):

- Header:** Cleverle, Version X.Y.Z, Philipp Pirrun.
- Breadcrumbs:** Anwendungen > Fahrzeugverwaltung > Fahrzeuge > Fahrzeug.
- Toolbar:** Schnellsprung zu: 991 4S, Save, Cut, Search, Help.
- Section:** Fahrzeug.
- Actions:** Aktionen.
- Tab Bar:** Fahrzeug Stammdaten, Fahrzeugkonfiguration / Ausstattungen, Versuche, Reifenprofil-Messungen.
- Sub-Actions:** Hauptgruppe, Motor, Getriebe, Karosserie, Verwaltung, Sonstiges.
- Table:** Displays vehicle details in a grouped list.

Fahrgestellnummer	ABC123DEF456GHI78
Equipmentnummer	00012469204
Bezeichnung	991 4S
Aktuelle Zulassung	LB-IE 5324
Modelljahr	2018
Hersteller	Porsche
Typ	991

Abbildung 3 Wireframe der Cleverle-Navigation (Teil 2)

der geöffneten Datenobjekt-Instanz enthält. Die übrigen Register beinhalten andere, zu der Datenobjekt-Instanz in Relation stehende Datenobjekte. Innerhalb der Detailübersicht werden die Eigenschaften des Datenobjekts in einer untergeordneten Liste an Registern gruppiert.

Um Anwendungsfälle von Fachabteilungen realisieren zu können, wird das System kontinuierlich ausgebaut. Hierfür werden einzelne Anwendungen mit einer gleichbleibenden Anwendungsstruktur geplant, entwickelt und im CIE eingebunden. Die so entstandenen Anwendungen (siehe Abbildung 2) werden den Benutzern an einer zentralen Stelle auf der Hauptseite des Systems angeboten. Innerhalb einer Anwendung befinden sich zu dem jeweiligen Themengebiet gehörende Daten, welche durch Navigationsknoten der gleichen Darstellung visualisiert werden.

Die Metadaten der Anwendungen, welche die angezeigten Daten beschreiben, können zur Laufzeit verwaltet werden, wodurch neue Anwendungen mit relativ geringem Aufwand erstellt werden können. Ein großer Vorteil der konsistenten Darstellung der Daten ist, dass Benutzer aus allen Themenbereichen auch die Anwendungen anderer Themenbereiche verwenden können ohne sich in die Bedienung der Anwendung einarbeiten zu müssen. Problematisch ist jedoch die durch das Datenmodell vorgegebene Bindung der erweiterten, über einfaches Einsehen und Bearbeiten hinausgehenden, Funktionalitäten.

2.2 Usability Engineering

Anwendungen so zu gestalten, dass sie eine gute Benutzbarkeit haben ist ein Problem das bei jeder Entwicklung eine Rolle spielt. Eine Anwendung, die den späteren Benutzer frustriert, oder mit der er seine Aufgaben nur langsamer lösen kann als mit anderen Mitteln, hat ihr eigentliches Ziel verfehlt. Anwendungen sollten daraufhin ausgelegt sein die Arbeit ihrer Benutzer zu vereinfachen und weniger belastend zu gestalten.

Das Usability Engineering befasst sich mit dieser Thematik, indem die Benutzbarkeit der entwickelten Anwendung strukturiert entworfen, umgesetzt und getestet wird. Hierbei ist eine direkte Einbindung der Benutzer wünschenswert und von zentraler Bedeutung.

2.2.1 Ermittlung von Anforderungen

Um die Benutzbarkeit einer Anwendung überhaupt angemessen gestalten zu können, ist es für das Entwicklungsteam von großem Interesse, die Anforderungen, welche an die Benutzbarkeit gestellt werden, zu erfassen. Hierbei gilt es zu

entscheiden, inwiefern die Anwendung auf Ease-Of-Use oder Ease-Of-Learning ausgelegt werden soll. Ease-Of-Learning, d.h. wenig Lernaufwand, oder einfache Erlernbarkeit bei der Verwendung der Anwendung, ist speziell dann von Relevanz, wenn der Benutzer eine Anwendung benutzen soll ohne vorher eine Einweisung in diese erhalten zu haben. Dies ist beispielsweise bei Anwendungen wie Fahrkarten- oder Getränkeautomaten der Fall, die einen einfachen Vorgang automatisieren sollen. Wenn die Anwendung jedoch eine komplexere Aufgabe erledigen soll, ist womöglich eine Fokussierung auf Ease-Of-User, d.h. eine comfortable, bzw. einfache Benutzung der Anwendung von höherer Bedeutung (Mayhew, 1999). Gleiches gilt, wenn vorausgesetzt werden kann, dass der Benutzer eine Einarbeitung in die Anwendung erhält, oder Vorkenntnisse über deren Benutzung hat. Für die Erfassung der Anforderungen gibt es unterschiedliche Möglichkeiten (Wixon, et al., 1997):

Analyse von Konkurrenzsystemen

Eigenschaften, die bei der Verwendung von Konkurrenzsystemen eine gute Benutzbarkeit gewähren, können auch in der eigenen Anwendung für eine gute Benutzbarkeit eingesetzt werden.

Informationen des technischen Supports

Dem technischen Support – speziell bei einer Weiter- oder Neuentwicklung von Vorgängersystemen – stehen große Mengen an Informationen darüber zur Verfügung, wie die Benutzer über die Anwendung denken. Sie können Auskunft darüber geben, wo sich in der aktuellen Anwendung die größten Probleme bei der Benutzung befinden.

Erfahrungswerte

Aus der Entwicklung von anderen Anwendungen können Erfahrungen, deren Erkenntnisse zur besseren Benutzbarkeit beitragen, eingebracht werden.

Benutzerbefragungen

Die Wünsche der Benutzer können auch während der Analysephase direkt erfragt werden. Während der Evaluationsphasen des Usability Engineerings werden die Wünsche der Benutzer aber ohnehin durch die direkte Einbindung in der Evaluation berücksichtigt.

Außer als Entwurfsgrundlage können die Anforderungen gleichzeitig auch als Grundlage für die Evaluation dienen, indem die Benutzbarkeit anhand der Spezifikationen überprüft und eingestuft wird. Bei Nichterfüllung von Anforderungen können diese gezielt angegangen und verbessert werden, wodurch die Entwicklungskapazitäten auf die Behebung der Inkonsistenzen zwischen Spezifikation und Evaluationsergebnissen gelegt werden können. Dies spielt speziell bei iterativen Entwicklungsvorgehen eine zentrale Rolle, da hier in jeder Iteration die Resultate bewertet und in der nächsten Iteration verbessert werden.

Anforderungen an die Anwendung können grob in Kategorien aufgeteilt werden:

Qualitative Usability-Anforderungen geben eine grundsätzliche Richtung vor, in die sich die Anwendung entwickeln soll. Beispielsweise soll eine zu entwickelnde Anwendung den Benutzer bei der Einarbeitung unterstützen. Da diese Anforderungen jedoch nicht durch exakt messbare Faktoren beschrieben werden, kann deren Erfüllungsgrad nur schwer bestimmt werden.

Quantitative Usability-Anforderungen geben genaue, objektive und messbare Ziele vor, die von der Anwendung erfüllt werden sollen. Wenn beispielsweise ein erfahrener Benutzer maximal 20 Sekunden benötigen soll um zu einem gewissen Punkt in der Anwendung zu navigieren, so kann dies in der späteren Evaluation genau überprüft werden. Speziell für die Weiterentwicklung von Anwendungen ist es sinnvoll, zusätzlich zu absolut quantifizierten Anforderungen auch relativ quantifizierte Anforderungen zu definieren. Durch diese, auf Erfahrungswerten basierenden Anforderungen wird definiert, wie gut der Benutzer die neu entwickelte Anwendung im Vergleich zur vorhergehenden Anwendung, oder Konkurrenzprodukten benutzen können soll. Weiter gilt zu unterscheiden, ob eine Anforderung die Performanz oder die Zufriedenheit von Benutzern beschreibt. Performanz-Anforderungen geben an, wie effizient die Anwendung von Benutzern verwendet werden soll. Sie quantifizieren hierzu beispielsweise den Navigationsaufwand oder die getätigten Fehler bei der Ausführung einer Aufgabe. Ziele die

die Zufriedenheit von Benutzern angeben, sollen bestimmte Eigenschaften der Anwendung mittels der subjektiven Eindrücke der Benutzer bewerten. Hierbei kann beispielsweise festgelegt werden, dass Benutzer auf einer Skala mit den Extrema Gut und Schlecht, ihre Zufriedenheit mit einem Teil der Anwendung im Mittel mit vier von fünf möglichen Punkten angeben.

Bevor die Anforderungen an die Benutzbarkeit der Anwendung spezifiziert werden, ist es sinnvoll den Umfang und die Art der Anforderungen festzulegen. Im Gegensatz zu neu entwickelten Anwendungen, ist es bei der Weiterentwicklung von Anwendungen wichtiger die geänderten Merkmale auf ihre Nutzbarkeit hin zu überprüfen, als die Verwendung der kompletten Anwendung. Anders ist es bei der Neuentwicklung ohne vorherigen Stand umso wichtiger, die Nutzbarkeit der kompletten Anwendung bewerten zu können (Mayhew, 1999).

Um die Eingewöhnungszeit von Nutzern zu reduzieren, wenn sie anfangen mit der neuen oder weiterentwickelten Anwendung zu arbeiten, ist es von Nutzen, die Benutzeroberfläche auf Grundlage von bisherigem Nutzungsverhalten zu entwerfen. Hierzu sind vor allem Anwendungsfälle, wie sie am Anfang der Entwicklung existieren, wichtig um einen Einblick zu erhalten, wie die Benutzer ihre Aufgaben derzeit erledigen. Diese Anwendungsfälle können im Folgenden dazu verwendet werden, um Vorgaben für den Entwurf der Benutzeroberfläche zu erstellen. Hierbei ist allerdings zu beachten, dass diese Vorgaben keine Vorgaben für die visuelle Gestaltung sind, sondern vorgeben wie die jeweilige Funktionalität dem Benutzer bereitgestellt werden soll. Durch Anpassungen der visuellen Gestaltung an die Bedürfnisse der Benutzer ist es möglich deren Effektivität und Zufriedenheit mit der Anwendung zu erhöhen.

2.2.2 Konzeptentwurf

Der Entwurf der Benutzeroberfläche ist nach (Mayhew, 1999) in drei Abschnitte unterteilt, die jeweils iterativ durchlaufen werden. Im ersten Schritt, dem Konzeptentwurf wird festgelegt wie der Benutzer die gesamte Anwendung wahrnehmen soll, das heißt wie die Anwendung die Realität modellieren soll. Im darauffolgenden Schritt werden sogenannte Screen Design Standards entworfen, die angeben wie die einzelnen Ansichten der Anwendung auszusehen haben. Die Details der Benutzeroberflächen werden im dritten und letzten Schritt beschrieben.

Der Konzeptentwurf verwendet die aus der Analyse verfügbaren Spezifikationen und gibt ein Modell vor, mit dessen Hilfe der Benutzer die Anwendung wahrnehmen soll. Da Nutzer bei der Benutzung von Anwendungen nach einem einheitlichen Modell suchen, welches das Verhalten der Anwendung repräsentiert (Mayhew, 1991), ist es von essentieller Bedeutung, das Konzept so zu entwerfen, dass es dieses mentale Modell unterstützt. Dadurch ist es für die Benutzer einfacher, sich in die Anwendung hineinzuversetzen und die Funktionalitäten zu erlernen. Eine effektivere und effizientere Nutzung der Anwendung ist die Folge.

Überlegungen, welche für den Entwurf des Anwendungskonzepts relevant sind, betreffen beispielsweise die Darstellung von Objekten und Funktionalitäten, sowie einen groben Entwurf der Hauptansichten und der Navigation. Diese Überlegungen gehen von einigen wenigen, aber repräsentativen Funktionalitäten aus, modellieren diese und haben Entwurfsgrundlagen auf hohen Abstraktionsebenen zur Folge. Diese Entwurfsgrundlagen sind meist Metaphern, mit deren Hilfe die Benutzer die modellierte Funktionalität besser in die Realität abbilden und sich in die Anwendung hineinversetzen können.

Es ist wichtig, schon beim Entwurf des konzeptuellen Modells die zu Anfang erstellten Anwendungsfälle zu berücksichtigen, mit in die Überlegungen einfließen zu lassen und auch durchzuspielen, da es dadurch einfacher wird, die getroffenen Entscheidungen schon frühzeitig zu validieren.

Um eine Evaluation des konzeptuellen Entwurfs zu ermöglichen, ist es notwendig dem Benutzer eine Repräsentation der Anwendung zur Verfügung zu stellen um – zumindest ansatzweise – damit zu interagieren (Mayhew, 1999). Es muss abgeschätzt werden, ob für eine Darstellung der Entwurfsentscheidungen einfache Zeichnungen (Mock-Ups) ausreichen, oder ob es sinnvoll ist Prototypen der Anwendung mit eingeschränkter Funktionalität zu entwickeln. Dies kann vor allem bei Weiterentwicklungen von Anwendungen, wenn schon eine gewisse funktionale Basis vorhanden ist, interessant sein und hat den Vorteil, dass die begleitenden Personen des Tests nicht so häufig in den Testablauf eingreifen müssen.

2.2.3 Prototyping

Usability Engineering bietet sich besonders bei iterativen Entwicklungsprozessen an, da die bei der Entwicklung entstehenden Zwischenstände als Prototypen

bei der Evaluation von Oberflächenentwürfen verwendet werden können. Diese Prototypen implementieren zu jeder Iteration unterschiedliche Funktionalitäten oder verändern die vorhandene Funktionalität (Rosson, et al., 2001). Die gleiche Verfeinerung kann auch simultan für die Oberflächen erfolgen, wodurch die Benutzbarkeit während des gesamten Entwicklungszeitraums kontinuierlich verbessert werden kann. Je nach Komplexität der Anwendung muss entschieden werden, welche Art von Prototyp für die Entwicklung in Frage kommt. Die Prototypen können, wenn sie wenig Funktionalität benötigen, mit wenig Aufwand erstellt werden, oder als funktionierende Teilsysteme kostenintensiver in der Herstellung sein.

Es muss überlegt werden, wie detailliert und mit welchem Funktionsumfang die Prototypen den Benutzern präsentiert werden sollen. Je mehr Details ein Prototyp darstellt, desto höher wird die Aussagekraft der mit ihm durchgeführten Usability-Tests. Zu detaillierte Darstellungen können allerdings bewirken, dass die visualisierten Komponenten schon frühzeitig als etabliert angesehen werden. Dies hat die Auswirkung, dass sich sowohl Benutzer, als auch Entwickler an die gewählte Darstellung gewöhnen und gebunden fühlen. Danach ist es nur schwer möglich, große Änderungen durchzusetzen, da sich die betroffenen Personen nicht mehr von dem Gewöhnnten lösen wollen (Rosson, et al., 2001).

Detailärmere (unscharfe) Prototypen haben jedoch bei der Präsentation vor den Benutzern das Problem, dass ihre Wahrnehmung stark von der Präsentationsgabe der präsentierenden Personen beeinflusst wird. Da sie eine große Menge an Erklärungen zur abgebildeten Funktionalität benötigen, hängt auch die Qualität der Rückmeldungen stark von der Präsentation ab. Können sich Benutzer nichts oder nur wenig unter den Erklärungen vorstellen, fällt es ihnen schwer, Überlegungen dazu zu tätigen und zu formulieren. Allerdings bieten unscharfe Prototypen auch wichtige Vorteile: Zum einen sind die Benutzer und Entwickler bei einfachen Zeichnungen eher auf demselben Stand was die Präsentationstechnik betrifft. Dadurch spielt das Fachwissen, welches in dem besprochenen Teil der Anwendung modelliert wird eine größere Rolle und die Entwickler müssen ihre Vorstellungen nicht von der verwendeten Technik beeinflussen lassen. Dies ist speziell hilfreich, wenn Benutzer auf den verwendeten Zeichnungen einfach mit Hilfe von Stiften ihre Ideen einzeichnen können, und somit das Dokument in gemeinsamer Arbeit verändert wird. Zum anderen können mit unscharfen

Prototypen schon frühzeitig Evaluationen mit Benutzern durchgeführt werden um sicherzustellen, dass die Vorstellungen der Entwickler auch mit den Vorstellungen der Benutzer übereinstimmen (Rosson, et al., 2001).

Der Funktionsumfang kann durch horizontales Prototyping entweder breit gefächert dargestellt werden, wobei die Funktionalität nur oberflächlich umgesetzt wird. Oder er kann im vertikalen Prototyping auf wenige Funktionalitäten spezifiziert sein, allerdings dadurch andere Funktionalitäten unberücksichtigt lassen. Beim horizontalen Prototyping können dabei Details der Anwendung verloren gehen, beim vertikalen Prototyping hingegen kann der Gesamtüberblick der Benutzer verloren gehen. Wenn mit einem iterativen Prozess entwickelt wird, können die horizontal aufgesetzten Prototypen weiterentwickelt werden und somit auch mehr Details darstellen.

Prototypen sind auch schon vor der Evaluationsphase der Iterationen nützlich, wenn es beispielsweise zu Unstimmigkeiten bei der Entscheidungsfindung kommt. In diesen Fällen ist es meist hilfreich, wenn die zur Entscheidung stehenden Möglichkeiten in einem Prototyp dargestellt werden (Rosson, et al., 2001). Auch können Prototypen verwendet werden um spezielle Fragen zu Auswirkungen von Entscheidungen auf die Benutzbarkeit der Anwendung zu beantworten. Beim Durchspielen der Anwendungsfälle mit den Entwicklern können schlechte Ideen somit schon frühzeitig, noch bevor sie mit den Benutzern besprochen werden, erkannt und verworfen werden.

2.2.4 Usability Evaluation

Das wichtigste Instrument des Usability Engineering ist die Usability Evaluation. Während dieser Phase werden die zum durchgeführten Zeitpunkt verfügbaren Ergebnisse von Testnutzern verwendet und Reaktionen und Meinungen gesammelt und ausgewertet. Sie ist eine Analyse oder Empirische Untersuchung, welche die Benutzbarkeit von Prototypen eines Systems einstufen und Feedback für den iterativen Entwicklungsprozess geben soll (Gould, et al., 1985). Durch das gesammelte Feedback können die Entwickler Fehler in den Entwürfen erkannt und behoben werden.

(Scriven, 1967) unterscheidet zwischen verschiedenen Arten der Evaluation:

Die gestaltende Evaluation findet während des gesamten Entwicklungsprozesses statt. Ihre Ergebnisse fließen in den Entwicklungsprozess ein, indem sie

bei weiteren Entwurfsentscheidungen berücksichtigt werden. Dies ist speziell bei iterativen Entwicklungsmodellen zutreffend.

Abschließende Evaluationen finden am Ende eines Projekts statt um das entwickelte System zu bewerten und Aussagen darüber treffen zu können, inwieweit die getroffenen Entscheidungen korrekt waren und die Anforderungen erfüllen.

Die empirische Evaluation untersucht die Benutzbarkeit des Systems durch eine Beobachtung reeller Nutzer bei der Benutzung des Systems. Sie liefert dabei Daten darüber, wie gut oder schlecht sich das System von den testenden Benutzern verwenden lässt und auf welche Probleme diese stoßen. Aussagen über den Ursprung der Probleme können mittels empirischer Evaluationen allerdings nur schwer getroffen werden, da sie über keine Informationen über die Auswirkungen der unterschiedlichen Systemeigenschaften verfügen.

In der analytischen Evaluation werden Systemeigenschaften objektiv untersucht, um festzustellen wie diese die Benutzbarkeit des Systems beeinflussen.

Um aussagekräftige Evaluationen zu ermöglichen, ist es gegebenenfalls sinnvoll, auf eine Mischung von analytischen und empirischen Evaluationen zu setzen. Hierbei sollen die analytischen Evaluationen Grundlagen schaffen, mit deren Hilfe entschieden werden kann, welche Stellen des Systems mittels empirischer Evaluation untersucht werden müssen (Rosson, et al., 2001).

Als Grundlage von Usability Evaluationen muss die Nutzbarkeit des Systems oder der Anwendung zu Beginn spezifiziert werden. Hierfür werden die allgemeinen Anforderungen an die Nutzbarkeit untersucht und daraus Aufgaben für die Testnutzer in den Evaluationen hergeleitet. Diese Aufgaben werden immer weiter verfeinert, bis sie die einzelnen Schritte beschreiben, die der Nutzer während der Ausführung der Aufgabe tätigt. Für diese Schritte können nachfolgend messbare Ziele definiert werden, welche Aufschluss darüber geben, wie gut die Anwendung verwendet werden kann. Die Anwendung kann abschließend darauf untersucht werden, bis zu welchem Grad die spezifizierten Punkte erfüllt werden.

Als Beispiele analytischer Evaluationsmethoden nennt (Rosson, et al., 2001) neben Usability Inspektionen, bei denen Usability-Experten die entwickelte Anwendung auf Schwachstellen hin untersuchen, die modellbasierte Analyse. In der modellbasierten Analyse erstellen die Entwickler als Grundlage ein Modell, welches die Interaktion eines Nutzers simuliert. Hierbei soll abgeschätzt werden, wie lange ein Nutzer für das Ausführen einer Aufgabe benötigt. Um die aufgewendete

Zeit gut approximieren zu können, müssen die Aufgaben so weit in Unteraufgaben unterteilt werden, bis einzelne Interaktionen mit der Anwendung, wie beispielsweise das Betätigen der Maustaste oder die Bewegung des Mauszeigers, berücksichtigt werden können. Diesen Interaktionen können im Anschluss geschätzte Zeitaufwände zugewiesen werden, wodurch sich eine Gesamtzeit für die übergeordneten Aufgaben ergibt.

Für empirische Evaluationen werden in (Rosson, et al., 2001) drei wesentliche Arten an Evaluationsmethoden unterschieden:

Usability Tests

Bei der Durchführung von Usability Tests in Usability-Laboren werden die Evaluationen, unabhängig von der späteren Arbeitsumgebung, in abgetrennten Räumen durchgeführt. Diese Tests werden von Testnutzern unter Beobachtung und gegebenenfalls Begleitung von Usability-Experten durchgeführt. Durch die Trennung von Arbeitsumgebung und Testumgebung können störende Einflüsse weitestgehend vermieden werden und die Nutzer können sich komplett auf die Untersuchung der Anwendung konzentrieren. Die testenden Personen sind bei dieser Art von Evaluation meist keine echten Endbenutzer der entwickelten Anwendung. Um die Ergebnisse dennoch aussagekräftig zu halten, müssen die ausgesuchten Testnutzer die Zielgruppe der Endbenutzer ausreichend genau repräsentieren. Hierbei sind vor allem Faktoren wie Alter, Ausbildung und Fähigkeiten der Nutzer ausschlaggebend.

Ein Risiko bei dieser Vorgehensweise ist, dass die Aufgaben nicht mit den tatsächlichen Aufgaben der Endnutzer übereinstimmen und dies nicht erkannt wird. Bei einer Durchführung unter Einbeziehung der späteren Nutzer der Anwendung ist es wahrscheinlicher, dass Fehler in den entworfenen Aufgabenstellungen erkannt werden. Da die Testnutzer jedoch keine echte Erfahrung mit der realen Arbeitsweise haben, fallen diese weniger häufig auf.

Ein weiteres Problem ist, dass durch die nur simulierte Arbeitsumgebung wichtige Hilfsmittel und Einflussfaktoren fehlen. Hierzu zählen beispielsweise der Umgang mit und die Hilfe von Kollegen, die Gestaltung der Arbeitsumgebung und der Leistungsdruck. Zwar können sich die Test-Beobachter dadurch auf die eigentlichen Auswirkungen der Entwurfsentscheidungen auf die Benutzbarkeit der Anwendung konzentrieren, andererseits fehlt der Bezug zur Gesamtsituation.

Außer den Bearbeitungszeiten der festgelegten Aufgaben und der Vorgehensweise der Testnutzer sind neben Fehlerraten bei der Durchführung auch Daten interessant und zu erfassen, die nicht direkt messbar sind. Diese beziehen sich auf Gedanken und Empfinden der Testnutzer. Hierzu zählen zum einen die geplante Vorgehensweise bei der Lösung von Aufgaben, sowie Meinungen über das Aussehen der Anwendung. Speziell Gedanken die mit der Interaktion zu tun haben sind für die Auswertung interessant und können mittels lautem Denken der Testnutzer erfasst werden. Bei der lauten Äußerung des eigenen geplanten Vorgehens kann dieses Vorgehen jedoch beeinflusst werden, da die Nutzer dazu verleitet werden, darüber nachzudenken ob die Planungen sinnvoll sind. Die subjektiven Eindrücke können auch direkt vor und nach den Tests mittels geeigneten Verfahren, wie dem semantischen Differential (2.2.8) oder dem NASA Task Load Index (2.2.9), erfasst werden.

Feldstudien

Feldstudien untersuchen die Benutzung der entwickelten Anwendung in realen Arbeitsumgebungen und mittels Beobachtung der Benutzer bei ihrer regulären Arbeit. Durch die fehlende Beeinflussung der Nutzer durch Test-Beobachter ist es möglich, die entworfenen Testaufgaben für andere Evaluationsmethoden zu validieren und zu verfeinern. Da Feldstudien typischerweise mit einer großen Anzahl an Benutzern durchgeführt werden, ist die Auswertung der Ergebnisse, wie zum Beispiel die Nachvollziehung des Vorgehens von Benutzern und deren Arbeitsweise, sehr kostenintensiv. Ein Mittel zur Verringerung dieser Kosten ist die Durchführung von Nutzerbefragungen zu einigen, von ihnen als auffällig eingestuften Nutzungsfällen. Diese Fälle müssen von den Nutzern dann jedoch aus ihrem Gedächtnis wiedergegeben werden, was meist mit Ungenauigkeiten der Beschreibungen verbunden ist. Durch die Durchführung in der produktiven Arbeit, ist eine wichtige Voraussetzung von Feldstudien, dass die Anwendung schon über ein gewisses Maß an Funktionalität verfügt, oder im Idealfall schon funktionell fertiggestellt ist.

Storefront Tests

Die von (Rosson, et al., 2001) erwähnten Storefront Tests sind ein Kompromiss aus Usability Tests und Feldstudien und sollen die positiven Eigenschaften der beiden Vorgehen vereinen. Die Durchführung in der finalen Arbeitsumgebung und die damit verbundene Validität der Evaluation sind genauso wichtiger Bestandteil wie die Begleitung der Benutzer bei ihrer Testdurchführung. Zur Durchführung von Storefront Tests wird ein System mit der zu testenden Anwendung an einer hinreichend zugänglichen Stelle in der Arbeitsumgebung platziert, an der die Benutzer die Anwendung testen und sofortige Rückmeldung dazu geben können. Durch die freie Zugänglichkeit und unkompliziertere Teilnahme an Tests ist es möglich, schon frühzeitig Entwurfsentscheidungen zu bewerten und verwendete Prototypen iterativ zu testen und zu verbessern.

Kontrollierte Experimente

Um bei Unentschlossenheit bezüglich Entwurfsentscheidungen, verschiedenen Möglichkeiten zu testen, bieten sich kontrollierte Experimente an, die unter gleichen Voraussetzungen unterschiedliche Ansätze bewerten. Für diese Art der Evaluation muss zuerst festgelegt werden, welche unabhängigen Faktoren getestet werden und welche von ihnen abhängig und für die Beobachtung der Auswirkungen relevant sind. Jede Kombination an unabhängigen Faktoren wird im Anschluss in separaten Testläufen ausgewertet.

Die Testläufe werden jeweils mit einer Gruppe an Testnutzern durchgeführt, wobei deren Zusammensetzungen repräsentativ zu der Zusammensetzung der Zielbenutzergruppe sein sollten. Es gilt zu überlegen, wie die unterschiedlichen Ansätze unter den Benutzergruppen aufgeteilt werden sollen. Bei Within-Subject Tests werden alle zu testenden Ansätze von der gleichen Benutzergruppe evaluiert, wodurch zusätzliche Benutzergruppen eingespart werden können. Gleichzeitig können die Einflüsse der Fähigkeiten der Benutzer vernachlässigt werden, wenn die verschiedenen Ansätze verglichen werden, da diese nicht von Benutzergruppe zu Benutzergruppe variieren. Allerdings bringt diese Methode den Nachteil mit sich, dass Testnutzer über die unterschiedlichen Testläufe hinweg Erfahrungen mit dem System sammeln und somit das Ergebnis verfälscht werden kann. Im Gegensatz zu Within-Subject können Tests auch Between-Subject

durchgeführt werden, wobei die Ansätze jeweils von unterschiedlichen Nutzergruppen evaluiert werden. Es ist meist vorteilhaft einen Kompromiss aus den beiden Methoden zu wählen um die jeweiligen Vorteile zu erhalten und die Nachteile auszugleichen. Hierfür werden einige der unabhängigen Variablen innerhalb der Benutzergruppen geändert, andere zwischen den Benutzergruppen.

Für die Aussagekraft der kontrollierten Experimente ist eine repräsentative Zusammensetzung der ausgesuchten Benutzer von hoher Bedeutung. Gleichzeitig ist es jedoch wichtig, diese Benutzer gleichmäßig auf die Gruppen zu verteilen. Während es bei einer großen Anzahl an ausgewählten Testnutzern durchaus plausibel ist, die Benutzer per Zufall auf die Benutzergruppen zu verteilen, kann dies bei kleineren Anzahlen zu einer ungleichmäßigen Verteilung kommen. In solchen Fällen ist es sinnvoller die Verteilung anhand einiger Variablen, welche eine Auswirkung auf die Ergebnisse haben könnten, gleichmäßig zu verteilen ohne auf die anderen Variablen zu achten. Wenn solche Variablen, wie beispielsweise das Benutzeralter gleichmäßig über die Benutzergruppen verteilt sind, können sie als Ursache von Ungenauigkeiten ausgeschlossen werden.

2.2.5 Evaluation des konzeptionellen Entwurfs

Die Evaluation des konzeptionellen Entwurfs dient in erster Linie der frühzeitigen Erkennung grober Fehler. Wenn diese schon früh erkannt werden, können sie im nächsten Iterationsschritt des konzeptionellen Entwurfs berücksichtigt werden und es entfallen kostspielige Verbesserungen zu späten Zeitpunkten in der Entwicklung. Gleichzeitig können durch die Evaluation verschiedene Entwürfe verglichen und die passendsten Ansätze für die weitere Entwicklung ausgewählt werden. Die Einbindung der Endnutzer in dieser frühen Phase der Anwendungsentwicklung ist ein wichtiger Bestand der Evaluation, da sich die Entwickler mehr Zeit haben um sich an die Mitarbeiter der Nutzer zu gewöhnen (Mayhew, 1999). Dies gestaltet vor allem die iterative Entwicklung viel agiler und effektiver.

Je nachdem welche Entscheidungen bei der Erstellung der Anforderungen an die Benutzbarkeit getroffen wurden, muss der Fokus auch bei der Evaluation auf Ease-Of-Learning oder Ease-Of-Use gelegt werden. Dadurch ergeben sich unterschiedliche Vorgehensweisen bei der Durchführung der Evaluation. Soll die Anwendung auf Ease-Of-Learning ausgelegt sein, so ist es wichtig, dass die Testnutzer vor Beginn des Tests keine Kenntnisse über die Anwendung besitzen.

Aus diesem Grund ist es wichtig, die Einweisungen knapp zu halten und nur eine kurze Anleitung über die generelle Benutzung der Anwendung zur Verfügung zu stellen. Wenn der Faktor Ease-Of-Use für die Entwicklung wichtiger ist, dann sollten die Testnutzer nach Möglichkeit wie erfahrene Nutzer agieren. Hierfür sind eine ausführlichere Einweisung und eine angemessene Einarbeitungszeit in die Anwendung nötig. Um aussagekräftige Ergebnisse zu erhalten, ist es wichtig, Tests mit einer ausreichenden Anzahl an repräsentativen Nutzern durchzuführen. Je nach Komplexität der getesteten Eigenschaften genügen laut (Mayhew, 1999) drei bis zehn Nutzer.

