



Eine strukturierte Vorgehensweise zur Entwicklung von Interaktionskonzepten auf Basis von Mustern am Beispiel eines Landkartenbrowsers

Markus Specker, Ingo Pöhler, Alexander Krebs
Siemens IT Solutions and Services GmbH

C-LAB Report

Vol. 10 (2011) No. 05

Cooperative Computing & Communication Laboratory

Die diesem C-LAB Report zugrunde liegenden Arbeiten wurden im Rahmen des vom Bundesministerium für Wirtschaft und Technologie (BMWi) initiierten Forschungsprogramms THESEUS gefördert (Förderkennzeichen 01MQ07014).

Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages

ISSN 1619-7879

C-LAB ist eine Kooperation
der Universität Paderborn und der Siemens IT Solutions and Services GmbH
www.c-lab.de
info@c-lab.de

C-LAB Report

**Herausgegeben von
Published by**

**Dr. Wolfgang Kern, Siemens IT Solutions and Services GmbH
Prof. Dr. Franz-Josef Rammig, Universität Paderborn**

Das C-LAB - Cooperative Computing & Communication Laboratory - leistet Forschungs- und Entwicklungsarbeiten und gewährleistet deren Transfer an den Markt. Es wurde 1985 von den Partnern Nixdorf Computer AG (nun Siemens IT Solutions and Services GmbH) und der Universität Paderborn im Einvernehmen mit dem Land Nordrhein-Westfalen gegründet.

Die Vision, die dem C-LAB zugrunde liegt, geht davon aus, dass die gewaltigen Herausforderungen beim Übergang in die kommende Informationsgesellschaft nur durch globale Kooperation und in tiefer Verzahnung von Theorie und Praxis gelöst werden können. Im C-LAB arbeiten deshalb Mitarbeiter von Hochschule und Industrie unter einem Dach in einer gemeinsamen Organisation an gemeinsamen Projekten mit internationalen Partnern eng zusammen.

C-LAB - the Cooperative Computing & Cooperation Laboratory - works in the area of research and development and safeguards its transfer into the market. It was founded in 1985 by Nixdorf Computer AG (now Siemens IT Solutions and Services GmbH) and the University of Paderborn under the auspices of the State of North-Rhine Westphalia.

C-LAB's vision is based on the fundamental premise that the gargantuan challenges thrown up by the transition to a future information society can only be met through global cooperation and deep interworking of theory and practice. This is why, under one roof, staff from the university and from industry cooperate closely on joint projects within a common research and development organization together with international partners. In doing so, C-LAB concentrates on those innovative subject areas in which cooperation is expected to bear particular fruit for the partners and their general well-being.

ISSN 1619-7879

C-LAB
Fürstenallee 11
33102 Paderborn

fon: +49 5251 60 60 60
fax: +49 5251 60 60 66
email: info@c-lab.de
Internet: www.c-lab.de

© Siemens IT Solutions and Services GmbH und Universität Paderborn 2011

Alle Rechte sind vorbehalten.

Insbesondere ist die Übernahme in maschinenlesbare Form sowie das Speichern in Informationssystemen, auch auszugsweise, nur mit schriftlicher Genehmigung der Siemens IT Solutions and Services GmbH und der Universität Paderborn gestattet.

All rights reserved.

In particular, the content of this document or extracts thereof are only permitted to be transferred into machine-readable form and stored in information systems when written consent has been obtained from Siemens IT Solutions and Services GmbH and the University of Paderborn.