Für die Durchführung der Tests werden verschiedene Dokumente benötigt. Neben den Dokumenten, auf denen das Vorgehen und die Kommentare der Nutzer vermerkt werden, sind dies beispielsweise eine Einweisung in die Funktionsweise der Anwendung, sowie Fragebögen zu den Grundkenntnissen und Eigenschaften der Testnutzer. Nach der Ausführung können die Tests ausgewertet und deren Ergebnisse in der nächsten Iterationsphase berücksichtigt werden.

2.2.6 Screen Design Standards

Der zweite Schritt des Benutzeroberflächenentwurfs nach (Mayhew, 1999) ist der Entwurf von Screen Design Standards, welche vorgeben wie die unterschiedlichen Bereiche und Seiten der Anwendung aufgebaut sein sollen. Dies ist wichtig um eine gewisse Konsistenz des Aussehens der Anwendung zu erreichen, da diese von Benutzern erwartet wird (Mayhew, 1991) und bei Nichtvorhandensein zu Fehlern führen kann. Durch die Vorgabe von zu verwendenden Elementen können diese zentral entwickelt werden und an den jeweiligen Stellen ohne erneuten Entwicklungsaufwand eingebunden werden.

Die in diesem Schritt entstehenden Styleguides geben den Entwicklern Vorgaben zu den zu verwendenden Farben, Positionen von Elementen (bspw. Bestätigungsbuttons unten rechts), Schriftarten, Layout usw. Bei einer an diese Styleguides angelehnten Entwicklung wird die Benutzung der Anwendung komfortabler gestaltet.

Die Evaluation der Screen Design Standards unterscheidet sich zu der des konzeptionellen Entwurfs im Wesentlichen dadurch, dass in diesem Schritt vermehrt Prototypen verwendet werden. Diese Prototypen implementieren jedoch

nur ein geringes Maß an Funktionalität um die getroffenen Entwurfsentscheidungen ausreichend testen zu können. Obwohl anstelle von Prototypen auch Techniken wie Wireframes verwendet werden können, ist es zu diesem Zeitpunkt sinnvoll auf Prototypen zu setzen, da diese für eine realitätsnähere Darstellung der Anwendung und der umgesetzten Eigenschaften sorgen, sowie eine minimale Bewertung der dargestellten Funktionalität ermöglichen. Ein großes Problem bei der Verwendung von Prototypen ist jedoch, dass diese meist nur mit geringem Aufwand entwickelt werden und somit nicht über eine so hohe Qualität verfügen wie es von der resultierenden Anwendung erwünscht ist. Da die für die Prototypen entwickelten Elemente aber schon zu einem gewissen Grad verwendbar sind, werden diese meist in der eigentlichen Anwendung selbst übernommen. Dadurch ergibt sich auch ein weiteres Problem, da die Entwickler bei bereits bestehenden Implementierungen Hemmungen haben, diese grundlegend zu ändern. Daher kann es sinnvoll sein, auf Prototypen zu setzen, bei denen von vorne herein klar ist, dass sie nach den Evaluationen verworfen und nicht in der weiteren Entwicklung verwendet werden.

Wie schon im vorherigen Schritt, geht es bei der Evaluation um die Bewertung der für die Screen Design Standards getroffenen Entscheidungen. Da die Evaluation zu einem Zeitpunkt stattfindet, zu dem der für die Entwicklung aufgebrachte Aufwand noch gering ist, können Änderungen die sich ergeben noch mit geringeren Kosten durchgesetzt werden. Die Durchführung der Tests ist im Allgemeinen gleich wie in 2.2.5 beschrieben, jedoch können zu diesem Zeitpunkt die für die Ausführung der gestellten Aufgaben benötigten Zeiten schon grob erfasst werden. Aus diesem Grund müssen die Testaufgaben, die aus den Anwendungsfällen hergeleitet werden, genauer spezifiziert sein als in der vorherigen Evaluation.

2.2.7 Detaillierter Entwurf der Benutzeroberfläche

Auf Grundlage des konzeptionellen Entwurfs und der Screen Design Standards werden im dritten und letzten Schritt nach (Mayhew, 1999) die Details der Benutzeroberfläche entworfen. Hierfür werden beispielsweise die verwendeten Kontrollelemente im Detail entworfen, Inhalte für Dialoge und andere Interaktionselemente festgelegt und die detaillierte Darstellung der Navigation umgesetzt.

Die Ergebnisse aus den vorherigen Schritten werden nun in der finalen Anwendung umgesetzt und mit den neuen Entscheidungen verfeinert.

Um die Details des Benutzeroberflächendesigns zu bewerten und anschließend zu verbessern wird auch in diesem Schritt in jeder Iteration eine Evaluation der Ergebnisse durchgeführt. Da zu diesem Zeitpunkt nur noch kleinere Änderungen am Design evaluiert werden und die größeren Aspekte schon zufriedenstellend verbessert wurden, hält sich der Aufwand zur Anpassung der Benutzeroberfläche in Grenzen. Durch die Umsetzung der Entwürfe ist in dieser Evaluationsphase bereits eine fertige Anwendung verfügbar die getestet werden kann, wodurch keine Entwicklung von Prototypen oder ähnlichen Hilfsmitteln mehr nötig ist.

Um die fertige Anwendung auch angemessen zu evaluieren, müssen die Aufgaben für die Testnutzer detailliert beschrieben werden. Diese Aufgaben sollten, da bereits Teile der Anwendung evaluiert wurden, andere Anwendungsfälle beschreiben wie die Aufgaben die zu Anfangs verwendet wurden. Bei der Auswertung der Tests sollte der Schwerpunkt auf die Ausführungszeit der Aufgaben gelegt werden, um festzustellen, ob die Anwendung die Arbeit der Nutzer genügend erleichtert, unterstützt und effektiver gestaltet. Die Entscheidungen zu Fokussierung auf Ease-Of-Learning und Ease-Of-Use, sowie die Durchführung der Tests geschehen analog zu 2.2.5 und 2.2.6.

2.2.8 Semantisches Differential

Mit dem semantischen Differential (Osgood, 1952) können Eindrücke des Nutzers zum verwendeten System erfasst werden. Hierzu wird dem Testnutzer eine Liste von gegensätzlichen Begriffen („Warm“ – „Kalt“, „Kompliziert“ – „Einfach“) vorgelegt, auf der er das System jeweils einem der Begriffe, mehr oder weniger stark ausgeprägt, zuordnen muss. Je nach Auswahl der Begriffe können unterschiedliche Aspekte des Systems ermittelt werden. Gegensatzpaare wie „Überladen“ – „Aufgeräumt“ bewerten beispielsweise eher die Benutzbarkeit eines Systems, Gegensatzpaare wie „Hässlich“ – „Schön“ bewerten hingegen ästhetische Aspekte. Um die Bewertungen nicht vom Gesamteindruck des Systems beeinflussen zu lassen werden die Positiva der Gegensatzpaare zufallsbasiert auf die rechte oder linke Seite der Bewertungsskala gelegt.

Für eine Bewertung des entwickelten Systems wird in dieser Arbeit der in Tabelle 1 dargestellte Auszug von (UEQ-Online) verwendet.

Für eine Aussage zur Benutzbarkeit werden die Gegensatzpaare zu Übersichtlichkeit, Geschwindigkeit, Güte, Stabilität, Aufgeräumtheit, Voraussagbarkeit, Einfachheit und Erlernbarkeit verwendet. Eine Aussage zum ästhetischen Eindruck wird anhand der Gegensatzpaaren Güte, Innovation, Attraktivität, Schönheit, Freude, Aufgeräumtheit und Behaglichkeit getroffen.

Verwirrend	Übersichtlich
Langsam	Schnell
Schlecht	Gut
Konservativ	Innovativ
Unattraktiv	Attraktiv
Instabil	Stabil
Hässlich	Schön
Unerfreulich	Erfreulich
Überladen	Aufgeräumt
Unberechenbar	Voraussagbar
Kompliziert	Einfach
Unangenehm	Angenehm
Schwer erlernbar	Leicht erlernbar

Tabelle 1 Für semantisches Differential verwendete Gegensatzpaare

2.2.9 NASA Task Load Index

Der NASA Task Load Index (NASA-TLX) (Hart, et al., 1988) wird verwendet um die Belastung der Benutzer bei der Ausführung von Aufgaben einzustufen und zu bewerten. Hierfür erhält der Benutzer die Möglichkeit, die gestellten Aufgaben nach deren Ausführung subjektiv zu bewerten. Dadurch kann im Hinblick auf die Benutzbarkeit festgestellt werden, inwiefern sich eine Entwurfsentscheidung auf die Durchführbarkeit für den Benutzer auswirkt.

Die verschiedenen Bewertungskriterien (Tabelle 2) können auf einer normalisierten Skala (0 – 100) dargestellt und verglichen werden. Hierbei ist hervorzuheben, dass ein niedriger Wert eher positiv zu werten ist, als ein hoher Wert. Im Anschluss an die Bewertung der einzelnen Kriterien wird dem Nutzer die Möglichkeit gegeben, die Kriterien zu gewichten. Hierzu kann er zwischen gegebenen

Paarungen an Kriterien diejenigen auswählen, die für ihn die größeren Auswirkungen bei der Durchführung der Aufgaben hatten. Durch diese Einstufungen kann eine Gewichtung der jeweiligen Kriterien ermittelt und auf die Bewertungen angewandt werden. Daraufhin wird ein Mittelwert errechnet, welcher die allgemeine Belastung des Nutzers angibt. Nach (Hart, 2006) stellte sich bei der Studie von verwendeten NASA-TLX heraus, dass eine Verwendung der Gewichtung nicht zwangsläufig eine verbesserte Aussagekraft bedeutet. Die Resultate der Befragungen ohne anschließende Gewichtung der Kriterien waren sowohl schlechter, besser als auch gleich gut wie die Resultate der Befragungen nach dem standardmäßigen Vorgehen.

<i>Kriterium</i>	<i>Skala</i>	<i>Bedeutung</i>
<i>Mentaler Anspruch</i>	Niedrig – Hoch	Wie stark musste sich der Nutzer bei der Durchführung der Aufgabe mental anstrengen? Hierzu zählen Dinge wie Nachdenken, Entscheiden, Suchen und ähnliche mentale Tätigkeiten.
<i>Physischer Anspruch</i>	Niedrig – Hoch	Wie viel körperliche Arbeit musste der Benutzer aufbringen um die Aufgabe durchzuführen? Hierbei sind alle körperlichen Tätigkeiten, wie das Betätigen von Tasten und das Bewegen der Maus relevant.
<i>Zeitlicher Anspruch</i>	Niedrig – Hoch	Wie war der zeitliche Druck während der Durchführung der Aufgabe?
<i>Erfolg</i>	Perfekt erfolgreich – Erfolgslos	Wie erfolgreich empfand der Nutzer seine Durchführung der Aufgabe?
<i>Aufwand</i>	Niedrig – Hoch	Wie viel Aufwand musste der Nutzer für die Durchführung der Aufgabe aufbringen, um auf sein Erfolgslevel zu erreichen?
<i>Frustration</i>	Niedrig – Hoch	Wie stark wurde der Nutzer während der Durchführung der Aufgabe von negativen Faktoren beeinflusst?

Tabelle 2 Bewertungskriterien des NASA-TLX

2.3 Natural Language Processing

Das folgende Kapitel beschreibt einige Konzepte die bei der Auswertung von natürlich sprachlichen Eingaben eine Rolle spielen. Für eine Korrektur von Fehleingaben kann die in 2.3.1 beschriebene Damerau-Levenshtein-Distanz verwendet werden. 2.3.2 erläutert die Funktionsweise von N-Grammen, die beispielsweise bei der automatischen Vervollständigung von Texteingaben eingesetzt werden können. Das in 2.3.3 beschriebene Part-Of-Speech-Tagging wird dazu verwendet, Wörtern in einem gegebenen Text ihre Wortarten zuzuweisen und diesen Text somit für weitere folgende Schritte vorzubereiten.

2.3.1 Damerau-Levenshtein-Distanz zur Einstufung der Qualität von Resultaten

Um den Benutzer bei eventuell auftretenden Tipp- und Rechtschreibfehlern zu unterstützen, ist es wichtig den eingegebenen Text mit dem Resultat zu vergleichen und zu bewerten. Hierfür ist die Levenshtein-Distanz gut geeignet. Sie gibt an, mit wie vielen Operationen aus einer gegebenen Zeichenfolge eine weitere gegebene Zeichenfolge gebildet werden kann. Speziell werden hierbei die Operationen „Einfügen“, „Löschen“ und „Ersetzen“ betrachtet, welche bei jedem Einsatz die resultierende Distanz um eins erhöhen.

Nach (Damerau, 1964) ist es weiter möglich, vertauschte Zeichen zu erkennen und diese mit einer angepassten Gewichtung zu berücksichtigen.

Die Distanz muss schlussendlich nur noch mit der Länge der verglichenen Wörter in Bezug gesetzt werden um eine prozentuale Übereinstimmung zu erhalten. Mit einer Länge l_{w_1} des ersten Wortes, des zweiten Wortes l_{w_2} und einer Damerau-Levenshtein-Distanz $d_{w_1w_2}$ kann die Übereinstimmung folgendermaßen ausgedrückt werden:

$$m(w_1, w_2) = \frac{\max(l_{w_1}, l_{w_2}) - d_{w_1w_2}}{\max(l_{w_1}, l_{w_2})}$$

Mit dieser Übereinstimmung ist es möglich, eine Aussage über die Qualität der Ergebnisse und der anfänglichen Eingabe zu machen.

Am Beispiel der beiden Wörter „Fahrzeug“ und „Fhareuzfg“ ist die Berechnung der Damerau-Levenshtein-Distanz in Abbildung 4 dargestellt. Die getätigten Schritte sind rot gekennzeichnet. Die Buchstaben „F“ und „F“ unterscheiden sich

nicht, weshalb keine Aktion notwendig ist. Nach einer Vertauschung der Buchstaben „ha“ stimmen die Teilwörter „Fhar“ und „Fahr“ nach einem Schritt überein. Ein Hinzufügen des Buchstabens „z“ nach „Fahr“ und ein anschließendes Löschen des überflüssigen „z“ würden den gleichen Aufwand benötigen wie ein Verschieben des Buchstabens an die korrekte Stelle. Nach dem Löschen des Buchstabens „f“ stimmen die beiden Zeichenketten überein.

	F	h	a	r	e	u	z	f	g
	1	2	3	4	5	6	7	8	9
F	1	0	1	2	3	4	5	6	7
a	2	1	1	1	2	3	4	5	6
h	3	2	1	1	2	3	4	5	6
r	4	3	2	2	1	2	3	4	5
z	5	4	3	3	2	2	3	3	4
e	6	5	4	4	3	2	3	4	5
u	7	6	5	5	4	3	2	3	4
g	8	7	6	6	5	4	3	3	4

Abbildung 4 Damerau-Levenshtein-Distanz am Beispiel der Wörter "Fahrzeug" und "Fhareuzfg"

Die Damerau-Levenshtein-Distanz der beiden Wörter kann somit mit 4 berechnet werden. Die Übereinstimmung zwischen den beiden Wörtern ist folglich:

$$m(\text{Fahrzeug}, \text{Fhareuzfg}) = \frac{9 - 4}{9} = 0, \bar{5}$$

2.3.2 N-Gramme

N-Gramme sind Wortfolgen mit N Wörtern und ein wichtiges Werkzeug des Natural Language Processing. Sie werden erstellt indem ein vorgegebener Text in Wortfolgen der Länge N aufgeteilt wird. Je nach Häufigkeit der vorkommenden N-Gramme kann später ermittelt werden, welches Wort am wahrscheinlichsten auf einen bestimmten Text folgen wird. Je nach Anforderungen und Gegebenheiten muss entschieden werden, welche Art von N-Grammen verwendet werden soll. N-Gramme mit einer großen Anzahl an Wörtern bieten zwar eine hohe Genauigkeit bei der Vorhersage der nächsten Wörter, müssen aber auf größeren Korpusen aufbauen um eine Vorhersage zu treffen die auch allgemein gültig sein kann (Jurafsky, et al., 2009). Typischerweise werden N-Gramme mit einer

Länge von 2 (Bi-Gramme) oder 3 (Tri-Gramme) verwendet, um je nach Voraussetzungen eine Kompromiss zwischen Genauigkeit und Größe der verwendeten Korpusse zu finden.

N-Gramme sind vor allem dann nützlich, wenn sich die Eingaben durch einen hohen Grad an Rauschen auszeichnen, wie dies unter anderem bei gesprochener Sprache und Übersetzungen der Fall ist. Gleichzeitig sind N-Gramme auch ein nützliches Werkzeug um Fehleingaben der Nutzer zu erkennen und gegebenenfalls zu beseitigen. Gibt der Benutzer ein Wort ein, das durch eine lexikalische Fehlererkennung nicht erkannt werden kann, da es sich um ein korrekt geschriebenes Wort handelt, kann mittels N-Grammen festgestellt werden, dass die somit entstandene Wortfolge nur äußerst selten auftritt. Ein weiteres Einsatzszenario für N-Gramme ist die Eingabevorhersage. Aus vorherigen Eingaben kann ermittelt werden, welche Eingabe am wahrscheinlichsten ist. Bei jedem eingegebenen Wort kann nachfolgend die Vorhersage verbessert und somit dem Nutzer eine passende Eingabe vorgeschlagen werden (Jurafsky, et al., 2009).

Die Wahrscheinlichkeiten, mittels derer die Vorhersagen getroffen werden, basieren wie schon erwähnt, auf sogenannten Korpusse. Ein Korpus ist eine Ansammlung an Texten, die einen möglichst großen Teil einer verwendeten Sprache modellieren soll. Je nach Auswahl der Texte können unterschiedliche Teilgebiete einer Sprache modelliert werden. Werden beispielsweise Artikel aus Fachzeitschriften des Bereichs Informatik verwendet um einen Korpus zu erstellen, so existieren darin N-Gramme die sich zu einem Korpus auf Basis von Tageszeitungen deutlich unterscheiden können. Um Wahrscheinlichkeiten für Wortfolgen zu bestimmen, werden beim Anlernen alle Vorkommen der N-Gramme gezählt und in einer Matrix notiert.

Für eine Wortfolge w_1^n wird die Anzahl der Vorkommen dieser Wortfolge im Folgenden mit $C(w_1^n)$ bezeichnet. Die Wahrscheinlichkeit, dass ein Wort w_n auf die vorherige Wortfolge w_1^{n-1} folgt, ist somit:

$$P(w_n | w_1^{n-1}) = \frac{C(w_1^n)}{C(w_1^{n-1})}$$

Die Verwendung von Bi-Grammen und Tri-Grammen resultieren in einer Abschätzung der Wahrscheinlichkeit des nächsten Wortes:

$$P(w_n | w_1^{n-1}) \approx P(w_n | w_{n-2} w_{n-1}) = \frac{C(w_{n-2} w_{n-1} w_n)}{C(w_{n-2} w_{n-1})}$$

Die Wahrscheinlichkeit für einen kompletten Satz ist, darauf aufbauend, das Produkt aller in diesem Satz unter Berücksichtigung der vorherigen Wortfolge auftretenden Wörter:

$$P(w_1^n) = P(w_1) \cdot P(w_2|w_1^1) \cdot \dots \cdot P(w_n|w_1^{n-1}) = \prod_{i=1}^n P(w_i|w_1^{i-1})$$

Nach den von Bi- und Tri-Grammen angenommenen Vereinfachungen ergibt sich eine geschätzte Satzwahrscheinlichkeit:

$$P(w_1^n) \approx \prod_{k=1}^n P(w_k|w_{k-2}w_{k-1})$$

Um die Genauigkeit von gelernten N-Grammen einzustufen, wird die Wahrscheinlichkeit, eine Test-Wortfolge aus den N-Grammen herzuleiten, bestimmt. Würde diese Testfolge aus dem gelernten Korpus genommen, wäre die dadurch errechnete Wahrscheinlichkeit fälschlicherweise hoch. Deshalb müssen die N-Gramme mit einer Wortfolge getestet werden, die nicht bei deren Erstellung verwendet wurde. Hierzu wird der verwendete Korpus in zwei Teile unterteilt. Mit dem ersten, größeren Teil wird die Wahrscheinlichkeitsmatrix der N-Gramme erstellt und mit dem zweiten Korpus-Teil die erstellten N-Gramme getestet.

Die Genauigkeit wird mittels der Perplexität $PP(W)$ angegeben:

$$PP(W) = P(w_1^N)^{-\frac{1}{N}}$$

Die Perplexität normalisiert somit die Wahrscheinlichkeit, die gegebene Wortfolge mittels der verfügbaren N-Gramme aufzubauen, mit der Länge der Wortfolge. Das Ergebnis ist eine positive Zahl, welche, je niedriger sie ist, für eine höhere Genauigkeit steht. Durch Umformung und Verwendung von Tri-Grammen erhält man:

$$PP(W) \approx \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i|w_{i-2}w_{i-1})}}$$

Durch $P(w_1^n)$ ergibt sich, dass eine Wortfolge, die eine Teilfolge enthält, welche nicht aus den N-Grammen hergeleitet werden kann, eine Wahrscheinlichkeit von 0 hat. Da ein Korpus aber niemals alle möglichen Sätze einer Sprache modellieren kann, muss dies in der Erstellung der Wahrscheinlichkeitsmatrix berück-

sichtigt werden und den nicht vorkommenden N-Grammen eine gewissen Grundwahrscheinlichkeit zugewiesen werden. Um diese Grundwahrscheinlichkeit zu ermitteln gibt es einige Vorgehensweisen (Jurafsky, et al., 2009):

Laplace-Glättung

Die Laplace-Glättung basiert auf der intuitiven Lösung des Problems, die Anzahl der Vorkommen aller N-Gramme um 1 zu erhöhen. Durch dieses Verfahren wird die restliche Matrix gefüllt und es existiert eine Wahrscheinlichkeit für jedes N-Gram. Bei der Verwendung von Tri-Grammen wird die Wahrscheinlichkeit folgendermaßen angegeben:

$$P_{Laplace}(w_n | w_{n-1}) = \frac{C(w_{n-2}w_{n-1}w_n) + 1}{C(w_{n-2}w_{n-1}) + V}$$

Allerdings ergibt sich ein Problem, da eine sehr hohe Masse an Vorkommen hinzugefügt werden. Speziell N-Gramme mit eigentlich hohen Vorkommen büßen einen Großteil ihrer Wahrscheinlichkeit ein, wodurch die Wahrscheinlichkeitsverteilung verfälscht wird.

Good-Turing-Glättung

Die Good-Turing-Glättung geht davon aus, dass die N-Gramme die 0 Vorkommen in der Wahrscheinlichkeitsmatrix haben, mit etwa derselben Wahrscheinlichkeit wie die N-Gramme die einmal vorkommen, auftreten. Mit N_c als Anzahl an Objekten die c -mal vorkommen, lässt sich die korrigierte Wahrscheinlichkeit P^* folgendermaßen darstellen:

$$P_{GT}^*(\text{Objekte mit Häufigkeit 0}) = \frac{N_0}{N}$$

Jedes Objekt mit 0 Vorkommen erhält nun die korrigierte Wahrscheinlichkeit geteilt durch N_0 . Die Häufigkeiten aller anderen Objekte werden daraufhin angepasst um die gesamte Wahrscheinlichkeit nicht zu verfälschen:

$$c^* = (c + 1) \frac{N_{c+1}}{N_c}$$

Interpolation

Durch die Verwendung von N-Grammen bietet es sich auch an, die Eigenschaft auszunützen, dass sie mit abnehmender Länge zwar ungenauer werden jedoch mehr Wortfolgen modellieren können. Wird beispielsweise ein Tri-Gramm für $w_{n-2}w_{n-1}w_n$ nicht gefunden, so ist es dennoch möglich, dass ein Bi-Gramm

für $w_{n-1}w_n$ existiert. Dies lässt sich bis zur absoluten Wahrscheinlichkeit $P(w_n)$ der Einzelnen Wörter durchführen. Die N-Gramme werden im Anschluss gewichtet zusammengesetzt, wodurch sich im Falle von Tri-Grammen, die folgende korrigierte Wahrscheinlichkeit ergibt:

$$\hat{P}(w_n|w_{n-2}w_{n-1}) = \lambda_1 P(w_n|w_{n-2}w_{n-1}) + \lambda_2 P(w_n|w_{n-1}) + \lambda_3 P(w_n)$$

2.3.3 Part of Speech Tagging

Part of Speech Tagging (POS-Tagging) ist ein wesentlicher Bestandteil der Verarbeitung von natürlicher Sprache. POS-Tagger werden dafür verwendet, Wörter in einer Wortfolge die Wortarten zuzuweisen die sie repräsentieren. Diese Zuweisung wird auf Basis eines Tagsets durchgeführt, das alle relevanten Wortarten beinhaltet. Im Deutschen wird meist das Stuttgart-Tübingen-Tagset (STTS) verwendet, welches neben Wortarten für Infinitivverben (VINF, VVIZU, VAINF, VMINF), Satzzeichen (\$., \$, \$,) und Eigennamen (NE) über 46 weitere Wortarten verfügt (Schiller, et al., 1999).

Da Wörter in verschiedenen Kontexten unterschiedliche Wortarten darstellen können, ist die intuitive Lösung, den Wörtern ihre wahrscheinlichste Wortart zuzuweisen, meist nicht ausreichend. Im Satz „Die Musik war laut.“ ist das Wort „laut“ ein die „Musik“ beschreibendes Adjektiv. Im Satz „Laut ersten Forschungsergebnissen sind Bananen meist krumm.“ ist dasselbe Wort hingegen eine Präposition. Um diese Wörtern korrekt zu markieren, ist es in solchen Fällen nötig eine erweiterte Logik zu verwenden, welche von POS-Taggern umgesetzt werden.

Regelbasierte POS-Tagger versuchen, die korrekten Wortarten mittels vorher vom Entwickler festgelegten Regeln zuzuweisen. Hierfür erhalten die zu markierenden Wörter alle für sie möglichen Wortarten, welche in den nächsten Schritten nach und nach eliminiert werden, bis nur noch eine einzige Wortart pro Wort übrig ist. Durch die Vielfältigkeit der natürlichen Sprache ist es für ein genaues Arbeiten der POS-Tagger notwendig, ihnen eine große Menge an Regeln zur Verfügung zu stellen.

Wahrscheinlichkeitsbasierte Tagger weisen den Wörtern aufgrund von vorher berechneten Wahrscheinlichkeiten die am besten zu ihnen passenden Wortarten zu. Hierzu wird die Sequenz an Wortarten gewählt die, je nach Voraussetzungen

des Taggers, die höchste Wahrscheinlichkeit hat. Die verwendeten Wahrscheinlichkeiten basieren, je nach POS-Tagger, auf unterschiedlichen Faktoren. Während der Hidden Markov Model-Tagger von den Wahrscheinlichkeiten von Wortartfolgen und deren Repräsentationen als Wörter beeinflusst wird, werden Maximum Entropy Markov Model-Tagger von zusätzlichen Daten, wie Groß- und Kleinschreibung beeinflusst.

Hidden Markov Model-Tagger

Hidden Markov Model (HMM)-Tagger versuchen die Wahrscheinlichkeit der ausgewählten Wortartsequenz für den vorgegebenen Text zu maximieren. Dabei wird allerdings nicht die Wahrscheinlichkeit verwendet, dass ein Wort w die Wortart t repräsentiert, sondern durch Anwendung der Bayes-Formel die Wahrscheinlichkeiten der Wortartsequenz und der Repräsentation als das vorgegebene Wort:

$$P(t_1^n | w_1^n) = \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}$$

Da der eingegebene Text sich während der Ermittlung der korrekten Wortartfolge nicht ändert, ist die Wahrscheinlichkeit die vorgegebene Wortsequenz zu erhalten konstant. Deshalb ist nur der Term $P(w_1^n | t_1^n)P(t_1^n)$ relevant. Durch Verwendung von Tri-Grammen ergibt sich:

$$P(t_1^n | w_1^n) \approx \prod_{i=1}^n P(w_i | t_i)P(t_i | t_{i-2}t_{i-1})$$

Für die Glättung der Tri-Gramme wird meist eine einfache Interpolation verwendet:

$$P^*(t_i | t_{i-2}t_{i-1}) = \lambda_1 P(t_i | t_{i-2}t_{i-1}) + \lambda_2 P(t_i | t_{i-1}) + \lambda_3 P(t_i)$$

Für die Maximierung dieses Ausdrucks werden von HMM-Taggern Hidden Markov Modelle verwendet. Diese basieren auf Markov-Ketten, welche als gewichtete endliche Automaten angesehen werden können. Markov-Ketten verfügen über eine Menge an Zuständen die reale Beobachtungen darstellen. Die Übergänge von einem Zustand in einen anderen Zustand werden von den n vorherigen Zuständen beeinflusst, wodurch eine Vorhersage von zukünftigen oder eine Herleitung von früheren Zuständen ermöglicht wird.

Zusätzlich zu den beobachteten Zuständen in Markov-Ketten, werden in Hidden Markov Modellen noch nicht-beobachtete Zustände verwendet um beeinflussende Faktoren zu berücksichtigen, die nicht beobachtet werden können, und

somit versteckt sind. In der Verarbeitung von natürlicher Sprache werden die Wörter des gegebenen Textes als Beobachtungen und die zugehörigen Wortarten als versteckte Zustände behandelt. Mit dem HMM werden durch Übergangswahrscheinlichkeiten zwischen den versteckten und Beobachtungswahrscheinlichkeiten der beobachteten Zustände die Wahrscheinlichkeiten errechnet, dass sich das Modell zu einem gegebenen beobachteten Zustand in einem speziellen, versteckten Zustand befindet. HMM können als 5-Tupel (Q, A, O, B, π) angegeben werden (Jurafsky, et al., 2009):

- $Q = q_1, q_2, \dots, q_N$ – Menge aller versteckten Zustände
- $A = a_{11}, a_{12}, \dots, a_{1n}, \dots, a_{nn}$ – Matrix mit den Übergangswahrscheinlichkeiten der versteckten Zuständen q_i nach q_j
- $O = o_1, o_2, \dots, o_T$ – Menge aller Beobachtungen, welche Rückschlüsse auf die versteckten Zustände ermöglichen
- $B = b_i(o_t)$ – Matrix mit Beobachtungswahrscheinlichkeiten, also von Zustand b_i ausgehend die Beobachtung o_t zu machen
- $\pi = \pi_1, \pi_2, \dots, \pi_n$ – Verteilung der Wahrscheinlichkeiten, welche Zustände zu Anfang gültig sind

Wie schon in Markov-Ketten, ist die Wahrscheinlichkeit sich in einem versteckten Zustand zu befinden abhängig von den n vorherigen versteckten Zuständen. Die Wahrscheinlichkeiten der Beobachtungen basieren hingegen ausschließlich auf dem Zustand der die jeweilige Beobachtung verursacht hat. Um die Sequenz an versteckten Zuständen zu ermitteln, die der gegebenen Beobachtungssequenz entspricht, muss die maximale Wahrscheinlichkeit der möglichen Sequenzen berechnet werden. Dies erfolgt nach (Jurafsky, et al., 2009) mittels Viterbi-Algorithmus, der sich die wahrscheinlichsten Übergänge zwischen den versteckten Zuständen merkt und nach erfolgtem Durchlauf der Beobachtungen ausgibt. Hierzu werden vom ersten Zustand $v_1(j)$ ausgehend alle Übergangswahrscheinlichkeiten in den nächsten Zustand berechnet. Der Ausgangszustand wird dann als $bt_1(j)$ notiert und in der nächsten Iteration mit dem dann aktuellen Zustand $v_t(j)$ gleich vorgegangen. Folgende Funktionen sind für die Auswertung der besten Zustandssequenz relevant:

$$v_1(j) = \pi_j b_j(o_1); \quad 1 < j \leq N$$

$$bt_1(j) = 0$$

$$v_t(j) = \max_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 < j \leq N, 1 < t \leq T$$

$$b_t(j) = \operatorname{argmax}_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 < j \leq N, 1 < t \leq T$$

Um die bestmögliche Sequenz der Wortarten zu ermitteln müssen zum Schluss nur noch die $b_t(j)$ ausgewertet werden, welche den verwendeten Pfad bis zum Ende der Beobachtungen beschreiben.

Damit ein HMM-Tagger eine Wortsequenz markieren kann, müssen zuerst die Übergangswahrscheinlichkeitsmatrizen A und B gelernt werden. Dieses Anlernen wird mittels Forward-Backward Algorithmus iterativ in zwei Schritten durchgeführt. Anhand einer ersten Schätzung werden die Wahrscheinlichkeiten so lange verbessert, bis keine merkliche Verbesserung mehr festgestellt werden kann. Die Zelle a_{ij} der Übergangsmatrix A werden ermittelt indem die erwarteten Übergänge von Zustand i nach j in Relation zu allen Übergängen von Zustand i zu beliebigen anderen Zuständen gesetzt wird. Der folgende Ausdruck gibt dabei die Wahrscheinlichkeit an, sich zu einem Zeitpunkt t in Zustand i und zum Zeitpunkt $t + 1$ in Zustand j zu befinden.

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}$$

$\alpha_t(i)$ gibt in diesem Zusammenhang die Wahrscheinlichkeit der vorherigen Zustände, $\beta_{t+1}(j)$ die Wahrscheinlichkeit der nachfolgenden Zustände an. $P(O|\lambda)$ ist die Wahrscheinlichkeit, die Beobachtungsfolge O mit den definierten Matrizen zu beobachten. Durch eine Summierung der $\xi_t(i, j)$ über alle Zeitschritte erhält man die Übergänge von Zustand i nach j und durch eine Summierung über alle Zustände j die Übergänge von Zustand i zu anderen Zuständen.

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-i} \xi_t(i, j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N \xi_t(i, j)}$$

Um die Matrix B der Beobachtungswahrscheinlichkeiten neu zu berechnen werden die erwarteten Beobachtungen des Symbols v_k von Zustand j in Bezug zu der erwarteten Häufigkeit von Zustand j gesetzt. Die Wahrscheinlichkeit, sich zu Zeitpunkt t in Zustand j zu befinden ist dabei:

$$\gamma_t(j) = \frac{\alpha_t(j) \beta_t(j)}{P(O|\lambda)}$$

Die Beobachtungswahrscheinlichkeiten werden folglich berechnet ($\sum_{t=1}^T \sum_{j=1}^N \gamma_t(j)$ steht für die Summe über t wenn Beobachtung O_t v_k war):

$$\hat{b}_j(v_k) = \frac{\sum_{t=1}^T s.t. o_t = v_k \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}$$

Maximum Entropy Markov Model

Im Markov Modell der maximalen Entropie (MEMM) wird die Wahrscheinlichkeit, dass ein Wort eine spezielle Wortart repräsentiert durch gewichtete Eigenschaften der Wortarten modelliert. Diese Eigenschaften beschreiben, welche Faktoren eine Beeinflussung auf das Zutreffen der jeweiligen Wortarten haben. Im Gegensatz zu HMM können MEMM mittels Eigenschaften auch Dinge wie Groß- und Kleinschreibung modellieren, die somit in die Auswertung des Textes miteinfließen. Die Gewichtungen der Eigenschaften geben an, wie stark diese Beeinflussung ausfällt (Jurafsky, et al., 2009).

$$\sum_{i=0}^N w_{ci} f_i(c, x)$$

Ein Problem mit diesem Ansatz ist, dass dadurch keine realistische Wahrscheinlichkeit ausgedrückt wird – die Summe aller gewichteten Eigenschaften wird nicht 1 sein. Somit ist keine gleichmäßige Verteilung der Wahrscheinlichkeiten gegeben. Hierfür wird in dem Konzept des MEMM die logistische Regression angewandt. Die logistische Regression verwendet das Verhältnis der Wahrscheinlichkeit, dass ein Ereignis eintritt zu dessen Gegenwahrscheinlichkeit um daraus die eigentliche Wahrscheinlichkeit zu ermitteln. Die Summe der Eigenschaften der Wortarten und ihre Gewichtungen werden hierbei als dieses Verhältnis betrachtet.

$$\frac{P(c|x)}{1 - P(c|x)} = \sum_{i=0}^N w_{ci} f_i(c, x)$$

Da die beiden Seiten unterschiedliche Wertebereiche haben, müssen diese mittels natürlichem Logarithmus angepasst werden:

$$\ln \left(\frac{P(c|x)}{1 - P(c|x)} \right) = \sum_{i=0}^N w_{ci} f_i(c, x)$$

Durch Umformungen kommt man auf eine Gleichung für die Wahrscheinlichkeit, dass ein Wort x die Wortart c hat:

$$P(c|x) = \frac{e^{\sum_{i=0}^N w_{ci} f_i(c, x)}}{1 - e^{\sum_{i=0}^N w_{ci} f_i(c, x)}}$$

Da die Gegenwahrscheinlichkeit zur Wahrscheinlichkeit, dass ein Wort x die Wortart c hat als Summe aller Wahrscheinlichkeiten, dass dieses Wort eine andere Wortart hat ausgedrückt werden kann, folgt:

$$P(c|x) = \frac{e^{\sum_{i=0}^N w_{ci} f_i(c, x)}}{\sum_{c' \in C} e^{\sum_{i=0}^N w_{c'i} f_i(c', x)}}$$

Die wahrscheinlichste Wortart für das zugrunde liegende Wort wird nachfolgend als Treffer ausgewählt:

$$\hat{c} = \operatorname{argmax}_{c \in C} P(c|x)$$

Gemischte (Transformationsbasierte) Tagger

Neben Regel- und Wahrscheinlichkeitsbasierten POS-Taggern existieren auch noch Mischformen der beiden Arten. Diese POS-Tagger weisen dem eingegebenen Text zuerst mittels Wahrscheinlichkeiten die plausibelsten Wortarten zu und verfeinern diese Wortartsequenzen anschließend mit automatisch generierten Regeln. Diese Regeln werden durch Vorlagen – beispielsweise in der Form „Ändere Wortart a zu b , wenn das folgende Wort Wortart z hat.“ – vorgegeben und vom POS-Tagger für die verschiedenen Wortarten und Wörter generiert. Der Tagger wählt dazu die Transformationsregeln aus allen möglichen Regeln aus, die die höchste Genauigkeit bewirken.

Für die Evaluation von POS-Taggern ist es, wie in bei N-Grammen (2.3.2) wichtig, dass diese nicht auf dem gleichen Korpus ausgeführt werden, der für das Anlernen verwendet wurde. Typischerweise wird hierfür der gesamte Korpus in drei Teile aufgeteilt: 80% werden verwendet um den POS-Tagger anzulernen, 10% um Feinabstimmungen zu treffen und die letzten 10% um den POS-Tagger zu evaluieren. Da durch eine solche Aufteilung ein erheblicher Anteil an Daten für das Anlernen verloren gehen und es auch wünschenswert ist, einen POS-Tagger auf dem gesamten Korpus zu testen, kann ein alternatives Verfahren verwendet werden. In der x -fachen Cross-Validation wird der Korpus zuerst per Zufall in die drei relevanten Teile unterteilt. Im Anschluss wird der Tagger angelernt, angepasst und evaluiert. Danach wird der Korpus neu aufgeteilt um den Tagger erneut zu evaluieren. Dies wird x -mal durchgeführt um eine möglichst hohe Abdeckung des gesamten Korpus zu gewährleisten.

POS-Tagger werden anhand ihrer Fehlerraten bewertet, wobei ein Fehler genau dann auftritt, wenn ein Wort der Wortart x fälschlicherweise mit der Wortart

y markiert wurde. Hierbei muss beachtet werden, dass menschliche Experten in den besten Fällen eine Fehlerrate von 3%-4% erreichen (Marcus, et al., 1993), was als Ziel für einen POS-Tagger gesehen werden kann. Als Maximum für die Fehlerrate eines POS-Tagger kann die Fehlerrate einer einfachen Zuweisung der wahrscheinlichsten Wortarten für jedes Wort angenommen werden, welche zwischen 6% und 7% liegt (Gale, et al., 1992) (Toutanova, et al., 2003).

3 Analyse

Dieses Kapitel beschreibt die Probleme welche von Benutzern bei der Verwendung des föderierten Informationssystems Cleverle wahrgenommen werden. Aus diesen Problemen werden anschließend konkretere Anforderungen an die neu zu entwickelnde Oberfläche formuliert. Die Analyse der Probleme fokussiert sich speziell auf den in 1.1 beschriebenen problematischen Aufbau der Navigation in föderierten Informationssystemen.

3.1 Usability Spezifikation

Für die Erfassung der Benutzeranforderungen wurden hauptsächlich Mitarbeiter des Cleverle-Supports befragt, da sie einen guten Einblick in die größten Probleme der Benutzer bei der Verwendung von Cleverle haben. Für einen genaueren Einblick in die Arbeitsweise und die Probleme im täglichen Arbeitsalltag wurden zusätzlich Benutzer bei ihrer Arbeit mit dem System beobachtet. Weitere Anforderungen an die Benutzbarkeit, speziell auch für die Verwaltung des Systems, wurden bei Gesprächen mit den Betreuern deutlich.

3.1.1 Anforderungen der Nutzer

Aus Sicht der Benutzer ist der erhebliche Navigationsaufwand um verschiedene Funktionen zu finden und auszuführen ein Problem bei der Benutzung von Cleverle. Befindet sich ein Benutzer beispielsweise auf der Ansichtsseite eines Hybridfahrzeugs und möchte diesem eine Batterie zuweisen die noch nicht im System erfasst ist, muss er im ersten Schritt auf die Listenansicht der Batterien navigieren. Im nächsten Schritt muss der Benutzer die gewünschte Batterie anlegen um anschließend wieder auf die Objektseite des Hybridfahrzeugs zu wechseln um dort die nun existierende Batterie einzutragen. Dieser Vorgang ist, obwohl recht simpel, von einem hohen Navigationsaufwand geprägt, der den Benutzer beansprucht und unnötig Zeit kostet. Ein weiteres Problem ist das erstmalige Auffinden der verfügbaren Funktionalität. Wenn ein Benutzer einen Artikel – beispielsweise ein für Tests benötigtes Werkzeug – reservieren will, kann er dies nicht über Funktionen des Artikels durchführen, sondern muss sich zuerst in die

Reservierungsverwaltung begeben um dort eine Artikelreservierung anzulegen. Der Kontext der Funktionalität ist somit nicht auf die Bedürfnisse des Benutzers ausgelegt, sondern wird stark von dem verwendeten Datenmodell beeinflusst. Speziell aus diesen Problemschilderungen ergaben sich einige wesentliche Anforderungen an die Benutzbarkeit des Systems:

Auffinden von Funktionalität

Die Oberfläche von Cleverle ist stark von den Vorstellungen der Entwickler geprägt. Für sie ist die Anbindung der Methoden an die betroffenen Objekte von der objektorientierten Programmierung her intuitiv verständlich. Der typische Benutzer hat hiermit jedoch ein erhebliches Problem, da er die Konzepte der objektorientierten Programmierung nicht kennt. Dass ein Fahrzeug nicht über eine Methode zum Reservieren desselben verfügt, sondern in einer anderen Anwendung des Systems eine Fahrzeugreservierung erstellt werden muss, ist für den Benutzer nicht ersichtlich. Dies fiel besonders durch Benutzer auf, die sich bei der Support-Hotline nach fehlenden Funktionalitäten erkundigten. Aussagen wie „Ich will doch nur ein Fahrzeug reservieren!“ sind für dieses Problem bezeichnend.

Gleichzeitig wurde in Benutergesprächen deutlich, dass der Navigationsaufwand bei der normalen, täglichen Arbeit zu hoch ist. Es kam mehrfach vor, dass ein Benutzer während der Arbeit in einer Anwendung einen Anruf erhielt, der eine Aufgabe in einer anderen Anwendung des Systems bedeutete. Um diese Aufgabe zu erledigen, musste der Benutzer die derzeitige Anwendung verlassen und zu den betroffenen Daten navigieren. Neben einem nicht unerheblichen zeitlichen Aufwand bedeutet dies für den Benutzer, dass er den derzeitigen Zustand des Systems verliert. Technisch versiertere Benutzer konnten dieses Problem einigermaßen umgehen, indem sie das System in einem neuen Register ihres Webbrowsers öffneten. Allerdings spricht auch dies nicht für eine gute Benutzbarkeit. Um diese Probleme anzugehen ist eine Schnittstelle für die Funktionalitäten nötig, welche den aktuellen Zustand der Anwendung nicht verändert und von jedem Zustand aus alle Funktionalitäten bereitstellen kann.

Um weitere Anforderungen zu erhalten, wurden Nutzerbefragungen durchgeführt. Hierzu wurden Benutzer bei der alltäglichen Verwendung von Cleverle beobachtet und sollten gleichzeitig ihre Gedanken über Probleme bei der Verwendung laut formulieren. Durch das so ermittelte Nutzerverhalten konnten Probleme

in der Benutzbarkeit identifiziert werden, die nicht durch eine Meldung bei der Support-Hotline von Cleverle erfasst wurden:

Grundfunktionalitäten ohne Aufwand erreichbar

In Cleverle ist jede für ein bestimmtes Datenobjekt verfügbare Methode nur über ein Dropdown-Menü erreichbar, dass bei jeder Ausführung neu geöffnet werden muss. Hierbei werden häufig benötigte Methoden gleich behandelt wie solche die nur sehr selten aufgerufen werden. Durch eine gewisse Verzögerung bei der Generierung der Auswahlliste benötigt der Benutzer eine nicht unerhebliche Zeit um die gewünschte Funktionalität zu lokalisieren. Um diesem entgegenzuwirken sollten Funktionalitäten die von Benutzern häufig verwendet werden – speziell Hinzufügen, Bearbeiten und Löschen – mit nur einem einzigen Klick erreichbar sein.

Filtern von Objekten

Um in Cleverle Objekte auszufiltern, muss der Benutzer in der aufklappbaren Filteransicht die für ihn wichtigen Eigenschaften der Objekte einschränken. Das Vorgehen bei der Auswahl der zu filternden Eigenschaften ist je nach Benutzer- und Aufgabenart unterschiedlich. Benutzer mit simplen Aufgaben filtern typischerweise nach Hauptmerkmalen der Objekte. Beispielsweise filtert ein Benutzer der sich hauptsächlich mit der Verwaltung von personenbezogenen Daten beschäftigt nach Personennamen und nicht nach Geburtsdatum oder Wohnort. Benutzer mit komplexeren Aufgaben verwenden hingegen auch Eingabefelder mit erweiterten Eigenschaften der Objekte. Ein Benutzer der mit der Verwaltung von Fahrzeugversuchen betraut ist, filtert beispielsweise vermehrt nach der Art eines Versuchs. Gleichzeitig ist aufgefallen, dass Benutzer aus unterschiedlichen Nutzergruppen auch unterschiedliche Eigenschaften der Objekte als aussagekräftig erachten. Ein Produktionsmitarbeiter arbeitet in der Regel mit den Produktionsnummern von Fahrzeugen, wohingegen ein Ingenieur seine Fahrzeuge anhand der intern verwendeten Bezeichnung verfolgt. Benutzer ohne Erfahrungen mit den speziellen Anwendungen in denen sie sich zum Zeitpunkt des Filtervorgangs befanden, hatten teilweise Probleme die richtigen Eingabefelder für die Filter zu finden. Dies ist bei der Fahrzeugverwaltung deutlich zu erkennen, da beispielsweise die intern verwendete Bezeichnung der Fahrzeuge der globalen

Bezeichnung ähnlich sieht, aber im Detail anders aufgebaut ist. Es sollte eine Möglichkeit gefunden werden, alle dieser Nutzerverhalten zu unterstützen.

Die gleiche Thematik findet sich auch bei der Auswahl von Objekten wieder. Objekte die in anderen Objekten als Fremdschlüssel angegeben werden können, sollen mittels Eingabe ihrer Bezeichnung ausgewählt werden können. Aus obig genannten Gründen ist es jedoch für einen Benutzer nötig, ein Objekt mittels ihm bekannten Daten auszuwählen. Um dies zu unterstützen, sollten sowohl die Filter, als auch Auswahllemente auf die unterschiedlichen Benutzerarten angepasst werden.

Navigation und Darstellungen innerhalb des Systems

Ein weiteres Problem in der Benutzbarkeit das erkannt werden konnte ist, dass die Anwender sich in ihnen unbekannten Anwendungen nicht ohne Probleme zurechtfinden. Vor allem durch die konsistente Gestaltung aller Sitemap-Knoten in der Anwendung verliert der Benutzer den Überblick über seinen derzeitigen Standpunkt im System. Eine visuelle Unterscheidung zwischen verschiedenen Knotentypen wäre hier sinnvoll.

Gleichzeitig fiel auf, dass Anwender mit der verwendeten Navigationsweise um ein Objekt zu öffnen nicht umgehen können. Um in Cleverle ein Objekt zu öffnen muss der Benutzer in der Listenansicht in der sich das jeweilige Objekt befindet auf das Objektsymbol (siehe Abbildung 5) – also die visuelle Darstellung des Objekts – klicken.

	Bezeichnung ▾	Typ	Fahrzeuggeneration	Modell	Modelljahr
	██████████	981	981I	Boxster S (Typ 981)	2013

Abbildung 5 Zeile eines Fahrzeugs in Cleverle (Vereinfacht)

Viele Benutzer versuchen stattdessen auf die identifizierenden Eigenschaften des Objekts zu klicken und sind überrascht, dass dies keine Auswirkungen hat. Hierfür sollte ein geeignetes Vorgehen ausgewählt werden.

Weiterhin stellte sich heraus, dass die Benutzer mit der Darstellung der Daten eines einzelnen Objekts wenig anfangen konnten. Die Darstellung ist in Cleverle derzeitig eine einfache Auflistung der verfügbaren, verknüpften Eigenschaften des jeweiligen Objekts, welche in entsprechenden Reiter aufgeteilt werden. Dies ist für den Benutzer nicht sofort ersichtlich und sorgte bei der Verwendung für einige Verwirrung. Aus diesem Grund wäre eine Metapher, welche die Auflistung

der Daten zu einem Objekt beschreibt, ein für das Verständnis der Benutzer wichtiger Punkt.

Aus den genannten Punkten wurden die folgenden Anforderungen von Benutzern erstellt:

Auffinden von verteilten Funktionalitäten

- Der Benutzer soll sich mit möglichst geringem Aufwand über die Funktionalitäten informieren können, die für ein beliebiges Objekt existieren.
- Der Benutzer soll sich mit möglichst geringem Aufwand über die Funktionalitäten informieren können, die in dem derzeitigen Zustand der Anwendung verfügbar sind.

Zentrale Benutzerschnittstelle für das Ausführen von Funktionen

- Der Benutzer soll alle Funktionalitäten des Systems aufrufen können, ohne zu einem speziellen Punkt in der Anwendung navigieren zu müssen.
- Der entstehende zeitliche Aufwand eines erfahrenen Benutzers bei der Auswahl der Funktionalität soll gering gehalten werden.

Verringerung des Navigationsaufwands

- Ein geübter Benutzer soll ohne großen zeitlichen Aufwand zu für ihn wichtige Punkte in der Anwendung gelangen können.

Umstieg von Cleverle mit geringem Lernaufwand

- Ein geübter Cleverle-Benutzer soll ohne großen Lernaufwand die neue Benutzeroberfläche verwenden können. Er soll die Erlernbarkeit auf einer Skala von „Schwer erlernbar“ bis „Leicht erlernbar“ mit einer Tendenz zu „Leicht erlernbar“ bewerten.

3.1.2 Anforderungen der Entwickler

Bei Gesprächen mit den Entwicklern von Cleverle stellte sich heraus, dass es speziell bei der Bereitstellung von Funktionalitäten große Schwächen gibt. Hat ein Entwickler die Aufgabe, eine Funktionalität auch nur geringfügig zu ändern, muss er den Quellcode der Hauptanwendung anpassen. Da die entwickelte

Funktionalität aus diesem Grund nicht selbstständig ausgeliefert werden kann, muss das komplette System neu veröffentlicht werden. In den Vorgaben des Unternehmens ist hierfür festgelegt, dass dies monatlich geschieht, weshalb die Funktionalität gegebenenfalls erst nach einer Zeit von einem Monat für den Benutzer verfügbar ist.

Als Mangel an der Verwaltung von Cleverle wurde auf den hohen Konfigurationsaufwand verwiesen, der nötig ist um simple Darstellungen von Objekten zu ermöglichen. Möchte ein Entwickler eine Einsicht in die Objekte einer Klasse bereitstellen, so muss er zuerst alle anzuzeigenden Ansichten erstellen und diesen die relevanten Eigenschaften zuweisen. Tut er dies nicht, gibt es keine Möglichkeit die Daten einzusehen. Hierfür wären automatisch generierte Daten mit sinnvollen Standards eine denkbare Lösung.

Weiter ist die Erstellung von Navigationsstrukturen für neue Anwendungen ein Aufwand, der von den Entwicklern als zu hoch eingestuft wird. Die Entwickler müssen für diese Tätigkeit zuerst in die Administrationsanwendung navigieren, um dort in einer untergliederten Ansicht die nötigen Elemente anzulegen. In dieser Ansicht müssen sie die Anwendungsstruktur als Baumstruktur anlegen und die Inhalte der einzelnen Knoten mittels fester URL angeben.

Eine von den Entwicklern konkret gestellte Anforderung war die einfache Änderung von Bestandteilen wie den Eingabemasken ohne diese Änderungen für jede betroffene Klasse einzeln durchzuführen. Speziell bei den Dialogen für Methoden ist es in Cleverle heute nicht möglich gewisse Bestandteile wie beispielsweise die verwendeten Eingabeboxen auszutauschen. Hierfür müssen die Entwickler durch alle Methodendialoge hinweg die Dialoge anpassen, was bei einer Anzahl von über 400 erweiterten Methoden einen erheblichen Aufwand bedeutet.

Aus den von Entwicklern genannten Wünschen und Problemen wurden folgende Anforderungen erstellt:

Generische Oberfläche mit sinnvollen Standards

- Um eine standardmäßige Einsicht in angebundene Datenquellen zu ermöglichen, soll keine Konfiguration der Daten der Oberfläche nötig sein.

Einfacheres Erstellen von Anwendungsstrukturen

- Anwendungsstrukturen sollen direkt in der Hauptansicht der Anwendung wie jegliche andere Objekte angelegt werden können.
- Ein erfahrener Entwickler soll zum Anlegen einer Anwendungsstruktur mit einer Hauptanwendung und drei Unterpunkten nicht länger als zehn Minuten benötigen.

Einfacheres Bereitstellen von Funktionalität

- Neu entwickelte oder geänderte Funktionalität soll zur Laufzeit bereitgestellt werden können.
- Nach erfolgreicher Änderung der Funktionalität soll ein erfahrener Entwickler höchstens fünf Minuten benötigen um diese bereitzustellen.

3.2 Anwendungsfälle

Die in 3.1.1 und 3.1.2 genannten Probleme bei der Verwendung des Systems sind im Folgenden anhand vier typischer Anwendungsfälle beschrieben. Diese Anwendungsfälle treten täglich auf und zeigen den nötigen Aufwand und die problematischen Stellen bei der Durchführung der von ihnen beschriebenen Tätigkeiten. Auffallend ist, dass die Anwendungsfälle einen sehr hohen Anteil an Navigationsschritten beinhalten, was auch als das Hauptproblem des Systems gesehen wird.

Der Anwendungsfall „Personalliste exportieren“ (Tabelle 4) wird von Personalverwaltern häufig ausgeführt wenn sie die aktuell angestellten Personen einer speziellen Abteilung als Bericht benötigen. Dieser Bericht wird von der Business-Logik der an Abteilungen hängenden Methode „CreateReport“ generiert und dem Benutzer anschließend als Download angeboten. Um den Bericht zu exportieren muss der Benutzer zuerst zu der gewünschten Abteilung navigieren und dort anschließend die entsprechende Aktion auswählen.

Im Anwendungsfall „Fahrzeug reservieren“ (Tabelle 5) ist das typische Verfahren beschrieben um ein beliebiges Fahrzeug zu reservieren. Hier ist auffallend, dass die Funktionalität nicht an der Stelle existiert an der sie ein Benutzer erwarten würde. Anstelle einer Aktion „reservieren“ auf einem Fahrzeug muss der Benutzer die Aktion „Neu“ auf einer Reservierung ausführen. Dies ist dem Datenmodell anzurechnen, da dort Reservierungen als eigenständige Klasse modelliert sind, die Fahrzeugen zugewiesen werden können. Diese Unstimmigkeit zwischen Erwartungen der Benutzer und realem Verhalten des Systems ist problematisch.

Um eine Änderung an Funktionalitäten durchzuführen müssen die im Anwendungsfall „Funktionalität ändern“ (Tabelle 3) beschriebenen Schritte getätigert werden. Der entstehende Aufwand ist für den Entwickler hierbei eher gering, da er seine Änderungen nur in der Versionsverwaltung registrieren muss und für ihn danach keine weiteren Schritte nötig sind. Jedoch muss für eine Bereitstellung der Funktionalität die gesamte Webapplikation neu aufgestellt und veröffentlicht werden. Diese Veröffentlichung ist durch Unternehmensrichtlinien geregelt und wird nur monatlich durchgeführt. Es ist dem Entwickler nicht möglich, den Benutzern die Funktionalität zu einem früheren Zeitpunkt bereitzustellen.

Der in Tabelle 6 beschriebene Anwendungsfall „Anwendungsstruktur erstellen“ beschreibt die Durchführung einer typischen Aufgabe bei der Erstellung einer neuen Teilanwendung in Cleverle. Hierbei wird die für den Benutzer sichtbare Darstellung der Anwendung erstellt und konfiguriert. Für die Navigation werden sogenannte SiteMap-Knoten verwendet, die hierarchisch strukturiert sind. Auf oberster Ebene existiert ein Knoten „Anwendungen“ der alle verfügbaren Anwendungen darstellt. Die zu diesem Knoten gehörigen Unterknoten repräsentieren folglich einzelne Anwendungen für die jeweiligen Themenbereiche. Um die Struktur einer neuen Anwendung anzulegen muss ein Entwickler in den entsprechenden Administrationsbereich navigieren und ausgehend von der obersten Ebene die SiteMap-Knoten erstellen. Bei der Erstellung der jeweiligen Knoten muss der Entwickler im nächsten Schritt Titel, Beschreibung und eine URL angeben. Diese URL gibt den durch einen Klick auf den Knoten anzuzeigenden Inhalt an. Um eine Übersicht der untergeordneten SiteMap-Knoten anzuzeigen muss beispielsweise eine andere URL angegeben werden als wenn eine Liste an Objekten einer speziellen Klasse angezeigt werden sollte. Die Parametrisierung der Inhalte erfolgt hierbei auch direkt über die URL und kann nicht als Eigenschaften der Knoten angegeben werden.

Die Anwendungsfälle sind nach (Ludewig, et al., 2007) aufgebaut.

Name	Funktionalität ändern
Ziel	Entwickler möchte bestehende Funktionalität ändern und den Benutzern verfügbar machen
Vorbedingung	Entwickler hat einen Fehler in einer bestehenden Funktionalität gefunden
Nachbedingung	Die Änderungen an der Funktionalität sind in der Versionsverwaltung registriert. Durch festgelegte Veröffentlichungstermine wird sie den Benutzern erst später zur Verfügung gestellt.
Akteure	Entwickler
Normalablauf	<ol style="list-style-type: none"> 1. Entwickler setzt seine Änderungen im Quellcode der Webanwendung um 2. Entwickler lädt die Änderungen in die Versionsverwaltung hoch

Tabelle 3 Anwendungsfall "Funktionalität ändern"

Name	Personalliste exportieren
Ziel	Benutzer möchte eine Personalliste zu einer speziellen Abteilung exportieren
Vorbedingung	Benutzer befindet sich in der Anwendungsübersicht
Nachbedingung	Bericht wurde generiert und wird dem Benutzer als Download angeboten
Nachbedingung im Sonderfall	Bericht wurde nicht generiert
Akteure	Benutzer, System
Normalablauf	<ol style="list-style-type: none"> 1. Benutzer sucht die Anwendung „Personen- & Organisationsverwaltung“ und klickt auf die Schaltfläche zum Öffnen 2. System öffnet die Ansichtsseite der Anwendung 3. Benutzer sucht den Unterknoten „Organisationseinheiten“ und klickt auf die Schaltfläche zum Öffnen 4. System öffnet die Ansichtsseite der „Organisationseinheiten“ 5. Benutzer klickt auf Schaltfläche „Filtern“ 6. System zeigt Filterliste an 7. Benutzer gibt den Namen der gesuchten Abteilung ein 8. Benutzer klickt auf die Schaltfläche „Filtern“ 9. System zeigt alle Abteilungen mit dem eingegebenen Namen an 10. Benutzer klickt auf das Abteilungssymbol, welches die gesuchte Abteilung repräsentiert, um diese zu öffnen 11. System öffnet die Ansichtsseite der Abteilung 12. Benutzer klickt auf die Schaltfläche „Aktionen“ 13. System öffnet ein Dropdown-Menü mit den verfügbaren Aktionen 14. Benutzer klickt auf die Schaltfläche „Bericht erstellen“ 15. System zeigt Popup zur Berichterstellung an 16. Benutzer wählt den Bericht „Personalbericht“ aus und gibt die relevanten Daten ein 17. Benutzer klickt auf „Bericht erstellen“
Sonderfälle	<ol style="list-style-type: none"> 8. Die gesuchte Abteilung existiert nicht <ol style="list-style-type: none"> a. System zeigt keine Daten an

Tabelle 4 Anwendungsfall „Personalliste exportieren“

Name	Fahrzeug reservieren
Ziel	Benutzer möchte ein Fahrzeug reservieren
Vorbedingung	Benutzer befindet sich in der Anwendungsübersicht
Nachbedingung	Die Reservierung wurde angelegt und wird dem Benutzer angezeigt
Nachbedingung im Sonderfall	Die Reservierung wurde nicht erstellt
Akteure	Benutzer, System
Normalablauf	<ol style="list-style-type: none"> 1. Benutzer sucht die Anwendung „Reservierungsverwaltung“ und klickt auf die Schaltfläche zum Öffnen 2. System öffnet die Ansichtsseite der Anwendung 3. Benutzer sucht den Unterknoten „Reservierungen“ und klickt auf die Schaltfläche zum Öffnen 4. System öffnet die Ansichtsseite der Reservierungen 5. Benutzer klickt auf die Schaltfläche „Aktionen“ 6. System öffnet ein Dropdown-Menü mit den verfügbaren Aktionen 7. Benutzer fährt mit der Maus über die Schaltfläche „Neu“ 8. System öffnet ein weiteres Dropdown-Menü mit den verfügbaren Reservierungsarten 9. Benutzer klickt auf die Schaltfläche „Fahrzeug-Reservierung“ 10. System öffnet ein Popup zur Erstellung von Reservierungen 11. Benutzer gibt die relevanten Daten ein 12. Benutzer klickt auf die Schaltfläche „Weiter“ 13. System zeigt eine Liste zur Auswahl der zu reservierenden Fahrzeuge an 14. Benutzer filtert die angezeigten Fahrzeuge 15. Benutzer wählt die zu reservierenden Fahrzeuge aus 16. Benutzer klickt auf die Schaltfläche „OK (Schließen)“
Sonderfälle	<ol style="list-style-type: none"> 14. Das gesuchte Fahrzeug existiert nicht <ol style="list-style-type: none"> a. System zeigt keine Daten an 16. Der Benutzer hat keine Berechtigungen das Fahrzeug zu reservieren <ol style="list-style-type: none"> a. System zeigt eine Fehlermeldung an

Tabelle 5 Anwendungsfall "Fahrzeug reservieren"

Name	Anwendungsstruktur erstellen
Ziel	Entwickler möchte eine App mit untergeordneten Knoten anlegen
Vorbedingung	Entwickler befindet sich in der Anwendungsübersicht
Nachbedingung	Die Anwendungsstruktur wurde erstellt und ist verfügbar
Akteure	Entwickler, System
Normalablauf	<ol style="list-style-type: none"> 1. Entwickler sucht die Anwendung „Administration“ und klickt auf die Schaltfläche zum Öffnen 2. System zeigt die Ansichtsseite der Anwendung an 3. Entwickler sucht den Unterknoten „Repository-Konfiguration“ und klickt auf die Schaltfläche zum Öffnen 4. System zeigt die Ansichtsseite des Unterknoten an 5. Entwickler sucht den Unterknoten „SiteMap-Knoten“ und klickt auf die Schaltfläche zum Öffnen 6. System zeigt die Ansichtsseite der SiteMap-Knoten an 7. Entwickler markiert in der angezeigten, hierarchischen Liste der SiteMap-Knoten den Knoten „Anwendungen“ 8. System zeigt die Details des ausgewählten Knoten an 9. Entwickler klickt auf den Reiter „Kinderknoten“ 10. System zeigt die untergeordneten Knoten in einer Liste an 11. Entwickler klickt auf die Schaltfläche „Aktionen“ 12. System öffnet Dropdown-Menü mit verfügbaren Aktionen 13. Entwickler klickt auf die Schaltfläche „Neu“ 14. System öffnet Popup zum Erstellen von SiteMap-Knoten 15. Entwickler gibt die relevanten Daten ein 16. Entwickler klickt auf die Schaltfläche „Anlegen“ 17. System erstellt SiteMap-Knoten mit den eingegebenen Daten und aktualisiert die untergeordneten Knoten 18. Entwickler navigiert zu dem neu erstellten SiteMap-Knoten und wiederholt Schritte 8-18 bis die Struktur angelegt ist 19. Entwickler klickt auf Schaltfläche für Administrationstools 20. System öffnet ein Dropdown-Menü mit Möglichkeiten 21. Entwickler klickt auf die Schaltfläche „Zustandsinformationen zurücksetzen“ 22. System öffnet Popup zur Auswahl der Informationen 23. Entwickler markiert nur den Eintrag „Sitemap neu laden“ 24. Entwickler klickt auf die Schaltfläche „Markierte neu laden“

Tabelle 6 Anwendungsfall "Anwendungsstruktur erstellen"

4 Lösungsansätze/Konzepte

Durch die eingeschränkt verfügbare Zeit zur Durchführung der Diplomarbeit musste entschieden werden, sich auf einige in der Analyse erkannte Probleme zu fokussieren. Im Folgenden werden Lösungsansätze zu den erkannten Problemen beschrieben und diese anschließend anhand eines Kosten-Nutzen-Verhältnisses bewertet. Aufgrund dieser Bewertung werden die Lösungsansätze ausgesucht, die im Rahmen dieser Arbeit weiter verfolgt werden. Andere Probleme mit der Benutzbarkeit werden im weiteren Verlauf der Arbeit nicht behandelt.