INHALTSVERZEICHNIS

1	EINLEITUNG UND MOTIVATION.....	4
2	VORSTELLUNG DER VORGEHENSWEISE	4
2.1	Der Problem Frame Ansatz.....	5
2.2	Erweiterung des Problem Frames-Ansatzes: Prinzipienanalyse in MCI-Entwurfsmustern	8
2.2.1	Herleitung von MCI-Prinzipien aus MCI-Entwurfsmustern	8
2.2.2	Berücksichtigung des Nutzungskontextes	9
2.2.3	Erweiterung der Problem Frames-Notation	11
2.2.4	Zusammenführung der MCI-Prinzipien mit MCI-Entwurfsmustern	12
3	ANWENDUNG DES ANSATZES FÜR DIE MODELLIERUNG EINES LANDKARTEN-BROWSERS	13
3.1	Identifizierte Tasks im Landkartenbrowser.....	13
3.1.1	Task 1: Artikel und Bilder sollen interaktiv über die Landkarte exploriert werden können. .	13
3.1.2	Task 2: Inhalte aus verschiedenen „Richtungen“ zugänglich machen	14
3.1.3	Task 3: Grenzverläufe bzw. deren Änderungen sollen nachvollzogen werden können.....	14
3.1.4	Task 4: Die Karte soll bei der Erstellung von Artikeln benutzen werden	15
3.1.5	Task 5: Es soll möglich sein, Zusammenhänge zwischen beliebigen Artikeln sichtbar zu machen. 15	
3.1.6	Task 6: Darstellungen komplexer Zusammenhänge	15
3.2	Vorgehensweise zur Modellierung des Landkartenbrowsers.....	15
3.2.1	Aufbau des Requirements R3.....	16
3.2.2	Herleitung der Problem Frames.....	17
3.2.3	Erweiterung der Problem Frames um MCI-Prinzipien und Matching mit MCI- Entwurfsmustern.....	18
4	UMSETZUNG DER KONZEPTE IN DAS USER INTERFACE	23
4.1	Darstellung der Entitäten als Symbole	24
4.2	Darstellung von verdichteten Entitäten.....	25
4.3	Umgang mit Platzbeschränkung	28
5	FAZIT	29

1 Einleitung und Motivation

Beim Aufbau von Wissensplattformen müssen unter anderem Navigations- und Interaktionskonzepte erarbeitet werden, um das dort gesammelte Wissen sinnvoll zu visualisieren und verständlich zu machen, sodass Nutzer dieses Wissen leicht nachvollziehen und explorieren können. Hierfür muss auf der einen Seite klar sein, welches Wissen visualisiert werden soll, und auf der anderen Seite feststehen, welche Ziele mit der Visualisierung verfolgt werden sollen. Diese Aspekte sind von entscheidender Bedeutung, um für Nutzer und deren Aufgaben angepasste Lösungen entwickeln zu können.

Im Rahmen des THESEUS-Programms, einem vom Bundesministerium für Wirtschaft und Technologie (BMWi) initiierten Forschungsprojekt¹, wurden im Anwendungsfall ALEXANDRIA verschiedene Interaktionskonzepte entwickelt, die Nutzer dabei unterstützen, sich Inhalte und Zusammenhänge zu erschließen, und es ihnen ermöglicht, auf diesen zu explorieren, um ihren Wissenstand zu erweitern. Unter anderem entstanden Konzepte zur Navigation auf Zeitleisten (Timeline-Browser, vgl. Specker/Pöhler/Krebs 2011a) sowie zum Explorieren von Beziehungen verschiedener Persönlichkeiten (Relationship-Browser).

Darüber hinaus entstand auch ein Konzept zum Browsen und Explorieren von historischen Informationen auf Landkarten. Dieses *Landkartenbrowser*-Konzept ist Gegenstand des vorliegenden Reportes. Ein Landkartenbrowser eignet sich insbesondere dann gut, wenn viele der zu visualisierenden Daten Verortungen (z. B. in Form von Geokoordinaten) besitzen. Dieser Report beschreibt eine musterbasierte Vorgehensweise zum Aufbau eines Landkartenbrowsers ausgehend von den Aufgaben des Nutzers und daraus abgeleitete Requirements bis hin zum fertigen Konzept und dessen Umsetzung, welche den gesamten Vorgang der Konzeption umfasst.

2 Vorstellung der Vorgehensweise

Das im Folgenden dargestellte Konzept zum Explorieren über Landkarten wurde mit Hilfe einer durch uns modifizierten Version des Problem Frames-Ansatzes von Jackson (2001) erarbeitet. Ziel des Ansatzes ist es, Lösungsvorschläge in Form von Entwurfsmustern zu generieren, um schon beim Entwurf Aspekte der Gebrauchstauglichkeit (Usability) mit in den Entwicklungsprozess einfließen zu lassen. Jacksons Ansatz wurde dafür so erweitert, dass speziell Probleme aus dem Bereich der Mensch-Computer-Interaktion (MCI) unter Berücksichtigung von Entwurfsmustern und Prinzipien betrachtet werden können. Diese Herangehensweise ermöglicht es, einerseits systematisch und strukturiert problemorientiert die verschiedenen Tasks, welche in Verbindung mit einem

¹ Die dargestellten Forschungsarbeiten wurden im Rahmen des THESEUS-Forschungsprogramms vom Bundesministerium für Wirtschaft und Technologie (BMWi) unter dem Förderkennzeichen 01MQ07014 gefördert.

Landkartenbrowser stehen, zu erfassen, andererseits für die modellierten Probleme Best Practise-Lösungen in Form von MCI-Entwurfsmustern zu finden. Durch diese Erweiterung kann das bisher eher wenig strukturierte Erfahrungswissen, welches in den MCI-Entwurfsmustern enthalten ist, systematisch angewendet werden.

Teile dieser Vorgehensweise wurden zuvor bereits in verschiedenen Beiträgen publiziert (vgl. Specker/Wentzlaff 2006; Specker/Wentzlaff 2007; Specker/Düchting 2009, Specker et al. 2011b). Für die Entwicklung des Landkartenbrowsers wurde diese Methodik aufgegriffen und konsequent auf die dort identifizierten Tasks angewandt.

In den folgenden Unterpunkten wird der Problem Frames-Ansatz sowie die von uns entwickelte Erweiterung detailliert beschrieben.

2.1 Der Problem Frame Ansatz

Der Problem Frames-Ansatz (vgl. Jackson 2001) dient dazu, Probleme musterbasiert zu analysieren und zu modellieren. Im Gegensatz zu anderen Problemanalyse-Ansätzen, wie z. B. UML-Use Cases, ist dieser Ansatz stringenter und strukturierter, so dass sich Problemsituationen insgesamt klarer darstellen lassen. Problem Frames ermöglichen es, die Komplexität von Problemen durch Problemzerlegung erheblich zu reduzieren und in Teilproblemen strukturiert darzustellen. Die Teilprobleme können sich überlagern, weswegen Jackson von einer Problemzerlegung durch Projektion spricht. Durch diese Komplexitätsreduktion ist es relativ einfach möglich, die analysierten Probleme in den Entwurf zu überführen.

Der Problem Frames-Ansatz ist geleitet durch das Requirements-Engineering. Für jedes Requirement bestimmt man den entsprechenden Problemzusammenhang. Ebenso wie andere Ansätze bieten auch Problem Frames die Möglichkeit, Probleme grafisch zu repräsentieren. Jackson (2001) beschreibt fünf Basis-Problem Frames, welche durch Kombination miteinander erweitert und variiert werden können.

Ein Problem Frame besteht aus wenigen Basiselementen (vgl. die nachfolgende Abbildung 1), die auf verschiedene Weise miteinander kombinierbar sind. Dies ermöglicht eine insgesamt sehr flexible Handhabung der Probleme. Zu den Basiselementen im Problem Frame zählen Domänen (in Abbildung 1 als Rechtecke dargestellt), mit denen Entitäten in einem Problem (ähnlich wie beim Entity-Relationship-Ansatz) beschrieben werden können.

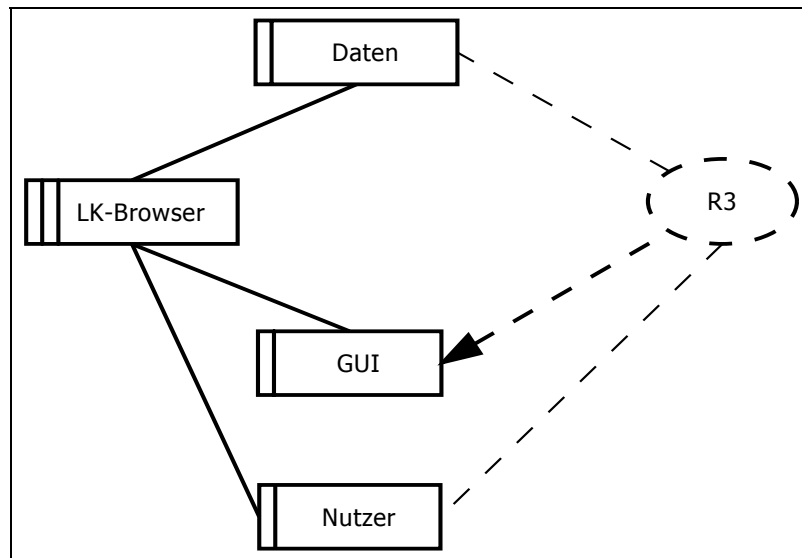


Abbildung 1: Beispiel eines Problem Frames

Man unterscheidet zwischen verschiedenen Domänentypen. So wird beispielsweise die zu erstellende Anwendung durch zwei Längsstriche besonders markiert. Domänen sind durch Schnittstellen (Interfaces) miteinander verbunden (einfache Verbindungslinien zwischen den Domänen in Abbildung 1). An diesen werden die Gemeinsamkeiten, die die Domänen miteinander teilen, notiert (gemeinsame Phänomene).