4.1 Filtern von Objekten

In der Analyse zeigte sich, dass ein Teil der Benutzer die Objekte nach ihren Haupteigenschaften wie Anzeigennamen filtert, ein anderer Teil jedoch auch andere Eigenschaften zur Einschränkung verwendet. Eine generelle Auflistung aller filterbaren Eigenschaften ist für einen Großteil der Benutzer somit verwirrend, wohingegen eine Reduzierung auf wichtige Eigenschaften für andere Benutzer gewünschte Funktionalitäten entfernen würde. Es sollte aus diesen Gründen für jede der beiden Benutzergruppen eine Möglichkeit geben, nach den gewünschten Kriterien zu filtern. Ein Vorschlag hierzu wäre eine einfache Suchleiste zum Filtern nach den wichtigsten Eigenschaften der Objekte und eine gleichzeitig angebotene erweiterte Filterauswahl. Zur schnellen Einschränkung der angezeigten Daten können bei diesem Vorgehen über die Suchleiste Stichworte eingegeben werden. Werden hingegen spezifischere Filter wie eine Einschränkung eines Erstelltdatums auf einen speziellen Zeitraum benötigt, können diese über eine separat zu öffnende Liste an Filterkriterien angegeben werden.

4.2 Navigation und Darstellungen innerhalb des Systems

Um von Cleverle auf das neue System umsteigenden Benutzern keinen hohen Lernaufwand zuzumuten, soll das alte Navigationskonzept beibehalten werden. Dieses sieht vor, alle Navigationselemente als Kacheln darzustellen. Diese

Kacheln sollen ansprechender entworfen werden um der Anwendung ein attraktiveres Aussehen zu verleihen.

Für ein besseres Verständnis der dargestellten Daten sollen diese Daten wie auf Blöcken geschrieben dargestellt werden (siehe Abbildung 6). Ein einzelnes Objekt hat hierbei in seiner Standardansicht als erstes Blatt des dargestellten Blocks seine zugehörigen Details aufgelistet. Über an dem Block angeordnete Register kann der Benutzer zwischen den Seiten des Blocks wechseln und so die zu dem Objekt gehörenden Relationen einsehen. Für eine möglichst realistische Darstellung der Blockansicht sollen an der oberen Kante des Blocks Ringe als Ringbindung dargestellt, und an der unteren Kante die existierenden Register angezeigt werden.

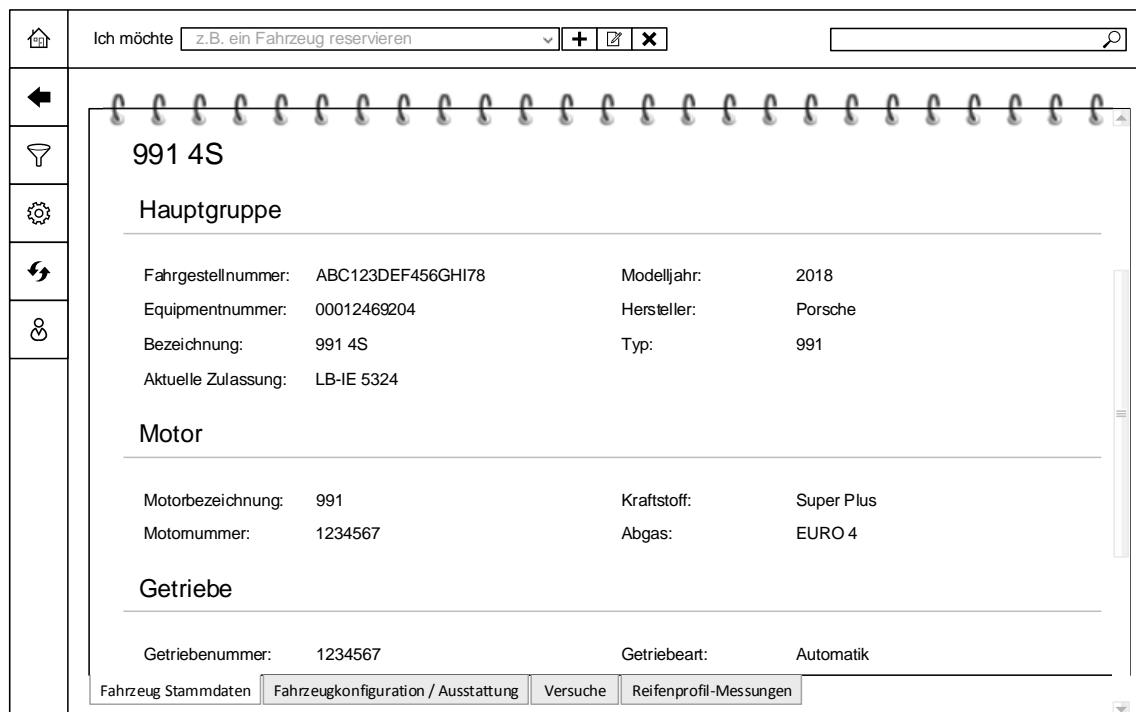


Abbildung 6 Visualisierte Block-Metapher

4.3 Schnellnavigation innerhalb des Systems

Für die schnelle und unkomplizierte Navigation zwischen für den Benutzer wichtigen Anwendungen oder Datenobjekten soll eine Schnellnavigation eingeführt werden. In dieser Schnellnavigation soll der Benutzer die Möglichkeit haben, ihm wichtige Positionen im System zu hinterlegen und später mit wenig Aufwand wie-

der zu öffnen. Der Benutzer soll jede Art von angezeigten Objekten als Schnellnavigationsobjekt definieren, sowie die angelegten Schnellnavigationsobjekte priorisieren und löschen können.

4.4 Navigieren zu Objekten

Das Öffnen von Objektseiten mittels in den Listen angezeigten Symbolen wurde in der Analyse als unverständlich und nicht intuitiv erkennbar eingestuft. Um dem entgegenzuwirken sollte eine andere Art der Interaktion gewählt werden um auf Objekte zu navigieren. Denkbare Alternativen wären hierbei ein Öffnen mittels Klick auf den Anzeigenamen, sowie ein Klick auf die komplette Zeile die das relevante Objekt repräsentiert. Je nach Kontext des Objekts sollte dies unterschiedlich ausgewählt werden. Objekte die im Kontext eines anderen Objekts, also mittels eines Anzeigenamens dargestellt werden, bieten sich schon von vornherein für eine Navigation mittels Klick auf den Anzeigenamen an. Wird jedoch eine Liste mit den zu öffnenden Objekten angezeigt, ist ein solcher Anzeigename nicht zwangsweise vorhanden. In diesen Fällen würde sich eine Navigation über einen Klick auf die Zeile welche die jeweiligen Objekte repräsentieren anbieten.

4.5 Generische Oberfläche

Eine beim Entwurf der Benutzerschnittstelle technologiebedingte Besonderheit von föderierten Informationssystemen ist, dass Objekte verschiedenster Arten und mit unterschiedlichsten Eigenschaften angezeigt werden müssen. Für jedes Objekt die Oberflächenelemente wie Eingabe- und Editiermasken, Objektansichten, Ansichten zum Löschen und Listenansichten einzeln anzulegen, wäre bei der enormen Datenmenge ein nicht vertretbarer Aufwand. In Cleverle werden zum Zeitpunkt der Verfassung dieser Arbeit 1983 Klassen verwaltet, für welche jeweils die genannten Ansichten erstellt werden müssten. Dies ist in vielen Unternehmen der Fall, wenn die entwickelten Anwendungen nicht durch ein föderiertes Informationssystem zusammengefasst, sondern je nach Bedarf von den Fachbereichen selbst entwickelt werden. Die Anwendungen sind dadurch unterschiedlich aufgebaut und verfügen somit über unterschiedliche Benutzeroberflächen, wodurch sich die Benutzer nie an eine Bedienungsart gewöhnen können.

Um eine gute Benutzbarkeit der verschiedenen Anwendungen zu gewähren, sollten deren Benutzeroberflächen jedoch gleich gehalten werden. Aus diesen Gründen ist ein generischer Entwurf der Oberfläche sinnvoll, da dieser für alle angelegten Objekte verwendet werden kann und ein konsistentes Aussehen hat.

4.6 Generische Aktionen

In Cleverle werden sowohl die Oberfläche von aufgerufenen Methoden, als auch deren Funktionalität fest im Quellcode des Systems implementiert. Dies resultiert in starren Vorgehensweise bei der Entwicklung oder Veränderung von Funktionalitäten. Möchte ein Entwickler eine neue Methode einführen, so muss er zuerst die Methode in den Metadaten von Cleverle definieren. Anschließend muss er die Funktionalität und die Oberfläche der Methode implementieren und in den Quellcode von Cleverle einbinden. Nach der Entwicklung besteht für den Entwickler keine Möglichkeit, dem Benutzer die Funktionalität verfügbar zu machen, da eine neue Version von Cleverle nur zu festgelegten Zeitpunkten veröffentlicht wird. Diese Veröffentlichung findet typischerweise monatlich statt, weshalb der anfordernde Benutzer unnötig lange Wartezeiten bei der Bereitstellung seiner gewünschten Funktionalitäten in Kauf nehmen muss. Besser wäre es, wenn die Funktionalität vom Quellcode der Basisanwendung getrennt wäre und somit schon während der Laufzeit bereitgestellt werden könnte. Um dies zu ermöglichen, ist es notwendig die Funktionalität in den Daten zu modellieren und zu verwalten.

4.7 Schnittstelle zur Lokalisierung von Funktionalität

Die Schnittstelle zur Lokalisierung der Funktionalitäten – im Folgenden TellMe-Eingabeleiste genannt – soll als Auswahl der betreffenden Klasse und der zugehörigen Aktion realisiert werden. Die Schnittstelle soll den Benutzern in der Kopfleiste der Anwendung zu jedem Zeitpunkt zur Verfügung stehen, damit diese von jeder Stelle im System aus die benötigten Funktionalitäten aufrufen können. Um das gleichzeitige Ausführen von unterschiedlichen Tätigkeiten nicht zu behindern soll sich das System nach Ausführung der Aktion in demselben Zustand befinden wie zuvor.

4.8 Bewertung

Um eine Bewertung der Lösungsansätze zu erhalten wird ein Kosten-Nutzen-Verhältnis erstellt. Hierfür wird anhand drei Kriterien und einem geschätzten benötigten Aufwand A_T in Tagen für die Umsetzung der Themen ein Kosten-Nutzen-Verhältnis (KNV) errechnet. Die Kriterien Bedeutung (K_B), Häufigkeit (K_H) sowie Aufwand (K_A) sind im Folgenden beschrieben:

Bedeutung

Welchen geschätzten Einfluss hätte eine Umsetzung des Konzepts auf die Benutzbarkeit des Systems, beziehungsweise wie stark werden Benutzer in Cleverle durch die Probleme des bewerteten Themas belastet? Die Bewertungsskala geht hierbei von „Kein Einfluss/Keine Beeinträchtigung“ bis „Sehr großer Einfluss/Unzumutbar“.

Häufigkeit

Wie häufig tritt ein Problem auf, beziehungsweise wie häufig würde der Benutzer oder Entwickler durch eine Umsetzung profitieren? Die Bewertungsskala geht bei diesem Kriterium von „Auftreten unwahrscheinlich“ bis „Tritt immer auf“.

Aufwand

Wieviel Mehraufwand hat der Benutzer durch die beschriebenen Probleme, oder wieviel Aufwand könnte dem Benutzer oder Entwickler durch eine Umsetzung erspart werden? Die Bewertung des Kriteriums liegt zwischen „Kein Aufwand/Kein Einsparungspotential“ und „Konzept hinfällig/Sehr hohes Einsparungspotential“.

In Tabelle 7 wird das Kosten-Nutzen-Verhältnis der beschriebenen Themen bewertet. Die grau hinterlegten Themen wurden für eine Umsetzung und Evaluation im Rahmen dieser Diplomarbeit ausgewählt.

Thema	A_T	K_B	K_H	K_A	KNV
<i>Schnellnavigation innerhalb des Systems</i>	5	6	8	6	57,60
<i>Generische Oberfläche</i>	20	8	7	9	25,20
<i>Schnittstelle zur Lokalisierung von Funktionalitäten</i>	25	8	8	9	23,04
<i>Generische Aktionen</i>	15	7	6	8	22,40
<i>Navigation innerhalb des Systems</i>	10	4	9	6	21,60
<i>Filtern von Objekten</i>	7	5	6	4	17,14
<i>Navigieren zu Objekten</i>	2	2	8	2	16,00

Tabelle 7 Bewertung von Lösungsansätzen

5 Evaluation anhand einer prototypischen Umsetzung

In diesem Kapitel wird beschrieben, wie die in 4.8 ausgewählten Lösungsansätze umgesetzt werden. Hierzu wird zunächst beschrieben, wie bei der Umsetzung und Validierung der Ergebnisse vorgegangen wird. Um bei den Evaluierungen über eine möglichst repräsentative Auswahl an Testnutzern zu verfügen, werden die aktiven Cleverle-Benutzer nachfolgend analysiert und die Verteilung verschiedener Kriterien ermittelt. Anschließend wird die erstmalige Umsetzung der Lösungsansätze beschrieben. In den folgenden untergeordneten Kapiteln wird die iterative Bewertung und Verbesserung der Benutzeroberfläche erläutert. Zuletzt wird die abschließende Evaluation des Prototyps durchgeführt und die erhaltenen Ergebnisse zusammengefasst und analysiert.

5.1 Vorgehen

Da schon zu Beginn dieser Diplomarbeit ein gewisser Stand des neu zu entwickelnden föderierten Informationssystems Cluu verfügbar war, bot es sich an, die Oberfläche nach dem Konzept des Prototyping zu entwickeln. Während der Umsetzung der Ideen zum Entwurf der Oberfläche war stets eine Version des Systems verfügbar, mit der die getroffenen Entscheidungen evaluiert werden konnten. Diese Evaluationen erfolgten zum größten Teil informell mittels interner Tests, welche mit Cleverle-Anwendern durchgeführt wurden. Da Cluu intern schon zu einem frühen Entwicklungsstand zur Zeiterfassung genutzt wurde, war ein Feedback zu neu umgesetzten Ideen schnell vorhanden.

Für die abschließende Evaluation des Systems werden Usability Tests eingesetzt bei deren Durchführung die Testnutzer die Benutzbarkeit des Systems testen können. Sie werden hierbei von vier gestellten Aufgaben geleitet, die gleichzeitig dazu dienen, Ausführungszeiten und Vorgehensweisen zu ermitteln. Die gestellten Aufgaben haben außerdem den Nutzen, dass ermittelt werden kann, wie häufig Benutzer die TellMe-Eingabeleiste anstelle der bisherigen Vorgehensweise verwenden. Speziell diese Erkenntnis gibt Aufschluss über die Nützlichkeit und Akzeptanz der Nutzer für eine solche zentrale Schnittstelle zur Auffindung

der Funktionalität. Für die Vergleichbarkeit der Testergebnisse werden diese abschließenden Tests mit jedem Benutzer mit demselben Stand der Anwendung und einer gleichbleibenden Testdatenbank ausgeführt. Durch diese Vorkehrungen können die Tests unter den gleichen Voraussetzungen durchgeführt werden und es kommt zu keiner Beeinträchtigung der Tests durch veränderte Daten.

Die vier verwendeten Aufgaben orientieren sich an den Anwendungsfällen und sind in zwei Kategorien aufgeteilt: Die ersten beiden Aufgaben sollen von allen Testern ausgeführt werden um die Benutzbarkeit des Systems bei allgemeinen Nutzungsszenarien zu ermitteln. Die letzten beiden Aufgaben dienen der Ermittlung der Benutzbarkeit bei Tätigkeiten der Entwicklung und Verwaltung.

Die Aufgaben und deren Szenarien sind wie folgt definiert:

1. Der Benutzer benötigt eine exportierte Personalliste für seine Abteilung „TST4“. Er soll die nötigen Schritte durchführen um diese zu erstellen. Nach einer Überprüfung der Liste fällt ihm auf, dass der Name eines Mitarbeiters inkorrekt ist und eine Mitarbeiterin nicht in der Liste auftaucht. Er soll die nötigen Schritte tätigen um diese fehlerhaften Daten zu korrigieren.
2. Der Benutzer muss Tests an einem Fahrzeug durchführen und dieses Fahrzeug hierfür reservieren. Hierzu soll der Benutzer zuerst überprüfen, wann das Fahrzeug als nächstes verfügbar ist und es anschließend reservieren.
3. Der Entwickler hat einige Fehler in der Implementierung einer Funktionalität korrigiert und möchte diese nun den Benutzern verfügbar machen. Hierzu soll er die Implementierung der betroffenen Aktion austauschen und sich anschließend versichern, dass die Funktionalität nun korrekt ist.
4. Der Entwickler erstellt eine neue Anwendung für die Benutzer. Hierzu soll er die Navigationsstruktur mit Anwendung und Unterknoten anlegen. Da für die Anwendung und Navigationsknoten Bilder hinzugefügt werden müssen, kann speziell die Verwendung der TellMe-Eingabeleiste beobachtet werden. Mithilfe dieser könnte der Entwickler die Bilder, ohne die Ansicht zu wechseln hinzufügen.

Während der Durchführung der Aufgaben werden sowohl die benötigte Zeit, als auch die begangenen Fehler notiert. Um eine Aussage über die Nützlichkeit der TellMe-Eingabeleiste treffen zu können, wird außerdem festgehalten, ob der Benutzer diese für seine Tätigkeiten verwendet. Des Weiteren wird der Benutzer

aufgefordert, seine Gedanken laut zu äußern und sich mit der Darstellung der Elemente auseinanderzusetzen. Nach der Bearbeitung jeder Aufgabe erhält der Benutzer ein Formular des NASA-TLX welches die Belastung des Benutzers während der Ausführung erfasst. Die Ergebnisse dieser Tests können im Anschluss mit den spezifizierten Nutzer- und Entwickleranforderungen verglichen werden und eine Aussage über die Güte der Benutzbarkeit getroffen werden.

Im Anschluss an die Aufgaben soll der Benutzer die Verwendung des Systems bewerten. Hierzu werden ihm einige Fragen gestellt die seine Meinung der Anwendung, auch im Vergleich zum Vorgängersystem Cleverle erfassen sollen. Zusätzlich soll er mittels eines semantischen Differentials sein Empfinden anhand der in 2.2.8 beschriebenen Gegensatzpaaren einstufen.

5.2 Aufbau der Nutzergruppen

In Cleverle sind zum Zeitpunkt dieser Diplomarbeit 5341 Benutzer registriert. Von diesen 5341 haben sich 1782 innerhalb des vorletzten Monats dieser Diplomarbeit angemeldet und das System verwendet. In Cleverle werden für automatisierte Abläufe sogenannte technische Benutzer verwendet, welche keinen realen Personen zugeordnet sind. Die Anzahl dieser technischen Benutzer beläuft sich auf 20, womit Cleverle von insgesamt 1762 konkreten Personen aktiv verwendet wird.

Um eine Verteilung des Benutzeralters zu erhalten, wurden die Geburtsdaten der aktiven Benutzer ausgewertet. Von den 1762 aktiven Benutzern hatten 175, also etwa 10% ihr Geburtsdatum angegeben. Da die Geburtsdaten von Abteilungsverwaltern und nicht den Benutzern selbst eingetragen wird, können Einflüsse der Benutzer auf das Vorhandensein des Geburtsdatums ausgeschlossen werden. Die Altersverteilung ist wie in Abbildung 7 dargestellt. Der Großteil der Benutzer (51%) ist zwischen 26 und 34 Jahren alt, der nächstkleinere Teil (18%) zwischen 35 und 40 Jahren. Neben einer weiteren ausschlaggebenden Gruppe der 43-47-Jährigen (13%), verteilen sich die restlichen 18% relativ gleichmäßig, mit dem Alter abfallend auf die übrigen Jahrgänge.

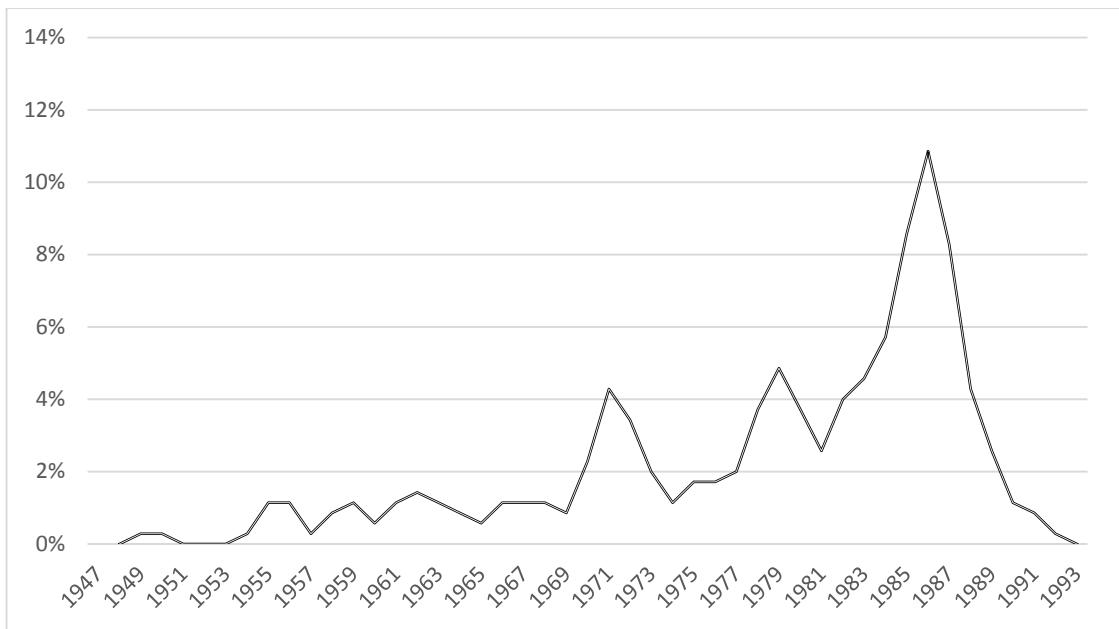


Abbildung 7 Altersverteilung von aktiven Cleverle-Benutzern

Das Verhältnis von Männern zu Frauen kann in Cleverle einfach ausgewertet werden, da bei den gespeicherten Personen auch das Geschlecht hinterlegt ist. Aus diesen Daten geht hervor, dass das Verhältnis bei 4:1 (82% zu 18%) liegt.

Für eine abschließende Evaluation der Umsetzungen mit repräsentativen Testnutzern, werden diese anhand der ermittelten Zusammensetzung der Cleverle-Benutzer ausgewählt. Die Evaluation wird mit insgesamt fünf Testnutzern durchgeführt, die einen Hochschul- oder Fachhochschulabschluss in einer Ingenieurwissenschaft besitzen. Für eine Unterscheidung des Ease-of-Learning und des Ease-of-Use wurde das System mit Testnutzern mit unterschiedlichen Vorkenntnissen getestet. Drei der ausgewählten Testnutzer hatten noch nie mit Cleverle gearbeitet wohingegen die anderen zwei es für ihre tägliche Arbeit verwenden. Drei der Testnutzer stufen ihre Kenntnisse mit Computern als sehr hoch, zwei weitere als mittel bis hoch. Die Altersverteilung der Testnutzer liegt innerhalb der ermittelten Schwerpunkte zwischen 26 und 40 Jahren.

5.3 Prototypische Umsetzung der Lösungsansätze

Für eine Evaluation der Lösungsansätze werden diese anhand eines Prototyps umgesetzt. Dieser Prototyp wird durch die Evaluationen hinweg kontinuierlich verbessert und die in den Lösungsansätzen beschriebenen Konzepte überarbeitet.

5.3.1 Generische Oberfläche

Die Oberfläche des Prototyps wurde im Vergleich zum Vorgängersystem deutlich reduziert (siehe Abbildung 8). Es existieren nur noch drei Bereiche innerhalb der Anwendung. Eine einzige Kopfzeile stellt Interaktionsmöglichkeit mit

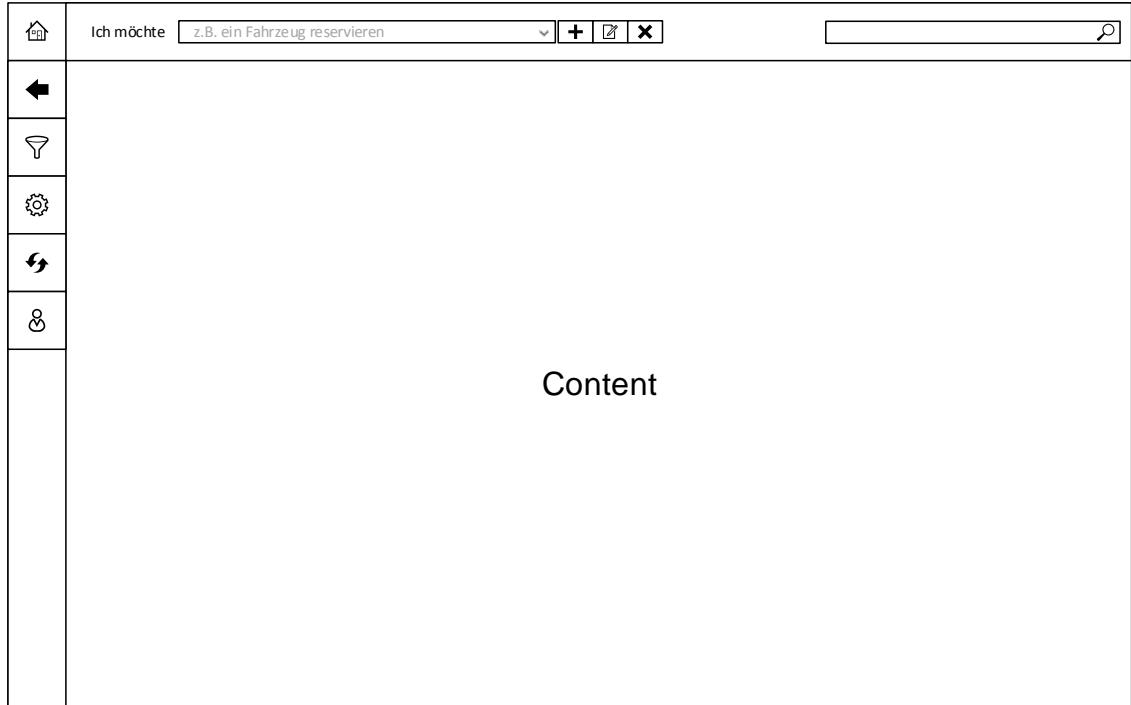


Abbildung 8 Aufteilung der Oberfläche

den Daten zur Verfügung. Eine Seitenleiste bietet hingegen Interaktionsmöglichkeiten an, welche die Darstellung der Daten, sowie das System an sich betreffen. Neben der Schnittstelle zum Auffinden der Funktionalitäten werden in der Kopfzeile die grundlegenden Aktionen (Erstellen, Bearbeiten, Löschen) des aktuell angezeigten Objekts dargestellt. Als weiteres Element wird eine Suchleiste angeboten, die es dem Benutzer später ermöglichen soll, alle angezeigten Daten zu durchsuchen. In der Seitenleiste des Systems befindet sich als oberstes Element die Schnellnavigation auf die Hauptseite der Anwendung. Unterhalb dieser Schaltfläche werden die Navigation innerhalb der Anwendungshierarchie, sowie Filtermöglichkeiten angeboten. Neben einer Aktualisierung und einer Auswahl der Darstellung der angezeigten Daten, erhält der Benutzer die Möglichkeit, seine Anmeldung am System zu verwalten. Weiterhin werden in der Seitenleiste die Favoriten des angemeldeten Benutzers dargestellt. Die vom Benutzer gewünschten Daten werden innerhalb des Content-Bereichs angezeigt. Die Darstellung der Daten kann hierbei in den Daten des Systems beschrieben und zur Laufzeit verwaltet werden.

Um Objekte in unterschiedlichen Situationen darzustellen, werden zwei Ebenen an Ansichten definiert. Die untere Ebene an Ansichten (Hauptansichten) geben an, wie die Objekte in eigenständigen Kontexten dargestellt werden. Die vier Hauptansichten sind in Tabelle 8 beschrieben. Neben den Hauptansichten wer-

Name	Zweck
<i>ObjectRendering</i>	Gibt an, wie ein Objekt dargestellt werden soll, wenn der Benutzer sich genau dieses Objekt ansieht
<i>CollectionRendering</i>	Gibt an, wie eine Liste an Objekten dargestellt werden soll
<i>SelectRendering</i>	Gibt an, wie ein Objekt dargestellt wird, wenn es zur Auswahl bereit gestellt wird
<i>EditRendering</i>	Gibt an, wie ein Objekt dargestellt wird, wenn es bearbeitet wird

Tabelle 8 Verwendete Hauptansichten

den die 7 in Tabelle 9 beschriebenen Unteransichten verwendet, um festzulegen, mit welchen Eigenschaften die verschiedenen Objekte dargestellt werden sollen. Um trotz der generischen Oberfläche eine Entwicklung von speziellen Oberflächenelementen zu ermöglichen, verfügen die Ansichten über die Möglichkeit eine Komponente anzugeben, die sie abweichend vom Standard darstellt. Die ObjectRendering-Ansicht stellt beispielsweise Objekte standardmäßig als Seite mit Reitern dar, in denen sich zum einen die Details des Objekts und zum anderen die Relationen des Objekts befinden. Welche Eigenschaften des Objekts angezeigt werden wird in der Details-Ansicht, die angezeigten Relationen in der Relation-Ansicht beschrieben.

Name	Zweck
<i>New</i>	Gibt an, welche Eigenschaften eines Objekts in der Eingabemaske zum Erstellen des Objekts angezeigt werden sollen
<i>Edit</i>	Gibt an, welche Eigenschaften eines Objekts in der Eingabemaske zum Bearbeiten des Objekts angezeigt werden sollen
<i>Collection</i>	Gibt an, welche Eigenschaften eines Objekts angezeigt werden sollen, wenn dieses in einer Liste dargestellt wird
<i>Details</i>	Gibt an, welche Eigenschaften eines Objekts angezeigt werden sollen, wenn die Objektseite des Objekts geöffnet ist
<i>Select</i>	Gibt an, welche Eigenschaften eines Objekts angezeigt werden sollen, wenn das Objekt zur Auswahl durch den Benutzer aufgelistet wird
<i>Relation</i>	Gibt an, welche Beziehungen zu anderen Objekten angezeigt werden sollen, wenn die Objektseite eines Objekts geöffnet ist
<i>Friendly</i>	Gibt an, wie ein Objekt dargestellt werden soll, wenn es im Kontext eines anderen Objekts angezeigt wird

Tabelle 9 Verwendete Unteransichten

Zur Veranschaulichung der Ansichten ist die Navigation von Cluu geeignet, da sie diese auch mit gesonderten Darstellungen verwendet. Die Navigationsstruktur von Cluu besteht aus zwei Ebenen: Anwendungen und Navigationsknoten. Die Anwendungen sind die oberste Ebene und beinhalten beliebig viele Navigationsknoten. Diese Navigationsknoten können unterschiedliche Funktionen haben. Sie können auf eine weitere Unterseite der Navigation, eine Liste an Objekten, ein einzelnes Objekt oder eine beliebige URL verweisen. Damit die Navigation durch die Anwendungsstrukturen übersichtlich gestaltet wird, können die verwendeten Objekte nicht mit ihren Standardansichten dargestellt werden. Für die Navigation in der Basisanwendung werden deshalb spezielle Renderer angegeben, die die relevanten Daten entsprechend darstellen. Die ObjectRenderer-Ansicht von Anwendungen gibt einen Renderer an, der nicht die Eigenschaften und Relationen der Anwendung, sondern eine Liste aller untergeordneten Navigationsknoten darstellt. Die Collection-Ansicht der Navigationsknoten gibt wiederum einen Renderer an, der die Navigationsknoten als Kacheln mit Bild, Überschrift und Beschreibung darstellt.

5.3.2 Navigation innerhalb des Systems

Um die Darstellung der Navigation ähnlich wie in Cleverle zu halten, wurde entschieden auf eine ähnliche Struktur zu setzen wie sie bereits verwendet wird. Logisch sind die einzelnen Elemente in der Navigation in Anwendungen und in Navigationsknoten aufgeteilt, für die unterschiedliche Eigenschaften relevant sind. Das Datenmodell der Anwendungen und Navigationsknoten ist in Abbildung 9 dargestellt. Anwendungen verfügen nur über Daten die für deren Verständnis interessant sind wie Titel, Beschreibung und anzuseigendes Bild. Für Navigationsknoten sind jedoch weitere Eigenschaften relevant. Als Hauptmerkmal haben Navigationsknoten eine Typenbeschreibung, welche die Art der von dem jeweiligen Knoten angezeigten Daten angibt. Navigationsknoten können in der geplanten Struktur entweder eine Auflistung der Objekte einer angegebenen Klasse, eine Objektansicht eines angegebenen Objekts oder weitere Navigationsknoten enthalten.

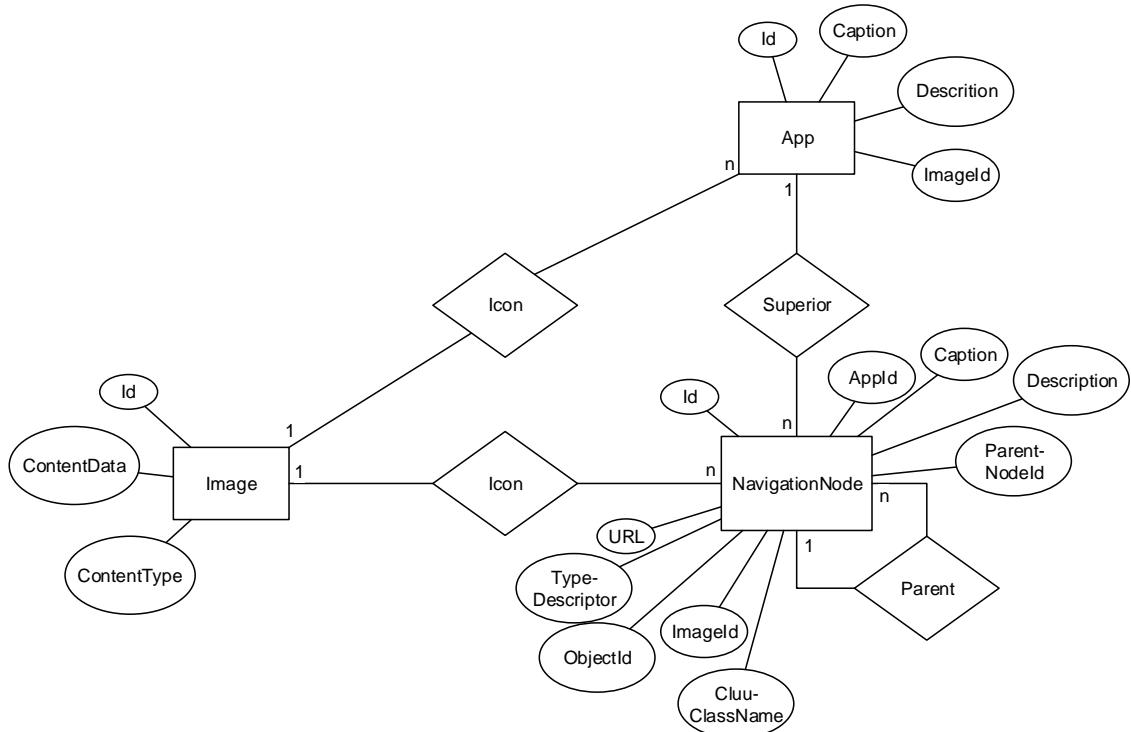


Abbildung 9 Datenmodell der Navigation mit Anwendungen und Navigationsknoten

Um auch Spezialfälle und externe Seiten anzeigen zu können, können Navigationsknoten auch eine beliebige URL enthalten auf die bei einem Öffnen des Navigationsknoten navigiert wird. Je nach Typ des Navigationsknoten müssen die benötigten Daten wie Klassename, Objekt oder URL angegeben werden. Für eine Sortierung der Navigationsknoten wird eine Ordinalzahl verwendet.

Die Hauptseite des Systems besteht aus einer einfachen Auflistung aller angelegten Anwendungen. In der Objektansicht einer speziellen Anwendung kann der Benutzer alle untergeordneten Navigationsknoten als Liste einsehen und sich weiter durch die Anwendungsstruktur bewegen. Für die Visualisierung der Navigation werden sowohl Anwendungen als auch Navigationsknoten als Kacheln dargestellt, die das angegebene Bild und die angegebene Überschrift und Beschreibung beinhalten. Somit wird eine Konsistenz bei der Darstellung von allen Navigationselementen erreicht.

5.3.3 Schnellnavigation innerhalb des Systems

Für eine schnellere und einfachere Navigation zu für Benutzer wichtige Punkte in der Anwendung, werden Favoriten verwendet. In diesen Favoriten sind ein Klassenname, ein Verweis auf das zu öffnende Objekt, sowie eine für die Sortierung relevante Ordinalzahl hinterlegt. Um die Favoriten für jeden Benutzer konfigurierbar zu gestalten wird außerdem der Benutzername des anlegenden Benutzers gespeichert. Bei einem Klick auf einen der angelegten Favoriten wird die Objektansicht des verwiesenen Objekts angezeigt. Mittels dieser Struktur kann jedes in Cluu dargestellte Objekt als Favorit angelegt werden, was auch Navigationsknoten und Anwendungen einschließt. Das verwendete Datenmodell ist in Abbildung 10 dargestellt.

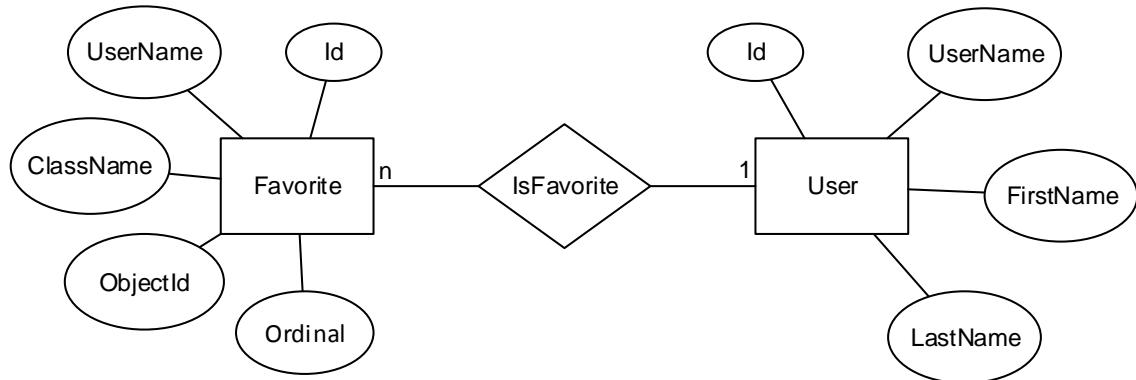


Abbildung 10 Datenmodell der Favoriten

Favoriten werden Benutzer in der Seitenleiste der Anwendung als einfache Kacheln mit Bild angezeigt. Die Verwaltung der Favoriten findet in einer Favoritenverwaltung in der Benutzer diese über Basisfunktionalitäten hinzufügen und löschen können statt.

5.3.4 Generische Aktionen

Für eine von der Oberfläche getrennte Verwaltung der Aktionen werden diese mit allen relevanten Eigenschaften als normale Klasse in den Metadaten des Systems angelegt (siehe Abbildung 11). Diese Klassen – ActionData genannt – beinhalten alle für die Ausführung der Aktion relevanten Eigenschaften. Jede Ausführung der Aktion ist somit ein Objekt der Aktionsdatenklasse und kann dementsprechend eingesehen und nachvollzogen werden. Für die Unterscheidung zwischen normalen Klassen wird jede Aktion in den Metadaten noch separat mit zusätzlichen Daten beschrieben. Diese Daten definieren zum einen die Semantik einiger Eigenschaften – solche die für Aktionsstatus, Fortschritt und Ergebnis stehen – zum anderen die Implementierung der Funktionalität und die zugehörige Aktionsdatenklasse. Eine Ausführung der Aktion besteht in dieser Struktur aus zwei Schritten: Als erstes wird ein Objekt der Aktionsdatenklasse mit allen für die Aktion relevanten Parametern angelegt.

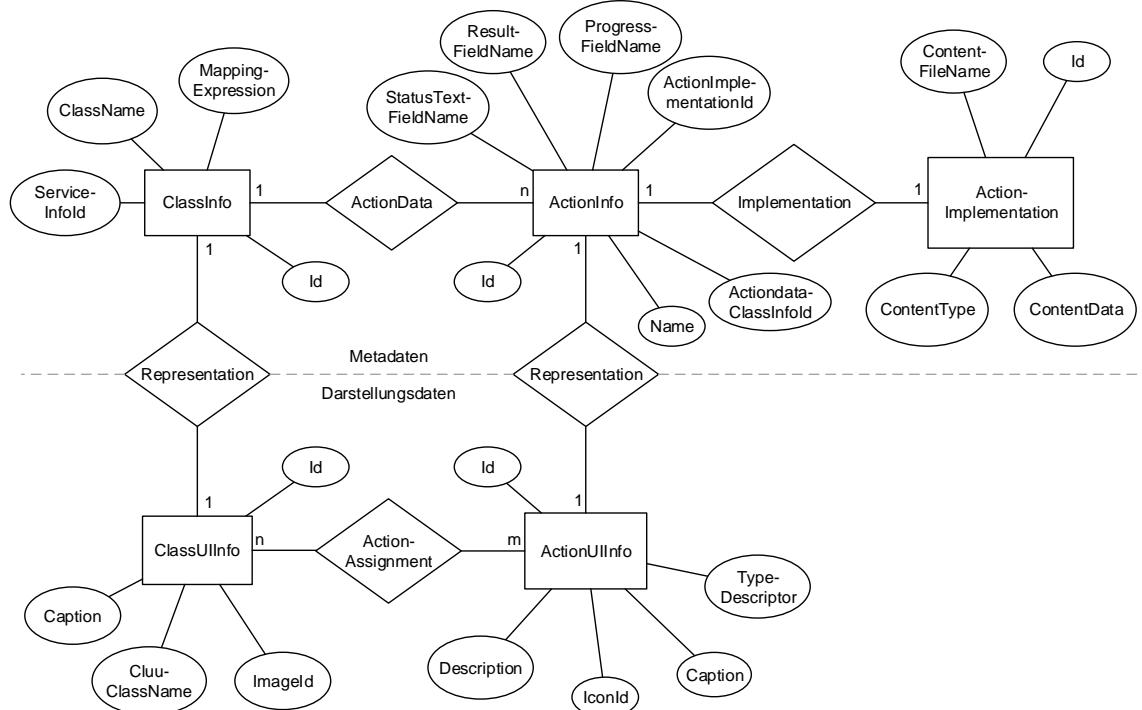


Abbildung 11 Datenmodell der Aktionen

Nachdem die Aktionsdaten erstellt wurden, wird die Aktion auf dem Cluu-Backend aufgerufen. Das Backend wurde hierfür um eine Invoke-Methode erweitert, die als Parameter ein Objekt einer Aktionsklasse, sowie ein Objekt einer Aktionsdatenklasse nimmt und anschließend die beschriebene Aktion mit den in den Aktionsdaten beschriebenen Parametern ausführt.

Auf Seiten der Oberfläche wird nur noch eine Zuweisung der Aktionen benötigt, die beschreibt im Kontext welcher Klassen die jeweilige Aktion angezeigt werden soll. Des Weiteren werden die Aktionen mit für den Benutzer relevanten Daten, beispielsweise einer Bezeichnung und einer Beschreibung, versehen.

Die Darstellung der CRUD-Operationen Hinzufügen, Bearbeiten und Löschen wird mittels Standarddialogen für alle Objekte generisch geregelt. Durch die Modellierung von erweiterten Aktionen in den Daten des Systems können Ausführungen dieser Aktionen als das Anlegen des entsprechenden Datensatzes angesehen werden. Durch diese Abstraktion können für erweiterte Aktionen die Standard-Neu-Dialoge verwendet werden, in denen alle Eigenschaften der Aktionsdaten angezeigt werden. Um die Entwicklung von speziellen Aktionseingabemasken zu unterstützen, können diese als Systemerweiterungen eingebunden und bei Vorhandensein anstelle des Standard-Neu-Dialogs angezeigt werden. Über eine am Backbone implementierte Invoke-Methode können die Aktionen nachfolgend aufgerufen werden. Hierzu erhält die Invoke-Methode als Parameter sowohl die ID der auszuführenden Aktion, als auch Klassenname und ID des Aktionsdatenobjekts. Über die Metadaten der auszuführenden Aktion kann im nächsten Schritt die Implementierung aus der Datenquelle geladen und anschließend mit den in den Aktionsdaten beschriebenen Eigenschaften ausgeführt werden.

5.3.5 Schnittstelle zur Lokalisierung von Funktionalitäten

Die TellMe genannte Schnittstelle zur Lokalisierung wird als Eingabeleiste in der Kopfleiste des Systems angebracht. Durch die ständige Darstellung der Kopfleiste in jedem Teil des Systems steht sie dem Benutzer somit zu jedem Zeitpunkt zur Verfügung. Funktional wird die Eingabeleiste durch zwei Auswahlelemente umgesetzt. In der ersten Auswahlbox kann der Benutzer die zu bearbeitende Klasse auswählen und anschließend in der zweiten Auswahlbox die zu der ausgewählten Klasse verfügbaren Aktionen auswählen und ausführen. Für die Ausführung wird dem Benutzer zuerst eine nach 5.3.4 beschriebene Eingabemaske der ausgesuchten Aktion angezeigt und diese Daten im Anschluss an das Cluu-Backend zur Verarbeitung weitergegeben.

5.4 Ergebnisse von informellen Evaluationen der ersten Iteration

Nach der erstmaligen Umsetzung der Lösungsansätze werden diese in einer ersten Bewertung von Cleverle-Nutzern getestet. Die so erhaltenen Rückmeldungen werden anschließend eingebracht und die Benutzeroberfläche entsprechend angepasst um eine bessere Benutzbarkeit zu erhalten. Im Folgenden sind die in der Bewertung bemängelten Aspekte des Systems beschrieben.

5.4.1 Navigation/Schnellnavigation und Darstellung innerhalb des Systems

Die Darstellung von einem Objekt als Block erwies sich als durchaus nachvollziehbar. Es konnte dadurch einfacher realisiert werden, dass sich in den Registern andere zu dem angezeigten Objekt gehörende Daten befinden. Allerdings stellte sich heraus, dass die Positionierung der Register am unteren Bildschirmrand für die Benutzer nicht hilfreich ist. Viele der testenden Benutzer übersahen die Register komplett und gingen fälschlicherweise davon aus, dass das angesuchte Objekt über keinerlei Relationen verfügt. Andere Benutzer beschwerten sich nach dieser Umsetzung über erhöhten Bewegungsaufwand mit der Maus, da sie diese ständig vom oberen an den unteren Bildschirmrand manövrierten mussten.

Die Navigation durch die Anwendung gestaltete sich nach der einfachen Umsetzung mit Anwendungen und Navigationsknoten fast identisch wie die des Vorgängersystems Cleverle. Um von der Ansicht eines Objekts in die Ansicht eines Objekts in einer anderen Anwendung zu wechseln muss der Benutzer weiterhin die gleichen Navigationsschritte durchführen. Als Erleichterung bei der Navigation wurde hierbei die umgesetzte Schnellnavigation angesehen, die es dem Benutzer nach einem Anlegen der relevanten Favoriten ermöglichte die umfangreiche Navigation zu überspringen und direkt zu den für ihn wichtigen Objekten zu navigieren. Jedoch waren insbesondere dieses Anlegen und auch die komplette Verwaltung der Favoriten nicht komfortabel durchzuführen, da der Benutzer zuerst in die Favoritenverwaltung navigieren musste. Nachfolgend konnte er seine persönlichen Favoriten einsehen, neue erstellen und bestehende bearbeiten und löschen. Da er sich aber für jeden Favoriten sowohl die zugehörige Klasse und Objekt-ID merken musste, war eine hohe mentale Belastung deutlich zu erkennen.

Als weiteres Problem ergab sich die identische Darstellung von Anwendungen und Navigationsknoten. Durch eine fehlende optische Unterscheidung zwischen den beiden Knotentypen konnten die testenden Benutzer nicht zu jedem Zeitpunkt sagen, an welcher Stelle des Systems sie sich befanden.

5.4.2 TellMe-Auswahl

Das TellMe-Eingabefeld als einfache Auswahl der Klasse und der zugehörigen Aktion wurde als nicht intuitiv bedienbar und zu unübersichtlich bewertet. Schon bei einem geringen, grundlegenden Funktionsumfang des Systems existieren bereits 44 Klassen mit je drei verfügbaren Aktionen. Aus dieser Menge ohne Möglichkeiten zur Einschränkung der Ergebnisse eine Aktion auszuwählen erwies sich bei internen Tests als frustrierend, da alle nicht benötigten Klassen von den Benutzern ausgeschlossen werden mussten. Eine für die weitere Entwicklung relevante Beobachtung war jedoch, dass Funktionalitäten entdeckt wurden die von den Benutzern nicht direkt gesucht wurden.

5.5 Umsetzung der Ergebnisse der ersten informellen Evaluation

Die Ergebnisse der ersten informellen Evaluation zeigen einige Probleme bei der Verwendung des Systems. Mit welchen Maßnahmen diese Probleme behoben werden, ist im Folgenden beschrieben.

5.5.1 Navigation/Schnellnavigation und Darstellung innerhalb des Systems

Die größte Änderung an der Block-Darstellung ist die Rotation der visualisierten Blöcke. Um den Bewegungsaufwand der Benutzer zu reduzieren und die Erkennungsrate zu erhöhen, werden die Registerkarten der Blöcke nun an der oberen Blockkante und die Blockringe an der linken Blockkante dargestellt. Dies soll den Block als vor dem Benutzer aufgeschlagen, beschreib- und lesbar visualisieren. Dabei wird, um die Metapher eines echten Blockes zu wahren das Verhältnis zwischen Breite und Höhe realistisch gehalten. Ist die Seite jedoch nur zum Teil gefüllt, wird der übrige Platz auf der Seite durch nicht vorhandene Ränder an der unteren Seite des Blocks visualisiert.

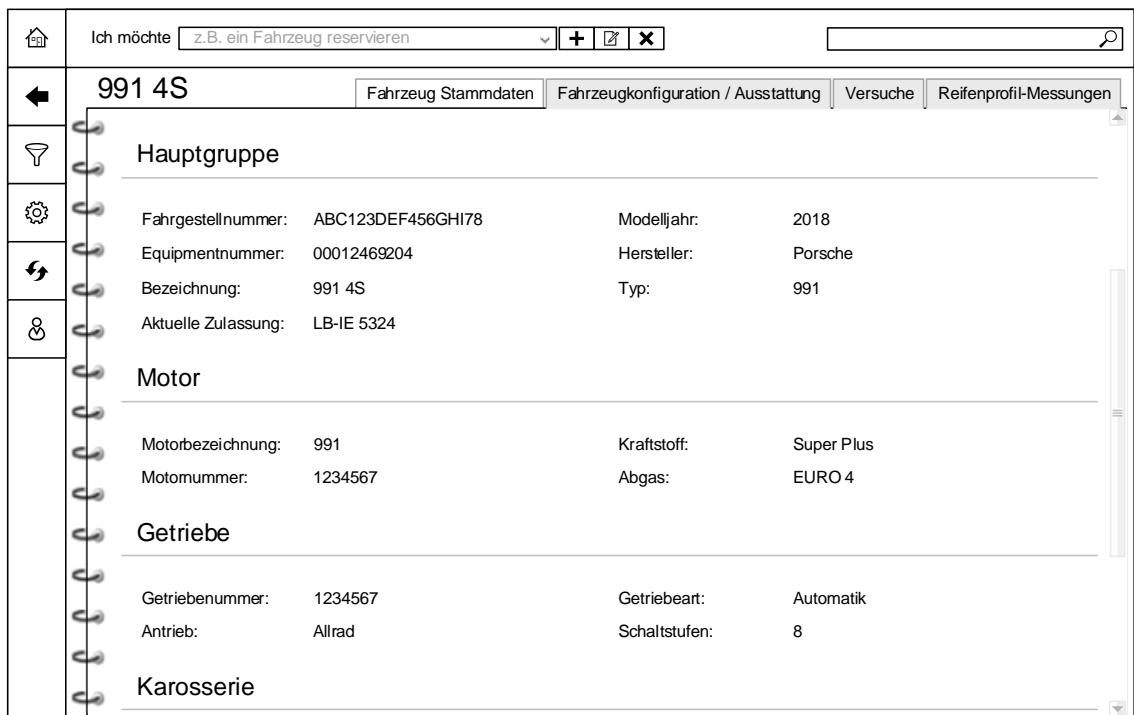


Abbildung 12 Objektansicht in Cluu als Wireframe

Um die Navigationselemente nachvollziehbar zu gestalten, wurde eine visuelle Unterscheidung zwischen Anwendungsknoten und untergeordneten Navigationsknoten in den einzelnen Anwendungen eingeführt. Diese unterschiedliche Darstellung soll es den Benutzern erleichtern, ihre aktuelle Position im System zu erkennen. Um die Block-Metapher auch in die Navigation miteinzubeziehen sollen untergeordnete Navigationsknoten die auf eine Objekt- oder Listenansicht verweisen fortlaufend als aufrecht dargestellte Blöcke visualisiert werden. Navigationsknoten die wiederrum über untergeordnete Navigationsknoten verfügen sollen als Mappe dargestellt werden (siehe Abbildung 13).

Für eine einfachere Verwendung der Favoriten wird auf die Verwaltung in einer separaten Liste verzichtet. Stattdessen wird es dem Benutzer ermöglicht, alle Objekte mittels Drag-n-Drop als Favoriten in der Seitenleiste aufzunehmen. Die Sortierung der einzelnen Favoriten wird in der Seitenleiste ebenfalls mittels Drag-n-Drop ermöglicht. Um Favoriten zu löschen, erhält der Benutzer beim Überfahren der Favoriten mit der Maus eine Schaltfläche die den jeweiligen Favoriten aus der Seitenleiste entfernt.

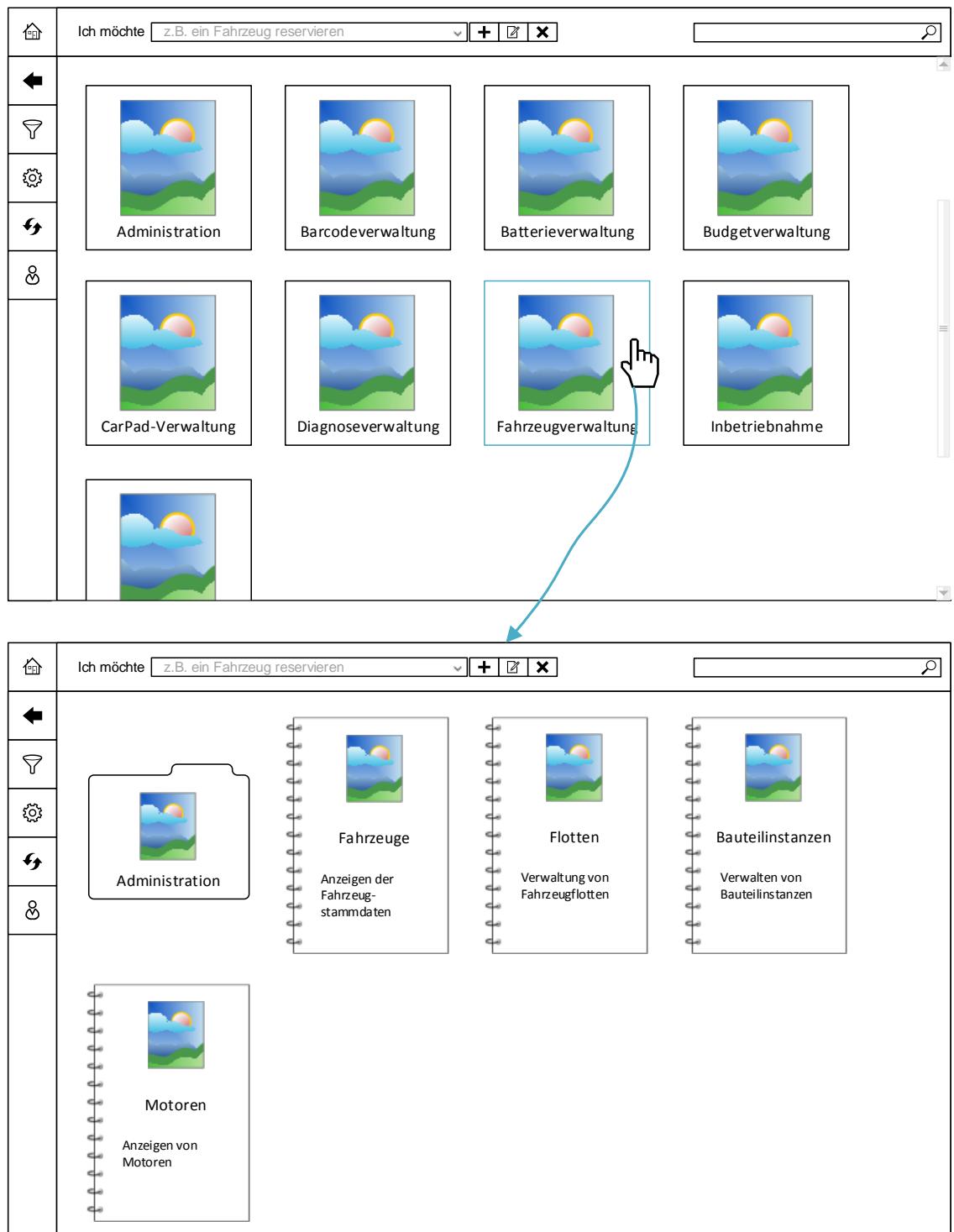


Abbildung 13 Navigationselemente in Cluu als Wireframe

5.5.2 TellMe-Eingabeleiste

Für eine übersichtlichere Gestaltung der Auswahl von Aktionen wird die Darstellung auf ein einziges Text-Eingabefeld reduziert. In diesem Feld soll der Benutzer die gesuchte Funktionalität, vorerst auf Klassenname und Aktionsname beschränkt, eingeben können. Nach der Eingabe soll das System die Eingabe

auswerten und alle relevanten Ergebnisse anzeigen. Aus diesen Ergebnissen kann der Benutzer anschließend auswählen um die gewünschte Aktion auszuführen. Ohne Eingabe eines Textes sollen dem Benutzer die am häufigsten ausgeführten Aktionen angezeigt werden.

Für die Umsetzung dieser Eingabe werden die existierenden Klassen- und Aktionsnamen nach den eingegebenen Wörtern durchsucht. Um Tippfehler berücksichtigen zu können, wird das in 2.3.1 beschriebene Verfahren verwendet um eine prozentuale Übereinstimmung der eingegebenen Wörter mit den verfügbaren Klassennamen zu erhalten. Die Treffer mit einer gewissen Übereinstimmung werden für die weitere Auswertung verwendet. Ein nachfolgender Abgleich mit den Aktionszuweisungen ergibt die Kombinationen an Klassen und Aktionen die für den eingegebenen Text verfügbar sind. Diese Kombinationen werden dem Benutzer als Dropdown-Menü unterhalb der Eingabeleiste eingeblendet und er kann mittels Tastatur oder Maus die jeweilige Aktion auswählen.

5.6 Ergebnisse von informellen Evaluationen der zweiten Iteration

Nach der Änderung einiger Eigenschaften der Oberfläche müssen diese im nächsten Schritt getestet und validiert werden. Hierzu werden wieder Bewertungen des Systems mit Cleverle-Benutzern durchgeführt. Die erkannten Probleme mit der Benutzbarkeit werden im Folgenden aufgelistet.

5.6.1 Navigation/Schnellnavigation und Darstellung innerhalb des Systems

Die hochkant dargestellten Blöcke konnten von den Testnutzern ähnlich gut in Bezug mit der Realität gesetzt werden wie die horizontale orientierten. Positiv konnte festgestellt werden, dass der Bewegungsaufwand der Maus deutlich reduziert wurde. Auch wurden die Register der Blöcke schneller und einfacher wahrgenommen als in der vorherigen Iteration. Von einigen Benutzern wurde jedoch weiterhin bemängelt, dass der Bewegungsaufwand weiterhin zu hoch sei, wenn verschiedene Tätigkeiten wie das Filtern von Daten und das Wechseln zwischen Reitern durchgeführt werden. Um die Block-Metapher jedoch nicht zu weit zu reduzieren wurde entschieden die Register an der aktuellen Position zu belassen und somit einen Kompromiss zwischen Wahrnehmung der Block-Metapher und Bewegungsaufwand einzugehen.