Jedes Problem Frame behandelt ein Teilproblem. Die Anforderungen innerhalb eines solchen Teilproblems werden in Requirements festgehalten (gestrichelte Ellipsen in Abbildung 1). Die Requirements beziehen sich auf bestimmte Domänen im Problem Frame. Nach Jackson (2001) sind dies Referenzen (gestrichelte ungerichtete Linien zwischen den Requirements und Domänen in Abbildung 1) oder Einschränkungen (gestrichelte gerichtete Linien zwischen den Requirements und Domänen Abbildung 1).

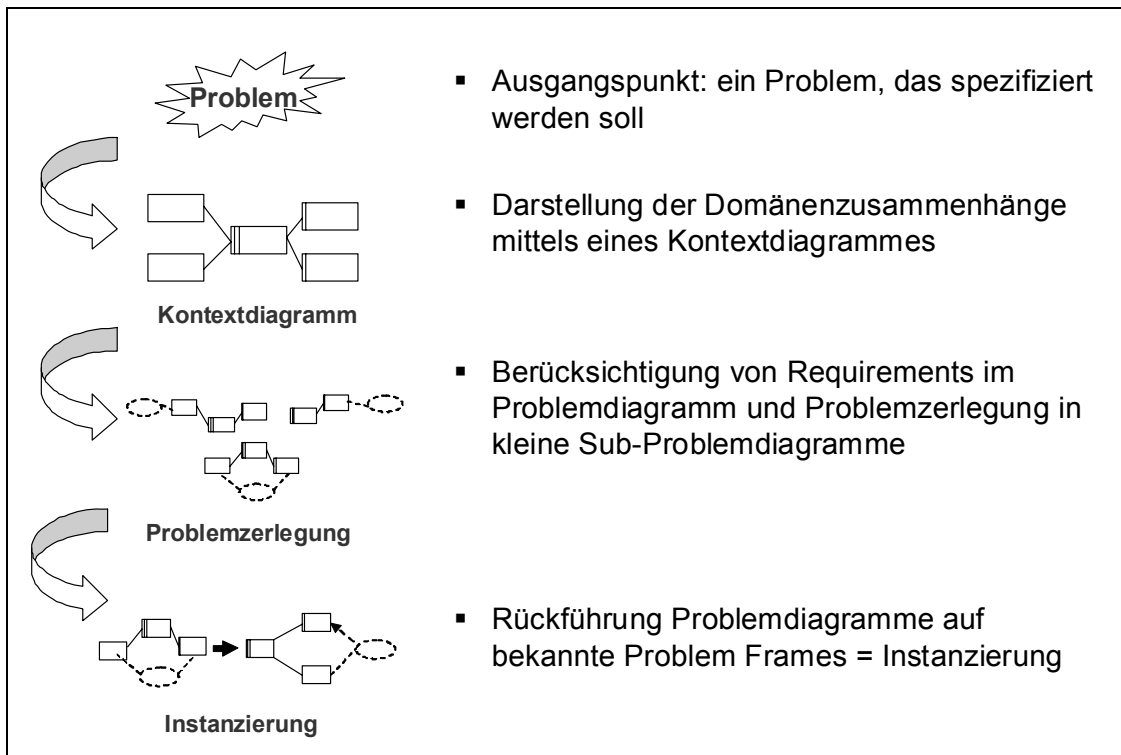


Abbildung 2: Vorgehensweise der Problemanalyse mittels Problem Frames

Um (MCI-)Probleme mit Problem Frames zu modellieren, sind folgende Schritte erforderlich (vgl. Abbildung 2):