Durch die Unterscheidung zwischen den unterschiedlichen Navigationselementen konnte beobachtet werden, dass sich Benutzer besser im System zurecht finden konnten. Speziell konnten sie einfacher unterscheiden auf welcher Ebene der Navigation sie sich zu einem gewissen Zeitpunkt befanden und auf welche Art von Seite ein dargestelltes Objekt verweist. Die Darstellung der Navigationselemente konnte somit wie umgesetzt belassen werden.

Die Schnellnavigation mittels Favoriten war durch die Umstellung auf eine Verwaltung direkt in der Seitenleiste besser bedienbar. Benutzer kamen mit der Erstellung, dem Sortieren und Löschen besser zurecht als dies mit einer separaten Favoritenverwaltung der Fall war. Jedoch wurde durch die einfachere Verwaltung der Favoriten deutlich, dass eine Platzierung dieser direkt in der Seitenleiste zu einer schnellen Überladung der dort dargestellten Informationen führte. Da in der Seitenleiste auch andere Elemente wie die Filterfunktion untergebracht waren, konnten diese nicht mehr ausreichend einfach lokalisiert und verwendet werden.

5.6.2 TellMe-Eingabeleiste

Die Verwendung der TellMe-Eingabeleiste gestaltete sich in der zweiten Iteration als deutlich übersichtlicher und einfacherer. Ein Mangel der bei der Benutzung festgestellt wurde war jedoch, dass das Aufrufen von Funktionalitäten in einem gegebenen Kontext zu aufwändig war. Will ein Benutzer eine Aktion auf dem derzeitig angezeigten Objekt ausführen, werden ihm diese nicht sofort angezeigt, sondern er sieht die global am häufigsten ausgeführten Aktionen. Er muss zuerst den Eingabetext mit Klassennamen und Aktionsname eingeben um die angezeigten Aktionen einzuschränken. Weiterhin wurde die fehlende Flexibilität bei der Eingabe von Texten bemängelt. Durch die Beschränkung auf den Klassennamen, beziehungsweise die benutzerfreundliche Bezeichnung der Klasse war die Eingabemöglichkeit um auf eine bestimmte Aktion einer bestimmten Klasse zu kommen stark eingeschränkt. Der Benutzer konnte beispielsweise Aktionen der Klasse „Person“ nur über Eingabe des Namens „Person“ finden. Andere gleichbedeutende Begriffe hatten keine Treffer zur Folge.

Als weiteres Problem mit der TellMe-Eingabeleiste wurde die starre Eingabefolge genannt. Für die Benutzer war es nicht möglich Texte einzugeben die eine

andere Form als „*Klasse* *Aktion“ hatten. Allerdings ist es speziell für Anwendungsfälle in denen der Benutzer eine Aktion auf ein spezielles Objekt ausführen will nützlich, wenn er dieses direkt in der TellMe-Eingabeleiste eingeben kann.

5.7 Umsetzung der Ergebnisse der zweiten informellen Evaluation

Auf Basis der Erkenntnisse der vorherigen Bewertung können weitere Verbesserungen an der Oberfläche des Prototyps vorgenommen werden. Die umgesetzten Änderungen sind im Folgenden beschrieben.

5.7.1 Schnellnavigation innerhalb des Systems

Für eine übersichtlichere Darstellung von vielen Favoriten wird ein einziges Element in der Seitenleiste angebracht, welches bei einem Klick eine Erweiterung der Seitenleiste mit den existierenden Favoriten anzeigt.

5.7.2 TellMe-Eingabeleiste

Um das Aufrufen der kontextbezogenen Aktionen zu vereinfachen werden die angezeigten Ergebnisse künftig in kontextbezogene und globale Aktionen unterteilt. In der kontextbezogenen Liste werden alle Aktionen angezeigt die für das derzeitig angezeigte Objekt verfügbar sind. Bei einem Aufruf der jeweiligen kontextbezogenen Aktion wird das Objekt standardmäßig in der Ansicht der Aktion eingetragen. In der globalen Liste werden zusätzliche Aktionen aufgelistet die nicht zu dem derzeitigen Kontext gehören. Bei deren Ausführung werden keine kontextbezogenen Daten eingetragen. Ohne eine vorhandene Eingabe sollen dem Benutzer alle im Kontext vorhandenen Aktionen, sowie die am häufigsten ausgeführten Aktionen angezeigt werden.

Für erweiterte Eingabemöglichkeiten wird die Eingabefolge „*Klasse* *Aktion“ verworfen. Stattdessen wird ein Part-Of-Speech-Tagger verwendet, der den eingegebenen Wörtern ihre zugehörigen Wortarten zuweist. Aus diesen Wortarten kann nachfolgend ein Schluss gezogen werden, welche Klasse und Aktion der Benutzer aufrufen möchte. Hierbei ist anzumerken, dass die Umsetzung eines kompletten Part-Of-Speech-Tagger den Umfang dieser Diplomarbeit um weites übersteigen würde. Da aus Kostengründen ein bereits existierender Part-Of-Speech-Tagger für die spätere Verwendung nicht in Frage kommt, wird

für die TellMe-Eingabeleiste eine vereinfachte Version umgesetzt. Diese Umsetzung basiert auf den vorhandenen Klassen- und Aktionsnamen sowie Synonymen. Diese Daten können als ein bereits getaggter Korpus betrachtet werden, da jedem Synonym, beziehungsweise Klassen- und Aktionsname seine Wortart zugewiesen werden kann. Namen und Synonyme die Klassen beschreiben, werden hierbei als Nomen betrachtet, wohingegen solche die eine Aktion beschreiben als Verb eingestuft werden. Im Satz „Ich möchte ein Fahrzeug reservieren“ würde dementsprechend „Fahrzeug“ als Nomen markiert und „reservieren“ als Verb. Um Eingaben mit Bezug auf ein spezielles Objekt zu berücksichtigen, werden Schlüsselwörter wie „für“, „zu“, „von“ und weitere als Präposition markiert. Der zwischen einer Präposition und einem Verb stehende Text wird anschließend als Einschränkung der Aktion interpretiert. Die für die Auswertung der Texte verwendete Komponente ist in Abbildung 14 als UML-Klassendiagramm dargestellt.

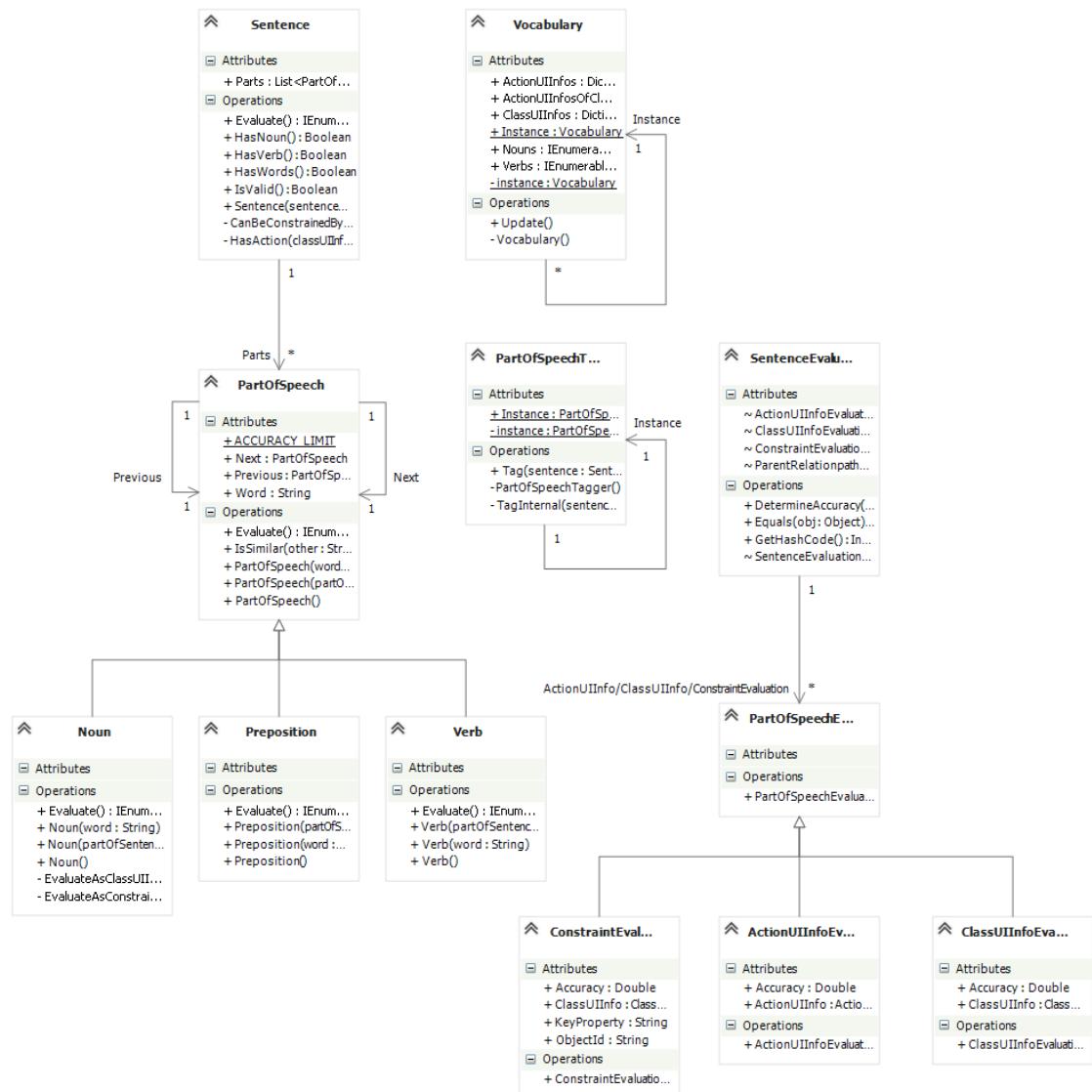


Abbildung 14 Klassenmodell der TellMe-Textevaluierung

Die Eingabe der Begriffe für Klassen und Aktionen wird um erstellbare Synonyme für diese erweitert. Diese Synonyme können für jede Klasse oder Aktion angelegt werden und werden bei der Zuweisung der Wortarten und Auswertung der Semantik berücksichtigt.

5.8 Ergebnisse der abschließenden Evaluation

Für eine Einstufung der Tauglichkeit der umgesetzten Maßnahmen zur Verbesserung der Benutzbarkeit des Systems werden abschließend formale Evaluationen mit den in 5.2 beschriebenen Testnutzern durchgeführt. Die Ergebnisse dieser Evaluation geben Aufschluss über die Güte der Benutzbarkeit des Systems. Für einen Vergleich zum Vorgängersystem werden weitere Cleverle-Benutzer nach ihrem Empfinden von Cleverle befragt. In diesem Kapitel werden die erhaltenen Ergebnisse analysiert und zusammengefasst.

5.8.1 Allgemeine Ergebnisse

Durch die Evaluationen hinweg gab es einige Dinge die bei mehreren Testnutzern zu Verwirrung führten.

Die Darstellung von m-n-Relationen ist für den Benutzer nicht ausreichend verständlich dargestellt. Dies ist im konkreten Fall der ersten Testaufgabe deutlich zu erkennen, da mehrere Testnutzer die Mitarbeiter-Relation der inkorrekt erfassten Person bearbeiten wollten um den Namen entsprechend zu ändern. Diese Änderung muss allerdings auf dem Objekt der eigentlichen Person durchgeführt werden, was für die Testnutzer nicht gleich erkennbar war.

In einigen Fällen ergibt die Darstellung der TellMe-Aktionsresultate keinen für den Benutzer ersichtlichen Sinn. Im Kontext einer Abteilung erhält der Benutzer beispielsweise das Resultat „Abteilung Personalliste“ (siehe Abbildung 15), was keine korrekte Formulierung der eigentlichen Funktionalität darstellt. Dies führte in der ersten Testaufgabe dazu, dass einige Testnutzer die verfügbare Aktion auf der Ansichtsseite der Abteilung übersahen und somit die Aufgabe nur mit einem gewissen Zeitverlust lösen konnten.

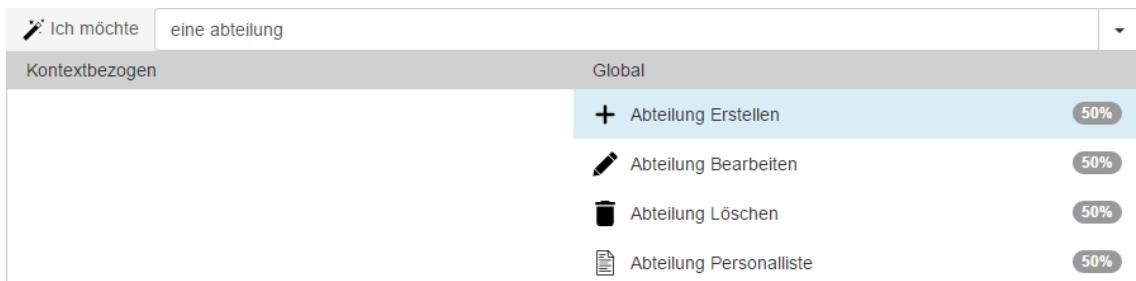


Abbildung 15 TellMe-Popup für Eingabetext "eine Abteilung"

Als ein großes Problem bei der testweisen Benutzung des Prototyps konnte die Verwendung von modalen Popups für Eingabemasken erkannt werden. Diese Popups überlagern die komplette Anwendung und verhindern jegliche Interaktion mit ihr. Speziell bei Eingabemasken welche die Auswahl eines Objekts benötigen kann dies hinderlich sein. Existiert das gewünschte Objekt in diesem Fall nicht, so muss es vom Benutzer zuerst angelegt werden. Um dies zu tun, muss der Benutzer entweder die Anwendung in einem anderen Fenster neu öffnen oder das Popup schließen. Vor allem durch letzteres wird der Benutzer aus seiner aktuellen Tätigkeit gerissen und muss sich folglich neu orientieren um diese nach der Erstellung der benötigten Objekte forzusetzen. Dieses Problem trat hauptsächlich in der dritten zu erledigenden Aufgabe auf, da für deren Ausführung die kulturneutralen Texte, sowie Bilder für jedes Navigationselement erstellt werden mussten. Die Testnutzer benötigten in diesen Fällen erheblich mehr Zeit um die Aufgabe zu lösen, als wenn sie die Texte und Bilder direkt während der Erstellung hätten anlegen können.

Ein fehlendes Feature der TellMe-Eingabeleiste wurde mit der Navigation zu Objekten erkannt. Alle Testnutzer versuchten während der Ausführung der Tests ein Objekt mittels einer Eingabe der Art „Abteilung TST4 ansehen“ zu öffnen. Diese Möglichkeit existiert im zum Zeitpunkt der Evaluationen verwendeten Stand des Systems jedoch noch nicht. Durch diese Eingabe hätten die Testnutzer bei der Ausführung ihrer Tätigkeiten einen Teil der aufgewendeten Zeit sparen können, auch ohne sich mit der Navigationsstruktur der einzelnen Anwendungen auseinanderzusetzen.

Von mehreren Testnutzern wurde auch die Funktionalität der Filter bemängelt. Da diese nicht im Rahmen dieser Arbeit entworfen wurden, verfügen sie nur über eine Grundfunktionalität, die es erlaubt einzelne Eigenschaften der Objekte einzuschränken. Die Filter befinden sich in einem über die Seitenleiste erreichbaren

Menü und beziehen sich jeweils auf die in den Listenansichten angezeigten Daten. Da die Filter visuell von den dargestellten Daten getrennt sind, geht der Bezug zu diesen verloren, weshalb einige Benutzer Probleme damit hatten die Einschränkung der Eigenschaften in den richtigen Feldern einzugeben. Die in den Eingabefeldern angezeigte Bezeichnung der zugehörigen Spalte ist für eine Zuordnung nicht ausreichend, da sie für einige Testnutzer nicht ersichtlich und somit nicht hilfreich war. Weiterhin ist es für die Benutzer nicht möglich die Filter zurückzusetzen, ohne dass sie alle eingetragenen Filter von Hand entfernen und die Daten erneut Filtern. Durch ein erneutes Laden der Seite konnten einige der Testnutzer dieses Problem umgehen, was jedoch zu einem Verlust des aktuellen Kontextes – beispielsweise des zu diesem Zeitpunkt geöffneten Reiters – führte.

5.8.2 Erlernbarkeit der Anwendung

Nach der Durcharbeitung der Testaufgaben stuften alle Testnutzer ohne Cleverle-Erfahrungen die Erlernbarkeit von Cluu als sehr einfach ein. Dies zeigt, dass auch unerfahrene Benutzer ohne größeren Lernaufwand gut mit der Anwendung umgehen können. Auch die für die Ausführung der Aufgaben benötigte Zeit und die dabei getätigten Fehler bestätigen diese Eindrücke der Testnutzer. Diese Gruppe erledigte Aufgaben 1 und 2 in durchschnittlich 8:55, respektive 6:40 Minuten und war somit ähnlich schnell oder nur unwesentlich langsamer als die Gruppe mit Cleverle-Erfahrung. Mit durchschnittlich 0 Fehlern für die erste und 0,33 Fehlern für die zweite Aufgabe waren die Anzahlen der getätigten Fehler ebenfalls gering.

5.8.3 Belastung der Anwendung nach NASA-TLX

In den Formularen zum NASA-Task-Load-Index (2.2.9) ist zu erkennen, dass die Belastung bei der Verwendung von Cluu mittelmäßig hoch ist, was bei einer erstmaligen Verwendung ohne vorherige Erfahrungen mit ähnlichen Systemen durchaus nachvollziehbar ist. Nach der ersten Aufgabe war die Durchführung der zweiten Ausgabe im Vergleich weniger belastend, wobei gleichzeitig der gefühlte Erfolg anstieg und die Frustration mit dem System sank. Die Durchschnittswerte der ausgefüllten NASA-TLX-Formulare – Aufgabe 1 in Rot, Aufgabe 2 in Blau – sind in Abbildung 16 zu sehen.

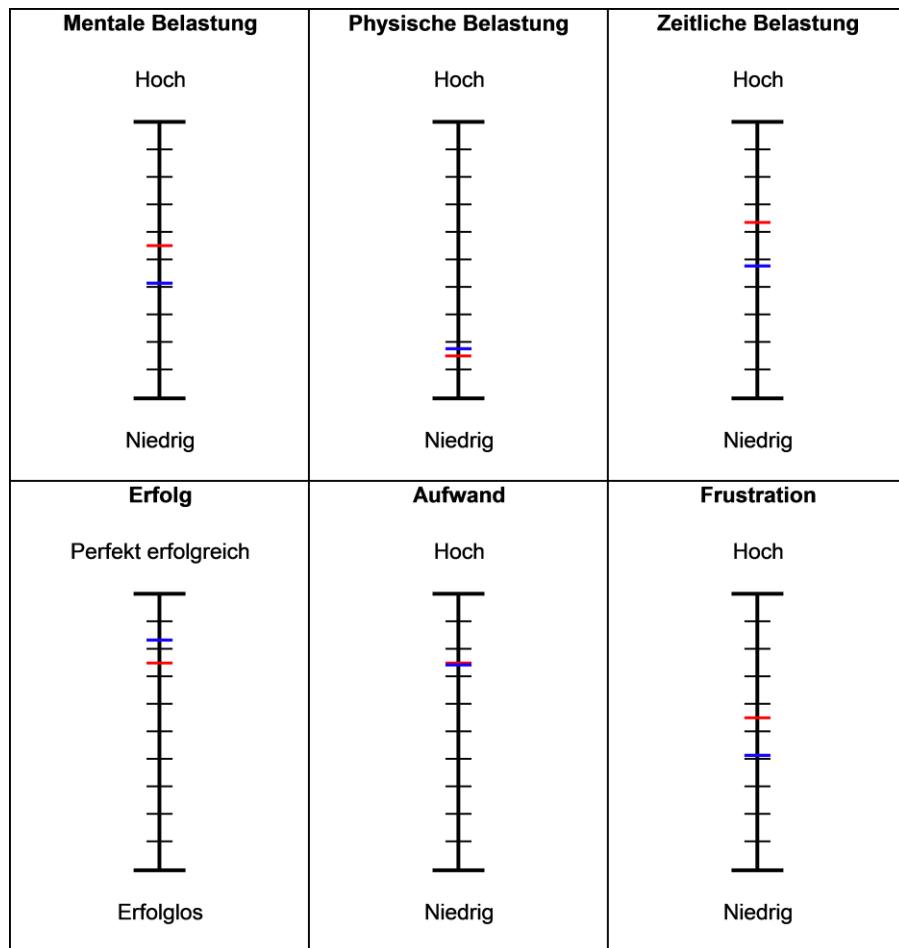


Abbildung 16 NASA-TLX von Benutzern ohne vorherige Erfahrungen mit Cleverle

5.8.4 Eindruck des getesteten Systems

Insgesamt wurde Cluu von den Testnutzern durchweg positiv bewertet. Im Vergleich zum Vorgängersystem Cleverle schnitt es in einigen Punkten deutlich besser ab. Cleverle wurde von Anwendern durchschnittlich mit 0.98 Punkten auf einer Skala von -3 bis 3 bewertet, wobei die Bewertung von benutzbarkeitsrelevanten Kriterien bei 0.97 Punkten lag. Cluu wurde insgesamt im Vergleich zu Cleverle mit 1.30 Punkten um 0.32 Punkte besser bewertet. Die benutzbarkeitsrelevanten Kriterien wurden in Cluu um 0.37 Punkte besser mit 1.34 Punkten bewertet. Ein Vergleich der Bewertungen der beiden Systeme ist in Abbildung 17 zu sehen. Im Punkt Voraussagbarkeit schnitt Cluu im Vergleich zu Cleverle deutlich schlechter ab, was durch die Verwendung eines sich noch in der Entwicklung befindlichen Prototypen erklärt werden kann. Da noch nicht alle Funktionalität umgesetzt und ausführlich getestet ist, kann es zu Fehlern und unerwarteten Rückmeldungen während der Benutzung kommen.

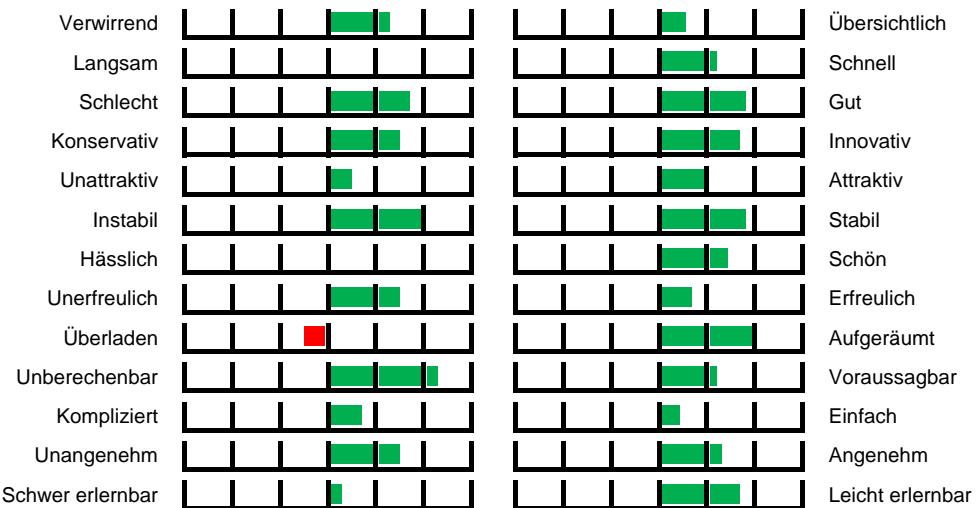


Abbildung 17 Bewertung von Cleverle (links) und Cluu (rechts) aller Testnutzer

Testnutzer ohne Cleverle-Erfahrung stuften Cluu wie in Abbildung 18 dargestellt ein. Ohne Beeinflussung von vorherigen Systemen ist eine Tendenz zu einer positiven Bewertung ersichtlich. Durchschnittlich wurde das System auf der verwendeten Skala von -3 bis 3 mit 1,25 Punkten bewertet. Hierbei wurden speziell benutzbarkeitsrelevante Kriterien mit durchschnittlich 1,33 Punkten gut bewertet. Am stärksten zeichneten sich hierbei die gute Erlernbarkeit des Systems, die aufgeräumte Oberfläche, sowie die hohe Stabilität aus. Eher negativ wurden Dinge wie die mittelmäßige Geschwindigkeit und Komplexität des Systems bewertet.

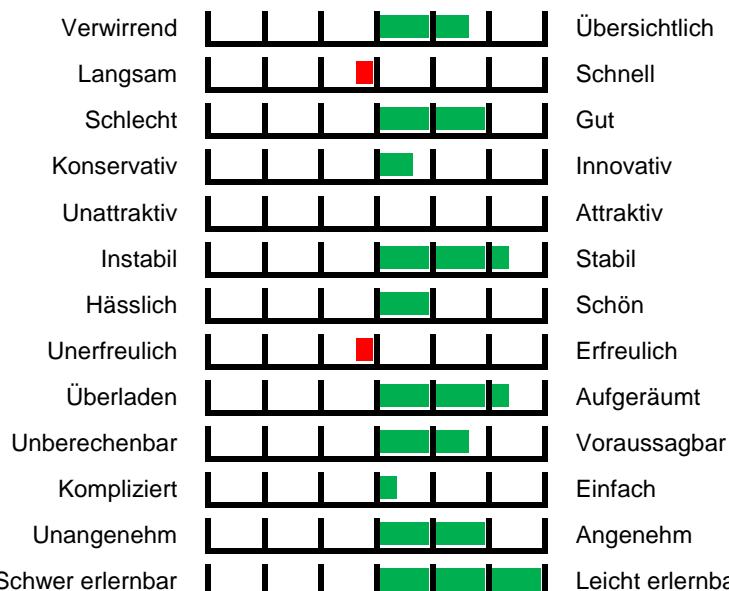


Abbildung 18 Bewertung von Cluu von Benutzern ohne vorherige Erfahrungen mit Cleverle

5.8.5 Ergebnisse der Evaluation mit Entwicklern

Bei den Evaluationen mit Entwicklern stellte sich heraus, dass das Anlegen von neuen Anwendungsstrukturen noch nicht ausreichend einfach zu bewerkstelligen ist. Durchschnittlich benötigten die Testnutzer 13:35 Minuten zum Ausführen der entsprechenden Aufgabe. Auffallend war vor allem die Verwirrung um das Eintragen der Überschriften und Beschreibungen der Anwendung und der Unterknoten. Aufgrund von Lokalisierungen werden diese Texte als Kulturneutrale Texte angegeben, die als Objekte der entsprechenden Klasse im System existieren. Neue Texte lassen sich somit nur über die Funktionalitäten des Systems (CRUD-Operationen) erstellen und anschließend bei der Erstellung der Navigationselemente auswählen. Eine direkte Erstellung in der Eingabemaske für neue Navigationselemente ist derzeit nicht möglich. Die Testnutzer wollten die Überschriften und Beschreibungen eintragen ohne vorher die entsprechenden kulturneutralen Texte erstellt zu haben. Sie benötigten einige Zeit um die Struktur hinter den kulturneutralen Texten zu verstehen, was für eine höhere zur Ausführung benötigte Zeit, sowie höhere mentale Belastungen und Frustration sorgte. Es ist jedoch anzumerken, dass nachdem die Benutzer ein gewisses Verständnis für die Struktur der Texteingabe erlangt hatten, die restlichen Schritte der Aufgabe deutlich schneller durchgeführt werden konnten. Die durchschnittliche Zeit für das Erstellen der Anwendung und dem damit verbundenen Anlegen der benötigten Texte lag bei 8:50 Minuten, die benötigte Zeit für das Erstellen der drei untergeordneten Knoten bei 4:45 Minuten. Bei der Ausführung dieser Aufgabe wurde auch die Bedeutung der zentralen Schnittstelle für das Auffinden der Funktionalitäten deutlich. Da die Benutzer keine Kenntnis über die Positionierung der kulturneutralen Texte innerhalb des Systems hatten, hätte eine Suche nach der gewünschten Funktionalität einen großen zeitlichen Aufwand bedeutet. Jedoch konnte dieser Aufwand durch die Verwendung der Eingabeleiste mit Eingabetexten wie „Text erstellen“ stark reduziert werden, da durch dieses Vorgehen der benötigte Text ohne einen Verlust des derzeitigen Kontexts erstellt werden konnte.

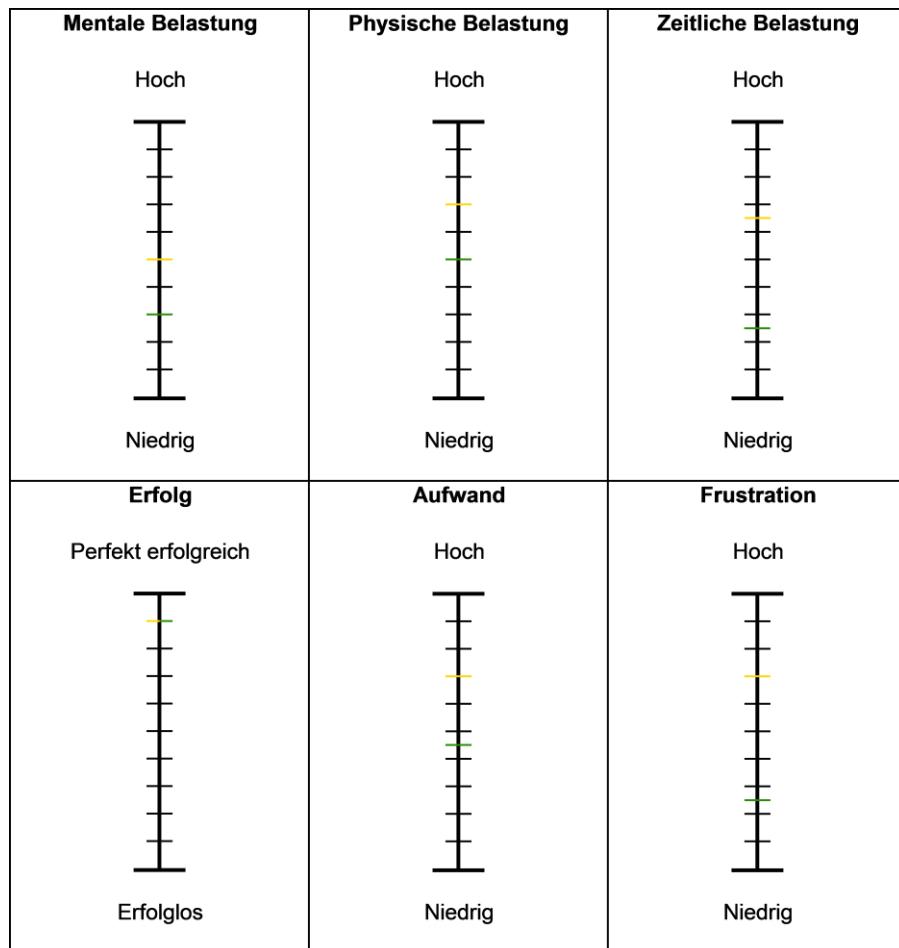


Abbildung 19 NASA-TLX von Entwicklern

Beim Austausch von Funktionalität konnte in den Tests mit Entwicklern festgestellt werden, dass die Verwendung der TellMe-Eingabeleiste einen großen Einfluss auf die benötigte Zeit für die Ausführung der Aufgabe hatte. Die Aufgabe konnte unter Zuhilfenahme der TellMe-Eingabeleiste von einem der testenden Entwickler innerhalb von 2:30 Minuten gelöst werden. Da sich die Verwaltung der Aktionsimplementierung jedoch unter den Systemmetadaten befindet, war die entsprechende Auflistung der Daten nur schwer ohne vorherige Kenntnisse zu finden. Mit dem hierfür nötigen Suchen der Funktionalität benötigte ein anderer Entwickler zur Lösung der Aufgabe insgesamt 12:30 Minuten. Dementsprechend wurde auch die empfundene Belastung der Aufgabe entweder sehr niedrig oder sehr hoch eingestuft, wodurch sich im Durchschnitt ein eher mittelhohes Belastungsniveau ergibt.