1. Am Anfang steht ein reales Problem, welches modelliert werden soll. Bei MCI-Problemen wird zunächst eine Analyse des Nutzungskontextes durchgeführt.
2. Danach wird das gesamte Problem in einem Kontextdiagramm dargestellt, welches alle Domänen sowie deren Beziehungen zueinander (Schnittstellen) enthält, jedoch noch keine Requirements. Das Kontextdiagramm gibt eine gute Übersicht über die beteiligten Domänen, ohne tiefer ins Detail zu gehen. An den Schnittstellen werden sogenannte gemeinsame Phänomene notiert, d. h. die Informationen, die sich beide Domänen teilen.
3. Das Kontextdiagramm bildet die Basis für die Problemzerlegung und Überführung in Problemdiagramme. In diesem Schritt werden Requirements berücksichtigt und das komplexe Problem in einzelne Teilprobleme zerlegt, welche sich einfacher lösen lassen als das Gesamtproblem. Die Problemzerlegung und Überführung in Problemdiagramme erfolgt parallel. Für jedes Teilproblem wird angenommen, dass alle anderen Teilprobleme jeweils bereits gelöst sind, um die Abhängigkeiten innerhalb der Teilprobleme zu minimieren. Auf diese Weise wird das ursprünglich komplexe Problem in kleine überschaubare Teilprobleme zerlegt.

4. Im letzten Schritt erfolgt, sofern möglich, eine Rückführung auf bereits bekannte Probleme, d. h. das Problem wird nach darin enthaltenen bekannten Problem Frames analysiert. Als Ergebnis erhält man für die Teilprobleme jeweils Problemdiagramme, welche konkrete Instanzen abstrakter Problem Frames darstellen, instanziiert durch die aktuelle Problembeschreibung

2.2 Erweiterung des Problem Frames-Ansatzes: Prinzipienanalyse in MCI-Entwurfsmustern

Jacksons Problem Frames-Ansatz (2001) hat den Nachteil, dass nichtfunktionale Aspekte, wie etwa usabilityspezifische Zusammenhänge, nicht berücksichtigt werden. Außerdem beschränkt sich Jacksons Ansatz lediglich auf die Problemdarstellung, erläutert jedoch nicht die Lösung der Probleme. Hier kommen MCI-Prinzipien und die damit verknüpften Entwurfsmuster ins Spiel.

Für die spezielle Berücksichtigung von MCI-Problemen wurde der Problem Frames-Ansatz dahingehend erweitert, dass zusätzliche MCI-Prinzipien in die Problem Frames integriert wurden. Diese verweisen auf MCI-Entwurfsmuster, so dass die modifizierten Problem Frames gleichzeitig Lösungsansätze in Form von Entwurfsmustern enthalten. Die bei der Problemanalyse hervorgebrachten Entwurfsmuster bilden die Grundlage für die dann folgende Modellierung. Um die Erweiterungen auch in der Problem Frame Notation sichtbar zu machen, ist eine Modifikation der Darstellung der Problem Frames erforderlich. Diese Erweiterung wird in den folgenden Unterabschnitten weiter detailliert.

2.2.1 Herleitung von MCI-Prinzipien aus MCI-Entwurfsmustern

Um in einer konkreten Problemsituation passende Entwurfsmuster als Lösungsansätze erhalten zu können, muss zunächst geklärt werden, welche Faktoren zur Auswahl eines Entwurfsmusters führen. Hinweise dazu befinden sich in den Beschreibungstexten der Entwurfsmuster. Aus diesen geht hervor, wann die Entwurfsmuster nach Ansicht des jeweiligen Autors zum Einsatz kommen sollten.

Zum anderen muss sich der User Interface (UI)-Designer des aktuellen Nutzungskontextes, in welchem das Problem beschrieben ist, bewusst sein. Der Beschreibungstext eines in Frage kommenden Entwurfsmusters muss dem UI-Designer ermöglichen, zu entscheiden, ob das in Frage kommende Entwurfsmuster zum aktuellen Nutzungskontext passt oder nicht. Eine Abstraktion der Entwurfsmusterbeschreibung hin zu vergleichbaren Faktoren, die erläutern, in welchen Fällen das jeweilige Entwurfsmuster zum Einsatz kommt, hilft, um zu einer strukturierten Vorgehensweise zu gelangen. Solche Faktoren nennen wir Entwurfsmusterprinzipien. Abbildung 3 veranschaulicht unsere Vorgehensweise.

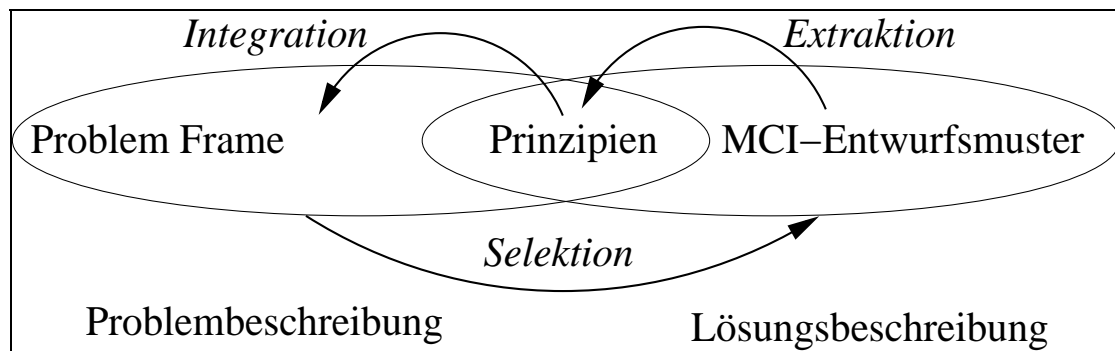


Abbildung 3: Verdeutlichung der Vorgehensweise zur Verknüpfung von Problem- und Lösungsmustern

Smith/Williams (2002, S. 263), beschreiben Performance-Muster als Umsetzung von einem oder mehreren Performance-Prinzipien. Für unsere Vorgehensweise betrachten wir die MCI-Entwurfsmuster ähnlich, d. h. jedes MCI-Entwurfsmuster wird jeweils durch einem oder mehrere Prinzipien repräsentiert. Diesen Zusammenhang nutzen wir aus, um aus MCI-Entwurfsmustern MCI-Prinzipien zu extrahieren. Ein Beispiel hierfür ist das Prinzip „Focus on Context“. Dieses Prinzip dient dazu, zu gewährleisten, dass ein Nutzer einen Zusammenhang überhaupt wahrnehmen kann, weil die Interaktion dann dort stattfindet, wo der Nutzer gerade seinen Fokus hat. Dieses Prinzip ist z. B. in den Entwurfsmustern „Same Page Error Messages“ und „Edit in Place“ (vgl. Tidwell 2005, S. 239 u. 249) enthalten.

Die extrahierten Prinzipien aus den Entwurfsmustern wurden darüber hinaus mit MCI-Prinzipien aus verschiedenen Quellen (vgl. Brown/Cunningham 1989; Constantine/Lockwood 2002; Cooper/Reimann/Cronin 2007; Mahemoff/Johnston 1998 a/b; Mayhew 1992; Oviatt 2006; Reeves et al. 2004) verglichen und konsolidiert, um sicherzustellen, dass diese Prinzipien generell bei der Gestaltung von MCI-Problemen eine Rolle spielen. Die konsolidierten Prinzipien integrieren wir sodann in den Problem Frames-Ansatz. Durch die Referenz der Prinzipien zu den Entwurfsmustern können dann in einer aktuellen Problemsituation passende Lösungsmuster selektiert werden.

2.2.2 Berücksichtigung des Nutzungskontextes

Nach DIN EN ISO 9241-11 (1998) setzt sich der Nutzungskontext aus den Komponenten „Nutzer“, „Nutzungsumgebung“, „Hard- und Software“ sowie „Aufgabe“ zusammen. Die den Entwurfsmustern zugrunde liegenden Prinzipien können diesen Nutzungskontextkomponenten zugeordnet werden, z. B. kann das Prinzip „Focus on Context“ dem Benutzer zugeordnet werden, da es sich auf dessen Eigenschaften (beschränkte Wahrnehmungsfähigkeit) bezieht. Für die Verknüpfung der Prinzipien mit Problem Frames kann dieser Zusammenhang ausgenutzt werden, denn eine Differenzierung der Prinzipien nach ihrem Nutzungskontext gibt Aufschluss darüber, an welcher Stelle im Problemdiagramm diese eingebettet werden können (vgl. die nachfolgende Abbildung 4). So ist die Nutzungskontextkomponente „Aufgabe“ den

Requirements im Problem Frame zugeordnet, während die Nutzungsumgebung und der Benutzer den Domänen zugeteilt sind. Die Interaktion zwischen den Nutzungskontextelementen findet ihre Entsprechung in den gemeinsamen Phänomenen.