Die Durchschnittswerte der ausgefüllten NASA-TLX-Formulare – Aufgabe 3 in Grün, Aufgabe 4 in Gelb – sind in Abbildung 19 zu sehen.

6 Fazit

Wie in diesem Dokument deutlich wird, ist die Einbeziehung von Benutzern während der Entwicklung von Oberflächen eine wichtige Voraussetzung für eine gute Benutzbarkeit. Sowohl bei der Erfassung von Anforderungen, als auch bei der Evaluation der entstehenden Komponenten sind die von realen Benutzern gegebenen Rückmeldungen wertvoll und nicht zu unterschätzen. Während dieser Diplomarbeit wurden mehrere Probleme und Wünsche bezüglich der Benutzbarkeit erkannt. Diese wurden nach ihrer Wichtigkeit und ihrem Nutzen bewertet und die vielversprechendsten Lösungsansätze weiter verfolgt und umgesetzt. Es konnte eine zentrale Schnittstelle zum Auffinden der verteilten Funktionalitäten in föderierten Informationssystemen implementiert werden, die dem Benutzer erlaubt, anhand von Klassen- und Aktionsnamen beliebige Funktionalitäten an allen Stellen des Systems aufzurufen. Weiterhin konnte das Verständnis und die Navigation innerhalb des Systems durch Umsetzung einer Block-Metapher verbessert werden.

Bei der abschließenden Evaluation der entwickelten Benutzeroberfläche konnte festgestellt werden, dass diese, speziell im Vergleich zum Vorgängersystem, über eine gute, aber dennoch verbesserungsfähige Benutzbarkeit verfügt. Selbst Benutzer ohne Erfahrung mit dem Vorgängersystem konnten die gestellten Testaufgaben meist erfolgreich lösen und empfanden den benötigten Lernaufwand als gering. Jedoch ist für die weitere Entwicklung eine weiterführende Zusammenarbeit mit den Endbenutzern wichtig, um die Oberfläche auf eine gute Benutzbarkeit hin abzustimmen. Das erarbeitete Evaluationsvorgehen kann hierfür als Grundlage für zukünftig durchzuführende Evaluationen mit Testnutzern verwendet werden.

Es konnte beobachtet werden, dass eine zentrale Positionierung der Funktionalitäten den zeitlichen Aufwand bei der Verwendung eines föderierten Informationssystems reduzieren kann. Weiterhin wirkt sich diese Schnittstelle bei geeigneter Umsetzung positiv auf die Erlernbarkeit der Anwendung aus, da Benutzer sich bei einer natürlich sprachlichen Suche der gewünschten Funktionalitäten nicht mit der grundlegenden Architektur des Systems und der eingebundenen Anwendungen auskennen müssen.

Die Funktionalitäten der Schnittstelle sollten daher eingesetzt, kontinuierlich verbessert und ausgebaut werden. Vor allem die Auswertung der Eingabe befindet sich noch nicht in einem Stand der beliebige Texte erkennt. Auch können gewisse Komfortfunktionen, wie eine automatische Vervollständigung der Eingabe, mit den in dieser Diplomarbeit beschriebenen Methoden umgesetzt werden.

Da in dieser Diplomarbeit nicht alle erkannten Probleme behandelt und gelöst werden konnten, sollten während der weiteren Entwicklung die erarbeiteten Lösungsansätze überprüft, umgesetzt und evaluiert werden.

7 Anhang

Folgend sind die Ergebnisse der Evaluationen mit Testnutzern angehängt.

Vorbefragung

Name: Sebastian Roth

Datum – Uhrzeit: 23.04.15 - 16:15 - 17:05

Um die Ergebnisse dieses Tests auswerten und in Verhältnis zu anderen Tests setzen zu können, benötige ich einige Informationen zu Ihrer Person.

Alter

- 18-25 Jahre
- 26-34 Jahre
- 35-40 Jahre
- 41-47 Jahre
- älter als 47 Jahre

Geschlecht

- Weiblich
- Männlich

Ausbildung

- Hauptschulabschluss**
- Mittlere Reife**
- Fachhochschulreife**
- Hochschulreife**
- Fachhochschulabschluss**
- Hochschulabschluss**
- Doktor**
- Anderer:** _____

Computerkenntnisse

In welchem Ausmaß verwenden Sie Computer und wie stufen Sie ihre Kenntnisse bei deren Verwendung ein?

- Keine (Noch nie einen Computer benutzt)**
- Niedrig (Verwendung von einigen wenigen Anwendungen)**
- Mittel (Verwendung von mehreren Anwendungen)**
- Hoch (Verwendung von vielen Anwendungen, ohne Programmierkenntnisse)**
- Sehr hoch (Verwendung von vielen Anwendungen, Programmierkenntnisse)**
- Andere:** _____

Nutzungszeitraum „Cleverle“

Seit wann verwenden Sie das System „Cleverle“?

- Seit weniger als 3 Monaten
- Zwischen 3 und 6 Monaten
- Zwischen 6 und 12 Monaten
- Länger als 12 Monate
- Noch nie benutzt

Nutzungshäufigkeit „Cleverle“

Wie oft verwenden Sie das System „Cleverle“?

- Sehr häufig (Täglich)
- Häufig (Wöchentlich)
- Selten (Monatlich)
- Sehr selten (Jährlich)
- Nie

Nutzungsfeld in „Cleverle“

Womit beschäftigen Sie Sich bei der Verwendung von „Cleverle“ hauptsächlich?

- Einsicht in Daten
- Pflege von Daten
- Administration des Systems
- Anderes: Noch nie benutzt

Beobachtungen

Allgemein

- Stufig für Selektion von Daten
- Dateneingabe
- Überschrift von Filten nicht ersichtlich
- Filten nach Objekten - Eingabe von Eigenschaften separat - Kennzeichen von Fahrzeugen

Aufgabe 1 – Personalliste zu einer Abteilung erstellen

Durchgeführter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Öffnen des Dialogs zur Generierung der Personalliste 1 Minute	✓	✓	Will Abteilung erstellen Navigiert zu Abteilung Denkt "Mehr" wäre Fehler Findet Popup für Monat nicht Versucht auf Abteilungsnamen zu klicken zum Navigieren darauf
Änderung des Mitarbeiters "Frank Testmann" 0:45	✓	✗	
Hinzufügen von "Dora Bluhm" zur Abteilung 4 Minuten	✓	✗	Wundert sich ob Dora Bluhm noch nicht da ist. Will Personalliste bearbeiten Erstellt Personalliste neu Kein Fehler bei Personen für Abteilungen Fügt Navigiert dort hin und fügt zu Abteilung hinzu
Erneute Erstellung des Berichts 0:30	✓	✓	

Aufgabe 2 – Fahrzeug reservieren

Durchgeführter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Ansicht des Fahrzeugs um Reservierungen zu sehen 1:30 min	✓	✗	Geht zu Reservierungen und filtert nach Fahrzeugen Gibt Kennzeichen im Reservierender ein "Wieso wird heutiges Datum nicht hervorgehoben?"
Erstellen einer Reservierung 3 min	✓	✓	Reservierung will Reservierung erstellen Sucht nach Stelle zum Erstellen Wusste nicht ob entsprechende Tag Arbeitstag ist Eingabe springt nicht zu dem nächsten Feld bei Datums-eingabe

Aufgabe: 1

Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

<p>Mentale Belastung Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Physische Belastung Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Zeitliche Belastung Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> <p>Hoch</p>  <p>Niedrig</p>
<p>Erfolg Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> <p>Perfekt erfolgreich</p>  <p>Erfolglos</p>	<p>Aufwand Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Frustration Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> <p>Hoch</p>  <p>Niedrig</p>

Aufgabe:

2

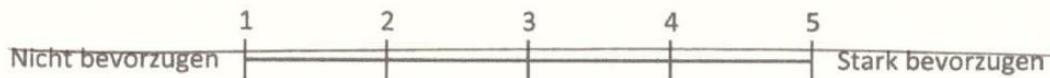
Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

<p>Mentale Belastung</p> <p>Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> 	<p>Physische Belastung</p> <p>Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> 	<p>Zeitliche Belastung</p> <p>Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> 
<p>Erfolg</p> <p>Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> 	<p>Aufwand</p> <p>Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> 	<p>Frustration</p> <p>Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> 

Abschließende Bewertung

Vielen Dank, dass Sie sich für diese Evaluation Zeit genommen haben. Zum Abschluss werden Ihnen einige Fragen gestellt, die für die Bewertung der getesteten Anwendung relevant sind. Bitte bewerten Sie die Anwendung nach Ihrem ehrlichen Empfinden und geben Sie eventuell Erläuterungen die zum Verständnis Ihres Empfindens beitragen an.

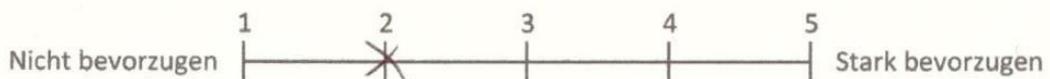
Würden Sie es bevorzugen, anstelle von Cleverle zukünftig Cluu zu verwenden?



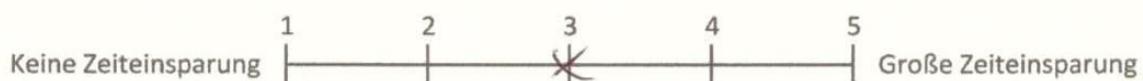
Wie wahrscheinlich ist es, dass Sie zukünftig die TellMe-Eingabeleiste verwenden werden?



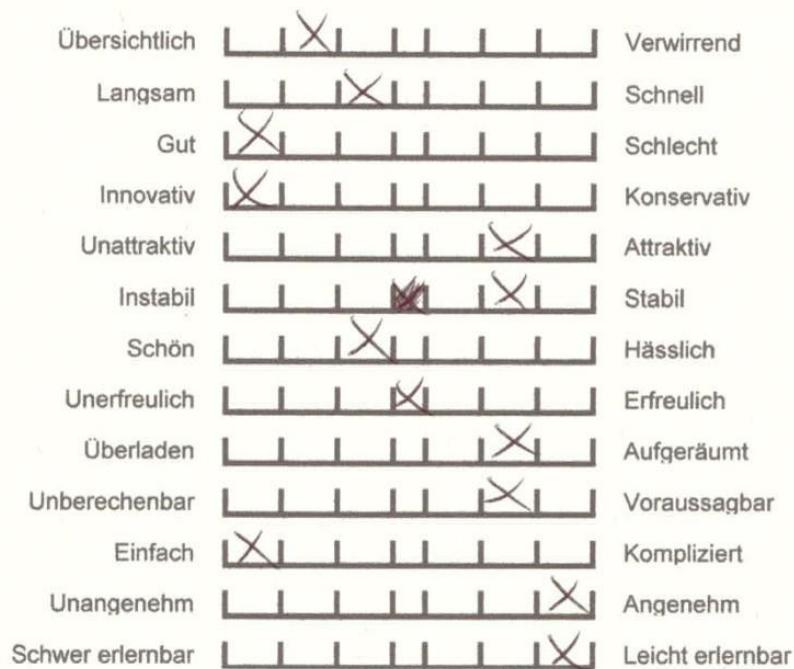
Würden Sie die TellMe-Eingabeleiste der herkömmlichen Vorgehensweise – Navigation zu den zu bearbeitenden Objekten um Aktionen auszuführen – bevorzugen?



Denken Sie, dass die TellMe-Eingabeleiste bei der alltäglichen Arbeit Zeit sparen kann?



Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.



Vorbefragung

Name:

Michael Staiger
24.04.15 - 9:50 - 10:35

Datum – Uhrzeit:

Um die Ergebnisse dieses Tests auswerten und in Verhältnis zu anderen Tests setzen zu können, benötige ich einige Informationen zu Ihrer Person.

Alter

- 18-25 Jahre
- 26-34 Jahre
- 35-40 Jahre
- 41-47 Jahre
- älter als 47 Jahre

Geschlecht

- Weiblich
- Männlich

Ausbildung

- Hauptschulabschluss
- Mittlere Reife
- Fachhochschulreife
- Hochschulreife
- Fachhochschulabschluss
- Hochschulabschluss
- Doktor
- Anderer: _____

Computerkenntnisse

In welchem Ausmaß verwenden Sie Computer und wie stufen Sie ihre Kenntnisse bei deren Verwendung ein?

- Keine (Noch nie einen Computer benutzt)
- Niedrig (Verwendung von einigen wenigen Anwendungen)
- Mittel (Verwendung von mehreren Anwendungen)
- Hoch (Verwendung von vielen Anwendungen, ohne Programmierkenntnisse)
- Sehr hoch (Verwendung von vielen Anwendungen, Programmierkenntnisse)
- Andere: _____

Nutzungszeitraum „Cleverle“

Seit wann verwenden Sie das System „Cleverle“?

- Seit weniger als 3 Monaten
- Zwischen 3 und 6 Monaten
- Zwischen 6 und 12 Monaten
- Länger als 12 Monate
- Noch nie benutzt

Nutzungshäufigkeit „Cleverle“

Wie oft verwenden Sie das System „Cleverle“?

- Sehr häufig (Täglich)
- Häufig (Wöchentlich)
- Selten (Monatlich)
- Sehr selten (Jährlich)
- Nie

Nutzungsfeld in „Cleverle“

Womit beschäftigen Sie Sich bei der Verwendung von „Cleverle“ hauptsächlich?

- Einsicht in Daten
- Pflege von Daten
- Administration des Systems
- Anderes: _____

Beobachtungen

Allgemein

Tellte öffnet Be: doppeltem Klick 2x
Speichern nicht mit Enter
Plt; Neu fehlt

Aufgabe 1 – Personalliste zu einer Abteilung erstellen

Durchgeföhrter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Öffnen des Dialogs zur Generierung der Personalliste 1:45	✓	—	Füllt Abteilungen will Abteilung erstellen Wicht bei Abteilung auf Personalliste Daten ohne Uhrzeit geht nicht
Änderung des Mitarbeiters "Frank Testmann" 1:30	✓	✗	Will Zuweisung bearbeiten Geht auf Person und bearbeitet
Hinzufügen von "Dora Bluhm" zur Abteilung 1:45	✗	✗	Will neue Person hinzufügen ↑ in Auswahl-Dialog Erstellt Person und nimmt sie auf Weist Neue Person zu
Erneute Erstellung des Berichts 0:20	✓	—	

4:00

Test-Nr.

2

Aufgabe 2 – Fahrzeug reservieren

Durchgeführter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Ansicht des Fahrzeugs um Reservierungen zu sehen	✓	✓	Gehört direkt auf Reservieren Will über TellMe Fahrzeug einsehen
Erstellen einer Reservierung	✗	✓	Will direkt Reservierung neu anlegen Lässt Reservierung überlappend an

Aufgabe 3 – Bestehende Funktionalität ändern

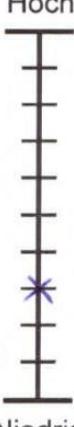
Durchgeführter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Bearbeiten der Aktionsimplementierung	✓	✓	Gehört auf Actions-TellMe Nicht auf Actionsimplementierung und bearbeitet
1:00			
Zustandsinformationen neu laden	✓	✓	Will im TellMe eingeben Gehört auf Admin... Ist im Reload Page gefangen
1:00			
Person beurlauben	✓	✓	Will Beurlaubungen ausführen
0:30			

Aufgabe 4 – Neue Anwendungsstruktur erstellen

Durchgeföhrter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Anwendung erstellen	✓	✗	<p>Will Überschrift direkt eingeben</p> <p>↑ Als Freitext</p> <p>↑ Verwirrend</p> <p>Da ich Vorschlag zu Texten zu</p> <p>Maxieren</p> <p>Legt Service an</p> <p>Legt anschließend Texte an</p>
Texte für Titel und Beschreibung anlegen	✓	✓	<p>Verwirrt wegen CultureNeutralText</p> <p>Nach Erkennung fügt er</p> <p>alle Texte/Bilder hinzu</p>
Navigationsknoten "Büroartikel"	✓	✓	Stößt auf CultureNeutralTextprobl.
Navigationsknoten "Arbeitsplätze"	✓	✓	
Navigationsknoten "Anträge"	✓	✓	

Aufgabe: 1

Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

Mentale Belastung	Physische Belastung	Zeitliche Belastung
<p>Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> <p>Hoch</p>  <p>Niedrig</p>
Erfolg	Aufwand	Frustration
<p>Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> <p>Perfekt erfolgreich</p>  <p>Erfolglos</p>	<p>Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> <p>Hoch</p>  <p>Niedrig</p>

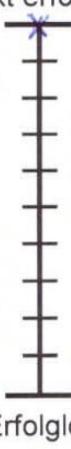
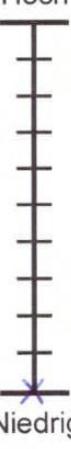
Aufgabe: 2

Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

Mentale Belastung	Physische Belastung	Zeitliche Belastung
<p>Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> <p>Hoch</p> <p>Niedrig</p>	<p>Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> <p>Hoch</p> <p>Niedrig</p>	<p>Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> <p>Hoch</p> <p>Niedrig</p>
<p>Erfolg</p> <p>Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> <p>Perfekt erfolgreich</p> <p>Erfolglos</p>	<p>Aufwand</p> <p>Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> <p>Hoch</p> <p>Niedrig</p>	<p>Frustration</p> <p>Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> <p>Hoch</p> <p>Niedrig</p>

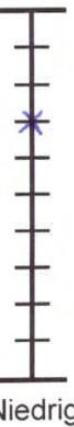
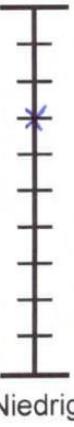
Aufgabe: 3

Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

Mentale Belastung	Physische Belastung	Zeitliche Belastung
<p>Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> <p>Hoch</p>  <p>Niedrig</p>
Erfolg	Aufwand	Frustration
<p>Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> <p>Perfekt erfolgreich</p>  <p>Erfolglos</p>	<p>Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> <p>Hoch</p>  <p>Niedrig</p>

Aufgabe: 4

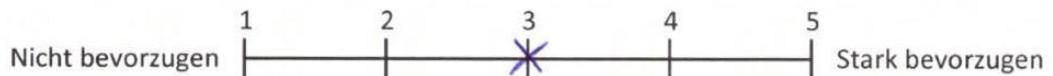
Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

<p>Mentale Belastung Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Physische Belastung Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Zeitliche Belastung Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> <p>Hoch</p>  <p>Niedrig</p>
<p>Erfolg Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> <p>Perfekt erfolgreich</p>  <p>Erfolglos</p>	<p>Aufwand Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Frustration Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> <p>Hoch</p>  <p>Niedrig</p>

Abschließende Bewertung

Vielen Dank, dass Sie sich für diese Evaluation Zeit genommen haben. Zum Abschluss werden Ihnen einige Fragen gestellt, die für die Bewertung der getesteten Anwendung relevant sind. Bitte bewerten Sie die Anwendung nach Ihrem ehrlichen Empfinden und geben Sie eventuell Erläuterungen die zum Verständnis Ihres Empfindens beitragen an.

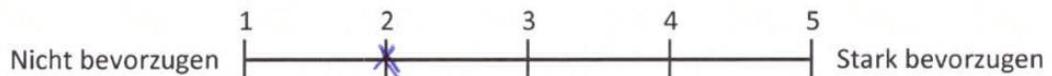
Würden Sie es bevorzugen, anstelle von Cleverle zukünftig Cluu zu verwenden?



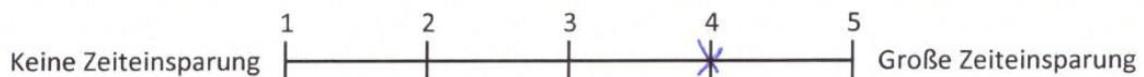
Wie wahrscheinlich ist es, dass Sie zukünftig die TellMe-Eingabeleiste verwenden werden?



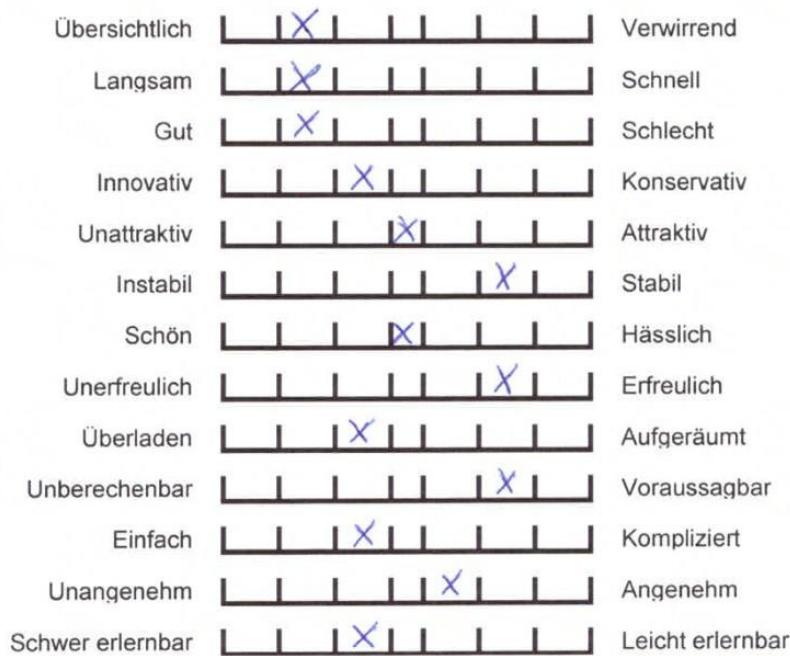
Würden Sie die TellMe-Eingabeleiste der herkömmlichen Vorgehensweise – Navigation zu den zu bearbeitenden Objekten um Aktionen auszuführen – bevorzugen?



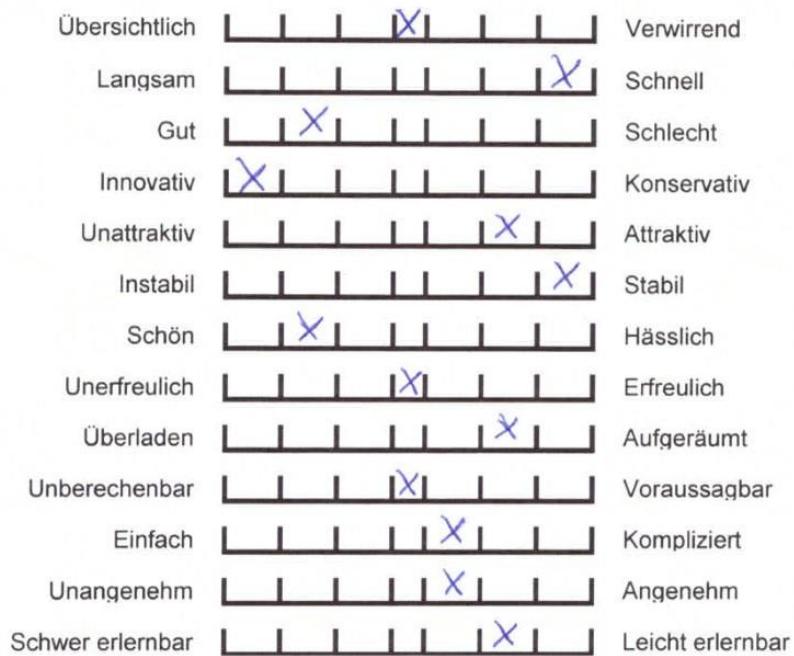
Denken Sie, dass die TellMe-Eingabeleiste bei der alltäglichen Arbeit Zeit sparen kann?



Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.



Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.



Vorbefragung

Name:

Martin Matanovic

Datum – Uhrzeit:

24.04.15 - 11:00 - 12:15

Um die Ergebnisse dieses Tests auswerten und in Verhältnis zu anderen Tests setzen zu können, benötige ich einige Informationen zu Ihrer Person.

Alter

- 18-25 Jahre
- 26-34 Jahre
- 35-40 Jahre
- 41-47 Jahre
- älter als 47 Jahre

Geschlecht

- Weiblich
- Männlich

Ausbildung

- Hauptschulabschluss
- Mittlere Reife
- Fachhochschulreife
- Hochschulreife
- Fachhochschulabschluss
- Hochschulabschluss
- Doktor
- Anderer: _____

Computerkenntnisse

In welchem Ausmaß verwenden Sie Computer und wie stufen Sie ihre Kenntnisse bei deren Verwendung ein?

- Keine (Noch nie einen Computer benutzt)
- Niedrig (Verwendung von einigen wenigen Anwendungen)
- Mittel (Verwendung von mehreren Anwendungen)
- Hoch (Verwendung von vielen Anwendungen, ohne Programmierkenntnisse)
- Sehr hoch (Verwendung von vielen Anwendungen, Programmierkenntnisse)
- Andere: _____

Nutzungszeitraum „Cleverle“

Seit wann verwenden Sie das System „Cleverle“?

- Seit weniger als 3 Monaten
- Zwischen 3 und 6 Monaten
- Zwischen 6 und 12 Monaten
- Länger als 12 Monate
- Noch nie benutzt

Nutzungshäufigkeit „Cleverle“

Wie oft verwenden Sie das System „Cleverle“?

- Sehr häufig (Täglich)
- Häufig (Wöchentlich)
- Selten (Monatlich)
- Sehr selten (Jährlich)
- Nie

Nutzungsfeld in „Cleverle“

Womit beschäftigen Sie Sich bei der Verwendung von „Cleverle“ hauptsächlich?

- Einsicht in Daten
- Pflege von Daten
- Administration des Systems
- Anderes: _____

Beobachtungen

Allgemein

- Ich möchte "eine Person aufrufen" während
- Denkt, dass nur Personen aufrufen werden können
- Icons mit Text in Schriftenleiste
- Filter zurücksetzen geht nicht
- Keine Rückmeldung bei Erstellung - Highlighting?
- Markierung in Kalender, wann F25. verfügbar
- Würde nicht von selbst ~~her~~ auf TellMe kommen
- ~~Aufgabe 1~~

Aufgabe 1 – Personalliste zu einer Abteilung erstellen

Durchgeföhrter Schritt	Richtig?	TeilMe?	Fehler/Bemerkungen
Öffnen des Dialogs zur Generierung der Personalliste 6:00	✓	✗	Navigiert zu Personenvorwaltung → Abteilungen Legt neue Abteilung an Füllt anschließend Markt, dass schon eine existiert Öffnet Kalender-Popup
Änderung des Mitarbeiters "Frank Testmann" 7:30	✓	✗	Füllt im Personen der Abt. Will Zuweisung ändern Gibt Testmann im TeilMe ein Wählt im Anschluss Person über Liste aus
Hinzufügen von "Dora Bluhm" zur Abteilung 4:00	✗	✗	Will Person neu anlegen Geht auf Person und legt neu an Weist dann zu Will einzelne Person filtern Gibt Vorname zuerst ein Fügt neu hinzugefügte Person hinzu
Erneute Erstellung des Berichts 0:30	✓	✗	Bericht ≠ Personalliste

Aufgabe 2 – Fahrzeug reservieren

Durchgeföhrter Schritt	Richtig?	TeilMe?	Fehler/Bemerkungen
Ansicht des Fahrzeugs um Reservierungen zu sehen 2:00	✓	✗	<p>Navigation zu Fahrzeugen Filtert und navigiert zu Fzg. Will Reservierung neu anlegen Navigation zu Reservierungen Übt Fahrzeug Reservieren in TeilMe Filtert nach Kennzeichen</p>
Erstellen einer Reservierung 4:15	✓	✓	<p>Erinnert sich nicht an verfügbares Datum</p>

Bewertung – Anfangs schlechter, danach besser, weil Über System gelernt

Aufgabe 3 – Bestehende Funktionalität ändern

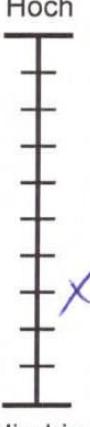
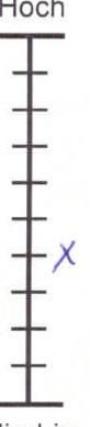
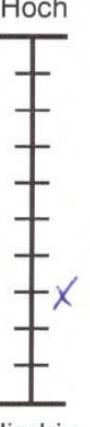
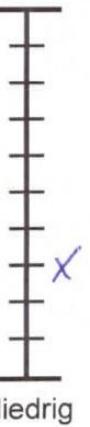
Durchgeföhrter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Bearbeiten der Aktionsimplementierung 8:30	✓	✗	<p>Geliefert Person Beurlauben-Aktion Naviert zu Personen</p> <p>Dankt Funktionalität außerhalb der Aktion, hochladen Naviert zu Bearbeiten-Aktion "Klassentafel" ⇒ Aktion Beurlauben ⇒ Bearbeiten impl. will neue Anlegen</p>
Zustandsinformationen neu laden 1:30	✓	✗	
Person beurlauben 2:30	✓	✓	<p>Schreißfehler Wählt sich als Testperson</p>

Aufgabe 4 – Neue Anwendungsstruktur erstellen

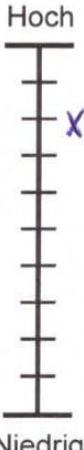
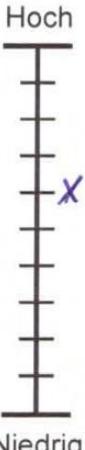
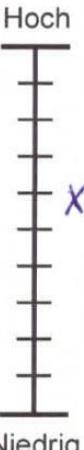
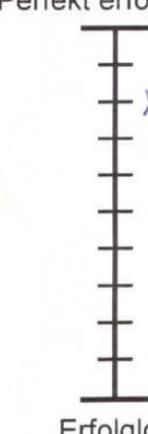
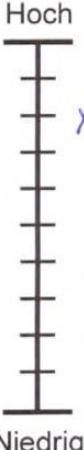
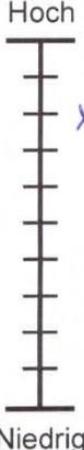
Durchgeföhrter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Anwendung erstellen 7:00	✓	✗	Gehört im Oberflächenadmin in. Neue App Öffnet Dialog zur Auswahl
Texte für Titel und Beschreibung anlegen 3:40 +1:00	✓	✓	Gibt einen "Überschrift erstellen" Danach "Text" Erstellt Texte Erstellt Bilder korrekt alle gleichzeitig
Navigationsknoten "Büroartikel" 3:00	✓	✗	Navigiert in Anwendung Knoten ohne Bezeichnung etc. Knoten bearbeiten
Navigationsknoten "Arbeitsplätze" 0:30	✓	✗	
Navigationsknoten "Anträge" 8:00	✓	✗	

Aufgabe: 1

Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

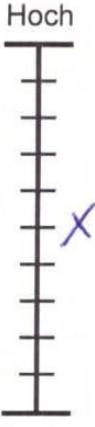
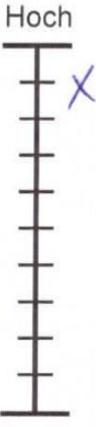
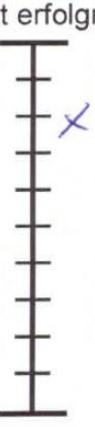
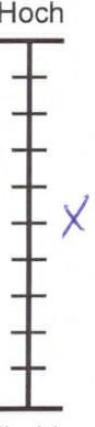
<p>Mentale Belastung Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Physische Belastung Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Zeitliche Belastung Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> <p>Hoch</p>  <p>Niedrig</p>
<p>Erfolg Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> <p>Perfekt erfolgreich</p>  <p>Erfolglos</p>	<p>Aufwand Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Frustration Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> <p>Hoch</p>  <p>Niedrig</p>

Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

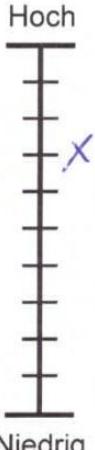
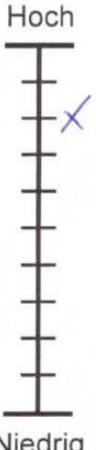
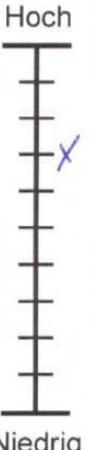
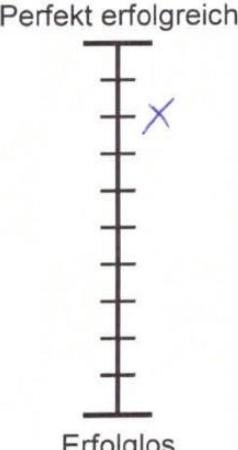
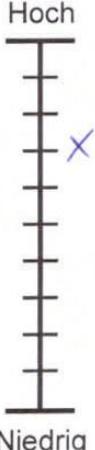
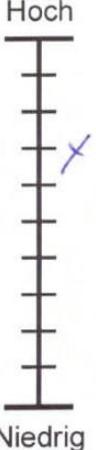
Mentale Belastung	Physische Belastung	Zeitliche Belastung
<p>Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> 	<p>Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> 	<p>Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> 
Erfolg	Aufwand	Frustration
<p>Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> 	<p>Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> 	<p>Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> 

Aufgabe: 3

Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

Mentale Belastung Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?	Physische Belastung Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?	Zeitliche Belastung Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?
Hoch  Niedrig	Hoch  Niedrig	Hoch  Niedrig
Erfolg Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?	Aufwand Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?	Frustration Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?
Perfekt erfolgreich  Erfolglos	Hoch  Niedrig	Hoch  Niedrig

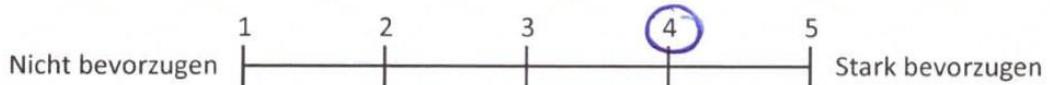
Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

Mentale Belastung	Physische Belastung	Zeitliche Belastung
<p>Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> 	<p>Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> 	<p>Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> 
Erfolg	Aufwand	Frustration
<p>Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> 	<p>Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> 	<p>Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> 

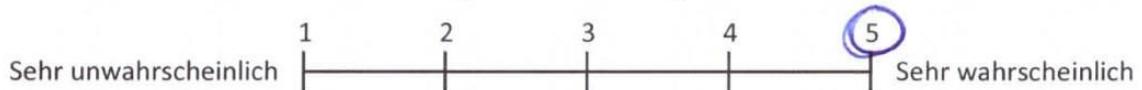
Abschließende Bewertung

Vielen Dank, dass Sie sich für diese Evaluation Zeit genommen haben. Zum Abschluss werden Ihnen einige Fragen gestellt, die für die Bewertung der getesteten Anwendung relevant sind. Bitte bewerten Sie die Anwendung nach Ihrem ehrlichen Empfinden und geben Sie eventuell Erläuterungen die zum Verständnis Ihres Empfindens beitragen an.

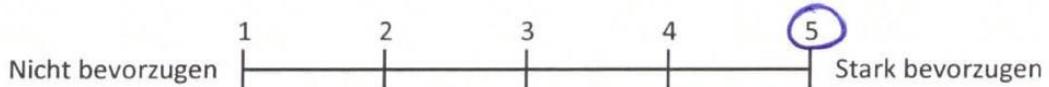
Würden Sie es bevorzugen, anstelle von Cleverle zukünftig Cluu zu verwenden?



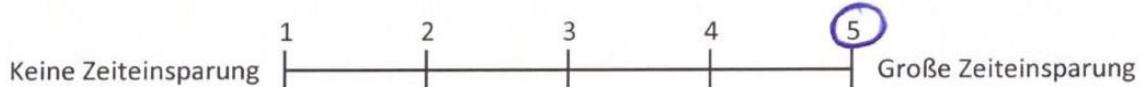
Wie wahrscheinlich ist es, dass Sie zukünftig die TellMe-Eingabeleiste verwenden werden?



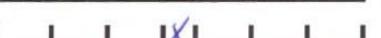
Würden Sie die TellMe-Eingabeleiste der herkömmlichen Vorgehensweise – Navigation zu den zu bearbeitenden Objekten um Aktionen auszuführen – bevorzugen?



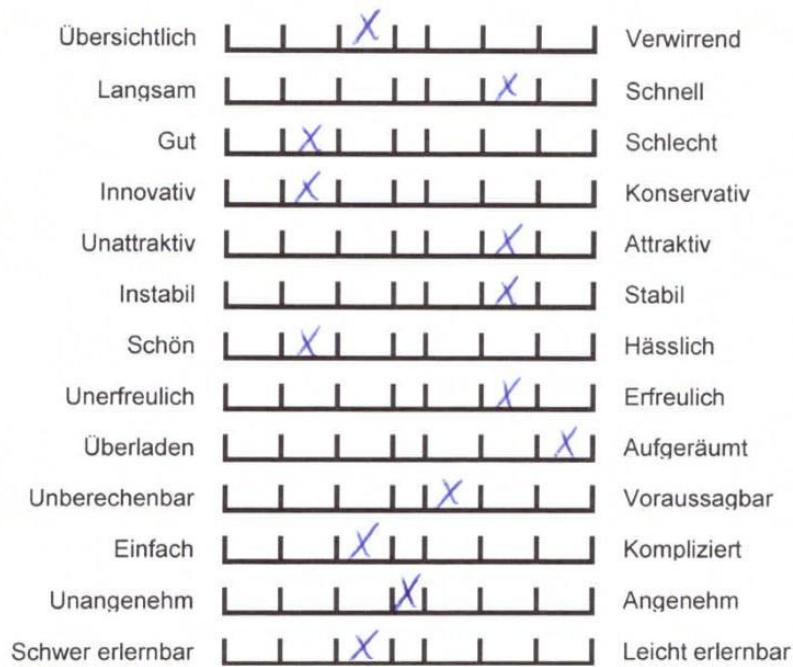
Denken Sie, dass die TellMe-Eingabeleiste bei der alltäglichen Arbeit Zeit sparen kann?



Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.

Übersichtlich		Verwirrend
Langsam		Schnell
Gut		Schlecht
Innovativ		Konservativ
Unattraktiv		Attraktiv
Instabil		Stabil
Schön		Hässlich
Unerfreulich		Erfreulich
Überladen		Aufgeräumt
Unberechenbar		Voraussagbar
Einfach		Kompliziert
Unangenehm		Angenehm
Schwer erlernbar		Leicht erlernbar

Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.



Vorbefragung

Name:

Jennifer Lieberwirth
26.04.15 - 16:00 - 16:40

Datum - Uhrzeit:

Um die Ergebnisse dieses Tests auswerten und in Verhältnis zu anderen Tests setzen zu können, benötige ich einige Informationen zu Ihrer Person.

Alter

- 18-25 Jahre
- 26-34 Jahre
- 35-40 Jahre
- 41-47 Jahre
- älter als 47 Jahre

Geschlecht

- Weiblich
- Männlich

Ausbildung

- Hauptschulabschluss
- Mittlere Reife
- Fachhochschulreife
- Hochschulreife
- Fachhochschulabschluss
- Hochschulabschluss
- Doktor
- Anderer: _____

Computerkenntnisse

In welchem Ausmaß verwenden Sie Computer und wie stufen Sie ihre Kenntnisse bei deren Verwendung ein?

- Keine (Noch nie einen Computer benutzt)
- Niedrig (Verwendung von einigen wenigen Anwendungen)
- Mittel (Verwendung von mehreren Anwendungen)
- Hoch (Verwendung von vielen Anwendungen, ohne Programmierkenntnisse)
- Sehr hoch (Verwendung von vielen Anwendungen, Programmierkenntnisse)
- Andere: _____

Nutzungszeitraum „Cleverle“

Seit wann verwenden Sie das System „Cleverle“?

- Seit weniger als 3 Monaten
- Zwischen 3 und 6 Monaten
- Zwischen 6 und 12 Monaten
- Länger als 12 Monate
- Noch nie benutzt

Nutzungshäufigkeit „Cleverle“

Wie oft verwenden Sie das System „Cleverle“?

- Sehr häufig (Täglich)
- Häufig (Wöchentlich)
- Selten (Monatlich)
- Sehr selten (Jährlich)
- Nie

Nutzungsfeld in „Cleverle“

Womit beschäftigen Sie Sich bei der Verwendung von „Cleverle“ hauptsächlich?

- Einsicht in Daten
- Pflege von Daten
- Administration des Systems
- Anderes: Noch nie benutzt

Beobachtungen

Allgemein

- Anzahl von Objekten wäre hilfreich
- Denkt zurück-Schritt verwirft Filter
- Navigationsmöglichkeit bei Auswahl

Aufgabe 1 – Personalliste zu einer Abteilung erstellen

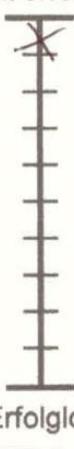
Durchgeföhrter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Öffnen des Dialogs zur Generierung der Personalliste 1:30 + 1:00	✓	✓	Wicht auf Pers-Verwalt. Gibt in TellMe ein Gibt TST 4 ein Sucht Button um weiter zu sehen Will Abt. einschränken-Filtert-Sicht ein
Änderung des Mitarbeiters "Frank Testmann" 2:00	✓	✗	Sucht Person in Liste der Personen ohne Filtern Wicht auf Link für Person
Hinzufügen von "Dora Bluhm" zur Abteilung 2:00	✓	✓	Fügt Dora Bluhm direkt hinz
Erneute Erstellung des Berichts 2:00	✓	✓	Erstellen → Exportieren

Aufgabe 2 – Fahrzeug reservieren

Durchgeführter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Ansicht des Fahrzeugs um Reservierungen zu sehen 3:00	X	-	Will bei der Reservierung den Zeitraum aussuchen Will dort Fahrzeug öffnen Geht im Fahr-Vorw. \Rightarrow Filtert Erkennt falsches Datum
Erstellen einer Reservierung 2:00 + 2:30	✓	✓	Will Kennzeichen als Fzg. Bez. einfügen Öffnet Popup für Auswahl

Aufgabe: 1

Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

Mentale Belastung Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?	Physische Belastung Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?	Zeitliche Belastung Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?
Hoch  Niedrig	Hoch  Niedrig	Hoch  Niedrig
Erfolg Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?	Aufwand Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?	Frustration Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?
Perfekt erfolgreich  Erfolglos	Hoch  Niedrig	Hoch  Niedrig

Aufgabe:

2

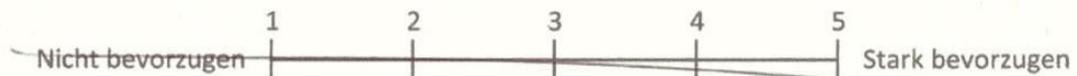
Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

<p>Mentale Belastung</p> <p>Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p>  <p>Hoch</p> <p>Niedrig</p>	<p>Physische Belastung</p> <p>Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p>  <p>Hoch</p> <p>Niedrig</p>	<p>Zeitliche Belastung</p> <p>Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p>  <p>Hoch</p> <p>Niedrig</p>
<p>Erfolg</p> <p>Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p>  <p>Perfekt erfolgreich</p> <p>Erfolglos</p>	<p>Aufwand</p> <p>Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p>  <p>Hoch</p> <p>Niedrig</p>	<p>Frustration</p> <p>Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p>  <p>Hoch</p> <p>Niedrig</p>

Abschließende Bewertung

Vielen Dank, dass Sie sich für diese Evaluation Zeit genommen haben. Zum Abschluss werden Ihnen einige Fragen gestellt, die für die Bewertung der getesteten Anwendung relevant sind. Bitte bewerten Sie die Anwendung nach Ihrem ehrlichen Empfinden und geben Sie eventuell Erläuterungen die zum Verständnis Ihres Empfindens beitragen an.

Würden Sie es bevorzugen, anstelle von Cleverle zukünftig Cluu zu verwenden?



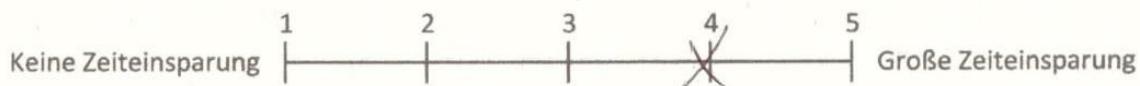
Wie wahrscheinlich ist es, dass Sie zukünftig die TellMe-Eingabeleiste verwenden werden?



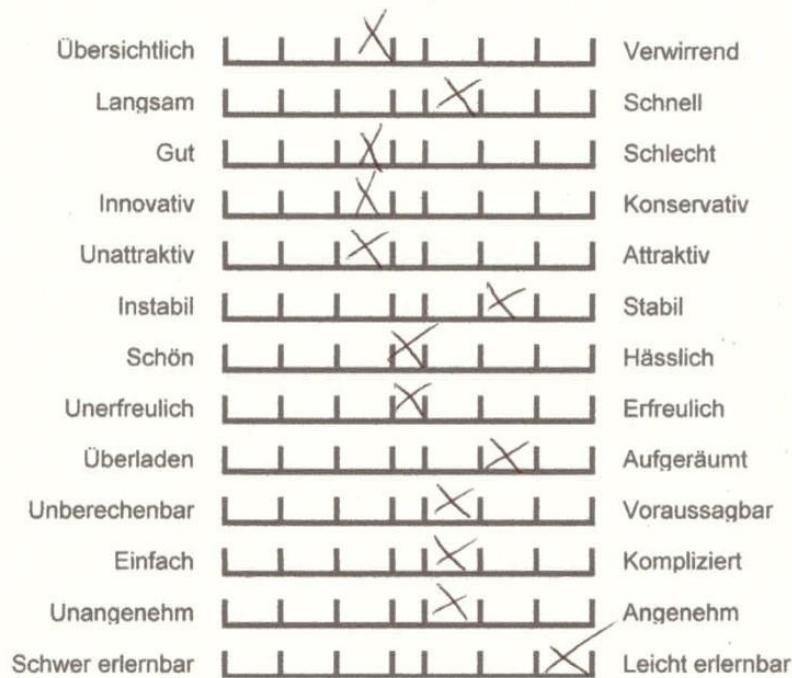
Würden Sie die TellMe-Eingabeleiste der herkömmlichen Vorgehensweise – Navigation zu den zu bearbeitenden Objekten um Aktionen auszuführen – bevorzugen?



Denken Sie, dass die TellMe-Eingabeleiste bei der alltäglichen Arbeit Zeit sparen kann?



Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.



Vorbefragung

Name:

Dennis Lieberwirth

Datum – Uhrzeit:

26.04.15 - 16:50 - 17:20

Um die Ergebnisse dieses Tests auswerten und in Verhältnis zu anderen Tests setzen zu können, benötige ich einige Informationen zu Ihrer Person.

Alter

- 18-25 Jahre
- 26-34 Jahre
- 35-40 Jahre
- 41-47 Jahre
- älter als 47 Jahre

Geschlecht

- Weiblich
- Männlich

Ausbildung

- Hauptschulabschluss
- Mittlere Reife
- Fachhochschulreife
- Hochschulreife
- Fachhochschulabschluss
- Hochschulabschluss
- Doktor
- Anderer: _____

Computerkenntnisse

In welchem Ausmaß verwenden Sie Computer und wie stufen Sie ihre Kenntnisse bei deren Verwendung ein?

- Keine (Noch nie einen Computer benutzt)
- Niedrig (Verwendung von einigen wenigen Anwendungen)
- Mittel (Verwendung von mehreren Anwendungen)
- Hoch (Verwendung von vielen Anwendungen, ohne Programmierkenntnisse)
- Sehr hoch (Verwendung von vielen Anwendungen, Programmierkenntnisse)
- Andere: _____

Nutzungszeitraum „Cleverle“

Seit wann verwenden Sie das System „Cleverle“?

- Seit weniger als 3 Monaten
- Zwischen 3 und 6 Monaten
- Zwischen 6 und 12 Monaten
- Länger als 12 Monate
- Noch nie benutzt

Nutzungshäufigkeit „Cleverle“

Wie oft verwenden Sie das System „Cleverle“?

- Sehr häufig (Täglich)
- Häufig (Wöchentlich)
- Selten (Monatlich)
- Sehr selten (Jährlich)
- Nie

Nutzungsfeld in „Cleverle“

Womit beschäftigen Sie Sich bei der Verwendung von „Cleverle“ hauptsächlich?

- Einsicht in Daten
- Pflege von Daten
- Administration des Systems
- Anderes: Noch nie benutzt

Beobachtungen

Allgemein

- Verwendet Filtern nicht wirklich
- Bestätigung bei Ausführung

12:00

Test-Nr.

5

Aufgabe 1 – Personalliste zu einer Abteilung erstellen

Durchgeföhrter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Öffnen des Dialogs zur Generierung der Personalliste 6:00	✓	✓	Navigiert zu Abteilungen und wählt Wicht auf Bezeichnung, Zeige Navigiert zu Abteilungsliste zurück Wieder zurück Navigiert zu Reiter Personalliste Gibt im TellMe ein Drückt Enter - Nichts passiert
Änderung des Mitarbeiters "Frank Testmann" 3:00	✓	✗	Will beim Filtern Vorname zuerst eingeben Will im Anzeigemasche ändern Geht zunächst auf Abteilung ⇒ Zurück zu Person
Hinzufügen von "Dora Bluhm" zur Abteilung 1:00	✓	✗	Geht korrekt vor
Erneute Erstellung des Berichts 2:00	✓	✗	Öffnet TellMe-Sicht "Abt. Pers. List" und sagt "nein" Geht zu Personallisten und erstellt neu

8:00

Test-Nr.

5

Aufgabe 2 – Fahrzeug reservieren

Durchgeführter Schritt	Richtig?	TellMe?	Fehler/Bemerkungen
Ansicht des Fahrzeugs um Reservierungen zu sehen 3:00	✓	✗	Navigator Fzg. Vw. Würde in Reserv. fehlt in Fzg. Filtiert nach Fzg. Navigator auf Fzg.
Erstellen einer Reservierung 5:00	✓	✓	Geht auf Fzg. Flotte Gibt in TellMe ein Gibt vollständigen Namen ein Sieht vorgefüllte Personen nicht

Aufgabe: 1

Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

<p>Mentale Belastung</p> <p>Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Physische Belastung</p> <p>Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Zeitliche Belastung</p> <p>Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p> <p>Hoch</p>  <p>Niedrig</p>
<p>Erfolg</p> <p>Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p> <p>Perfekt erfolgreich</p>  <p>Erfolglos</p>	<p>Aufwand</p> <p>Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p> <p>Hoch</p>  <p>Niedrig</p>	<p>Frustration</p> <p>Wie stark wurden Sie durch die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p> <p>Hoch</p>  <p>Niedrig</p>

Aufgabe:

2

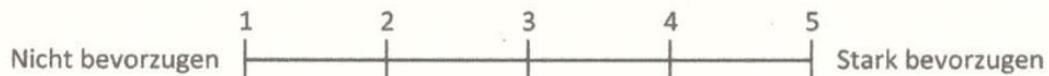
Im Folgenden sollen Sie die Belastung der von Ihnen getesteten Anwendung angeben. Markieren Sie dazu das Niveau der Belastung auf der jeweiligen Skala.

<p>Mentale Belastung</p> <p>Wie stark hat Sie die Durchführung der Aufgabe mental beansprucht?</p>  <p>Hoch</p> <p>Niedrig</p>	<p>Physische Belastung</p> <p>Wieviel körperliche Arbeit (Bewegung der Maus, Betätigen von Tasten etc.) mussten Sie zur Durchführung der Aufgabe aufbringen?</p>  <p>Hoch</p> <p>Niedrig</p>	<p>Zeitliche Belastung</p> <p>Wie war der zeitliche Druck/das zeitliche Vorranschreiten während der Durchführung der Aufgabe?</p>  <p>Hoch</p> <p>Niedrig</p>
<p>Erfolg</p> <p>Wie erfolgreich konnten Sie die Ihnen gestellte Aufgabe lösen?</p>  <p>Perfekt erfolgreich</p> <p>Erfolglos</p>	<p>Aufwand</p> <p>Wieviel Aufwand mussten Sie aufbringen um Ihr Erfolgsniveau zu erreichen?</p>  <p>Hoch</p> <p>Niedrig</p>	<p>Frustration</p> <p>Wie stark wurden Sie durch 'die Anwendung bei der Durchführung Ihrer Aufgabe frustriert?</p>  <p>Hoch</p> <p>Niedrig</p>

Abschließende Bewertung

Vielen Dank, dass Sie sich für diese Evaluation Zeit genommen haben. Zum Abschluss werden Ihnen einige Fragen gestellt, die für die Bewertung der getesteten Anwendung relevant sind. Bitte bewerten Sie die Anwendung nach Ihrem ehrlichen Empfinden und geben Sie eventuell Erläuterungen die zum Verständnis Ihres Empfindens beitragen an.

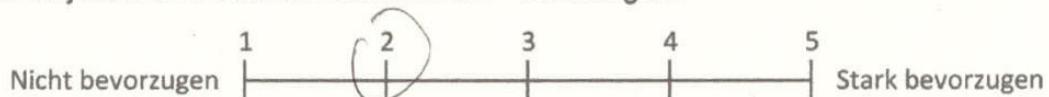
Würden Sie es bevorzugen, anstelle von Cleverle zukünftig Cluu zu verwenden?



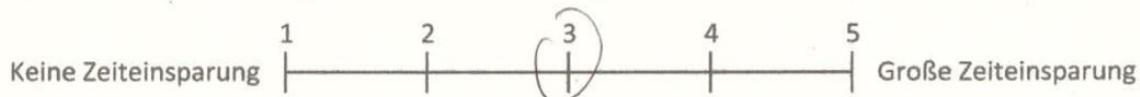
Wie wahrscheinlich ist es, dass Sie zukünftig die TellMe-Eingabeleiste verwenden werden?



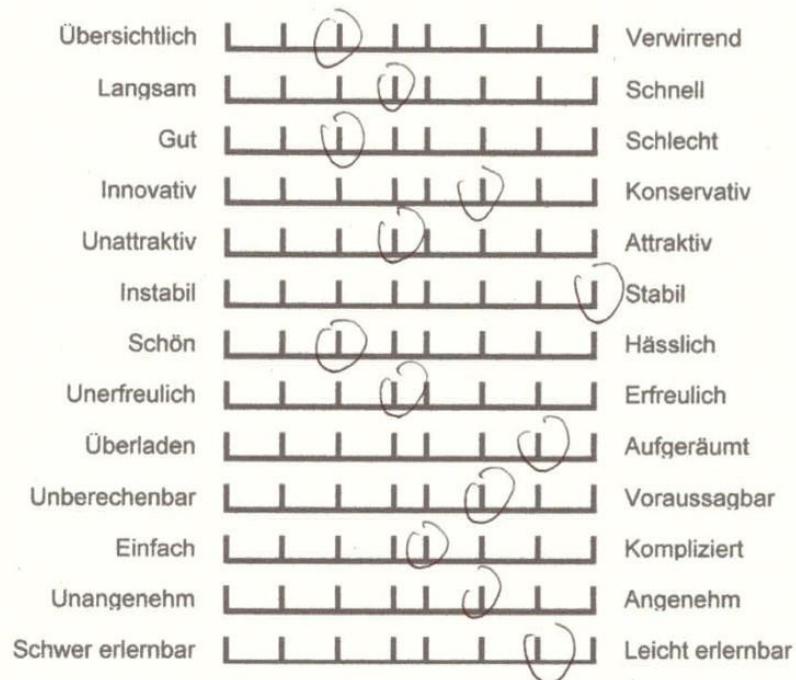
Würden Sie die TellMe-Eingabeleiste der herkömmlichen Vorgehensweise – Navigation zu den zu bearbeitenden Objekten um Aktionen auszuführen – bevorzugen?



Denken Sie, dass die TellMe-Eingabeleiste bei der alltäglichen Arbeit Zeit sparen kann?



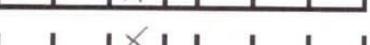
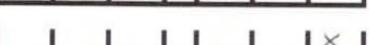
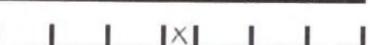
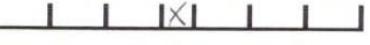
Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.



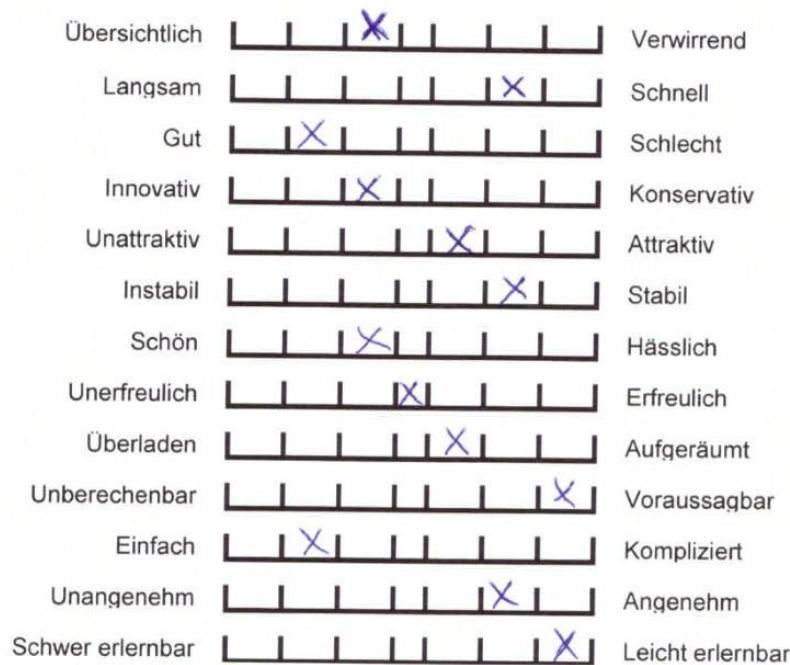
Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.

Übersichtlich		Verwirrend
Langsam		Schnell
Gut		Schlecht
Innovativ		Konservativ
Unattraktiv		Attraktiv
Instabil		Stabil
Schön		Hässlich
Unerfreulich		Erfreulich
Überladen		Aufgeräumt
Unberechenbar		Voraussagbar
Einfach		Kompliziert
Unangenehm		Angenehm
Schwer erlernbar		Leicht erlernbar

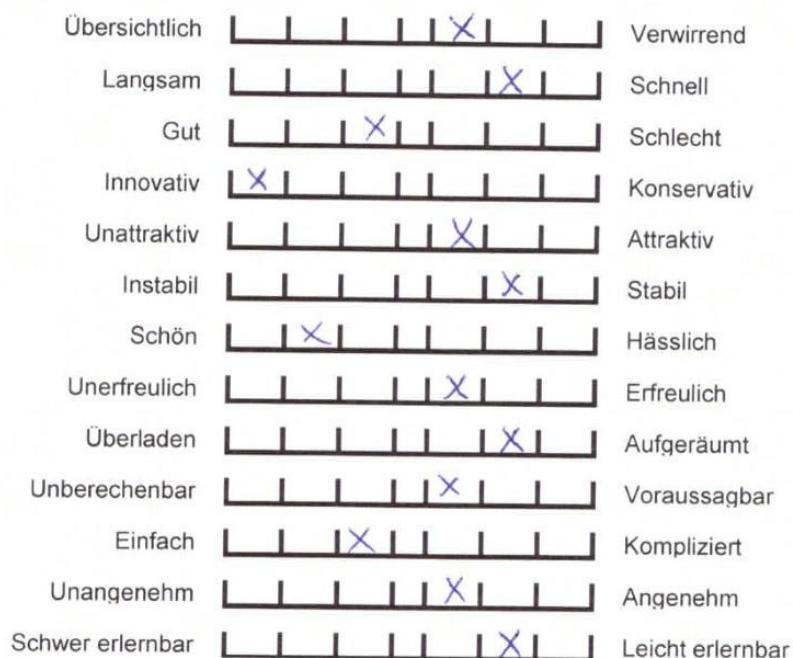
Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.

Übersichtlich		Verwirrend
Langsam		Schnell
Gut		Schlecht
Innovativ		Konservativ
Unattraktiv		Attraktiv
Instabil		Stabil
Schön		Hässlich
Unerfreulich		Erfreulich
Überladen		Aufgeräumt
Unberechenbar		Voraussagbar
Einfach		Kompliziert
Unangenehm		Angenehm
Schwer erlernbar		Leicht erlernbar

Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.



Bitte tragen Sie im Folgenden Ihren Eindruck der getesteten Anwendung ein. Markieren Sie hierzu für jedes Paar an Gegensätzen, welche der beiden Beschreibungen eher auf die Anwendung zutrifft.



8 Literaturverzeichnis

- Busse Susanne [et al.]** Federated Information Systems: Concepts, Terminology and Architectures [Buch]. - [s.l.] : Technische Universität Berlin, 1999.
- Damerau Fred J.** A technique for computer detection and correction of spelling errors [Journal] // Communications of the ACM. - 1964. - 3 : Bd. 7. - S. 171-176.
- Gale William, Church Kenneth Ward und Yarowsky David** Estimating the upper and lower bounds on the performance of word-sense disambiguation programs [Konferenz] // Proceedings of the 30th Annual Meeting on Association for Computational Linguistics. - Newark, Delaware : [s.n.], 1992. - S. 249-256.
- Gould John D und Lewis Clayton** Designing for Usability: Key Principles and What Designers Think [Artikel] // Communications of the ACM / Hrsg. ACM. - 1985. - 3. - 28. - S. 300-311.
- Hart Sandra G** NASA-Task Load Index (NASA-TLX); 20 Years Later [Konferenz] // Proceedings of the the human factors and ergonomics society annual meeting. - 2006. - Bd. 50. - S. 904-908.
- Hart Sandra G und Staveland Lowell E** Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research [Journal] // Advances in psychology. - [s.l.] : Elsevier, 1988. - Bd. 52. - S. 139-183.
- Held Uli** Cleverle Schnelleinstieg. - Weissach : [s.n.], 2009.
- Jurafsky Daniel und Martin James H** Speech and language processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition [Buch]. - Upper Saddle River, NJ : Prentice-Hall, Pearson Education International, 2009.
- Ludewig Jochen und Licher Horst** Software Engineering - Grundlagen, Menschen, Prozesse, Techniken [Buch]. - [s.l.] : dpunkt.verlag, 2007.
- Marcus Mitchell P, Marcinkiewicz Mary Ann und Santorini Beatrice** Building a large annotated corpus of English: The Penn Treebank [Journal] // Computational linguistics. - [s.l.] : MIT Press, 1993. - 2 : Bd. 19. - S. 313-330.
- Mayhew Deborah J** Principles and Guidelines in Software User Interface Design [Buch]. - [s.l.] : Prentice-Hall, Inc., 1991.
- Mayhew Deborah J** The Usability Engineering Lifecycle: A Practitioner's Guide to User Interface Design [Buch]. - [s.l.] : Morgan Kaufmann Publishers, 1999.

- Osgood Charles E** The nature and measurement of meaning [Journal] // Psychological Bulletin. - 5 1952. - 3 : Bd. 49. - S. 197-237.
- Rosson Mary Beth und Carroll John M** Usability Engineering: Scenario-based development of human-computer-interaction [Buch]. - [s.l.] : Elsevier, 2001.
- Schiller Anne [et al.]** Guidelines für das Tagging deutscher Textcorpora mit STTS [Bericht] / Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart. - 1999.
- Scriven Michael S** The methodology of evaluation [Buchabschnitt] // Perspectives of curriculum evaluation. - [s.l.] : Chicago: Rand McNally, 1967.
- Toutanova Kristina [et al.]** Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network [Konferenz] // Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. - Edmonton, Canada : Association for Computational Linguistics, 2003. - S. 173-180.
- UEQ-Online** UEQ-Online User Experience Questionnaire [Online]. - UEQ-Online. - 23. 4 2015. - <http://www.ueq-online.org>.
- Wixon Dennis und Wilson Chauncey** The usability engineering framework for product design and evaluation [Journal] // Handbook of Human-Computer Interaction. - [s.l.] : North-Holland Amsterdam, 1997. - Bd. 2. - S. 653-68.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben.

Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.

Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens.

Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht.

Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Stuttgart, 12.05.2015

Philipp Pirrun