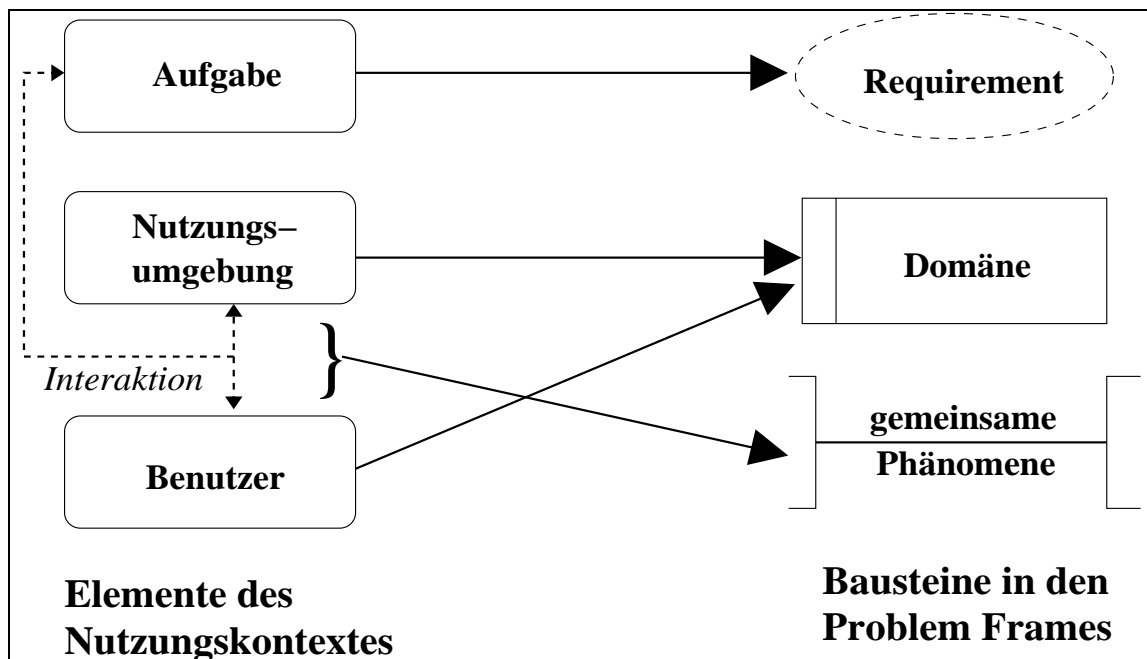


Abbildung 4: Mapping der Nutzungskontextelemente auf Elemente aus Problem Frames

Eine Zuordnung der Prinzipien zu den Elementen des Nutzungskontextes ohne weitere Hilfsmittel ist jedoch nicht immer unmittelbar durchführbar. Es fehlt bisher noch eine Erklärung, aus der ersichtlich wird, warum die Elemente den jeweiligen Elementen des Nutzungskontextes zugeordnet werden. Hier hilft die im Folgenden beschriebene Vorgehensweise.

Bei der Modellierung mit Problem Frames existieren neben Domänen, welche individuell auf die jeweilige Situation zugeschnitten sind, auch Domänen, die immer wieder vorkommen. Beispielsweise gibt es regelmäßig eine Domäne „Benutzer“, sobald Nutzerinteraktionen modelliert werden. Daneben gibt es eine Domäne „GUI“, welche die Interaktion zwischen dem Anwender und dem System abbildet. Unabhängig davon, was modelliert werden soll, gibt es immer eine Domäne „Maschine“, die für das zu erstellende System steht. Die Maschinendomäne kann dabei mit einem auf die Situation angepassten Begriff instanziiert sein (z. B. würde sie im Fall des hier modellierten Landkartenbrowsers „Landkartenbrowser“ heißen). Diese Domänen kann man jeweils nach verschiedenen Eigenschaften kategorisieren. Diese Domäneneigenschaften dienen als Hilfsmittel, um Prinzipien Domänen zuzuordnen.

Abbildung 5 zeigt zum Beispiel die Domäne „GUI“ mit verschiedenen zugehörigen Domäneneigenschaften (dargestellt durch Rauten). An den Domäneneigenschaften angehängt sind passende Prinzipien, welche die jeweiligen Domäneneigenschaften unterstützen (dargestellt durch halbrunde Rechtecke). Diese können bedingt sein (z. B. kann die Domäneneigenschaft „Platz“ je nachdem, ob viel oder wenig Platz zur Verfügung steht, durch das Prinzip „Platz sparen“ oder das Prinzip „Platz ausnutzen“ unterstützt werden), oder an keine Bedingungen geknüpft sein.

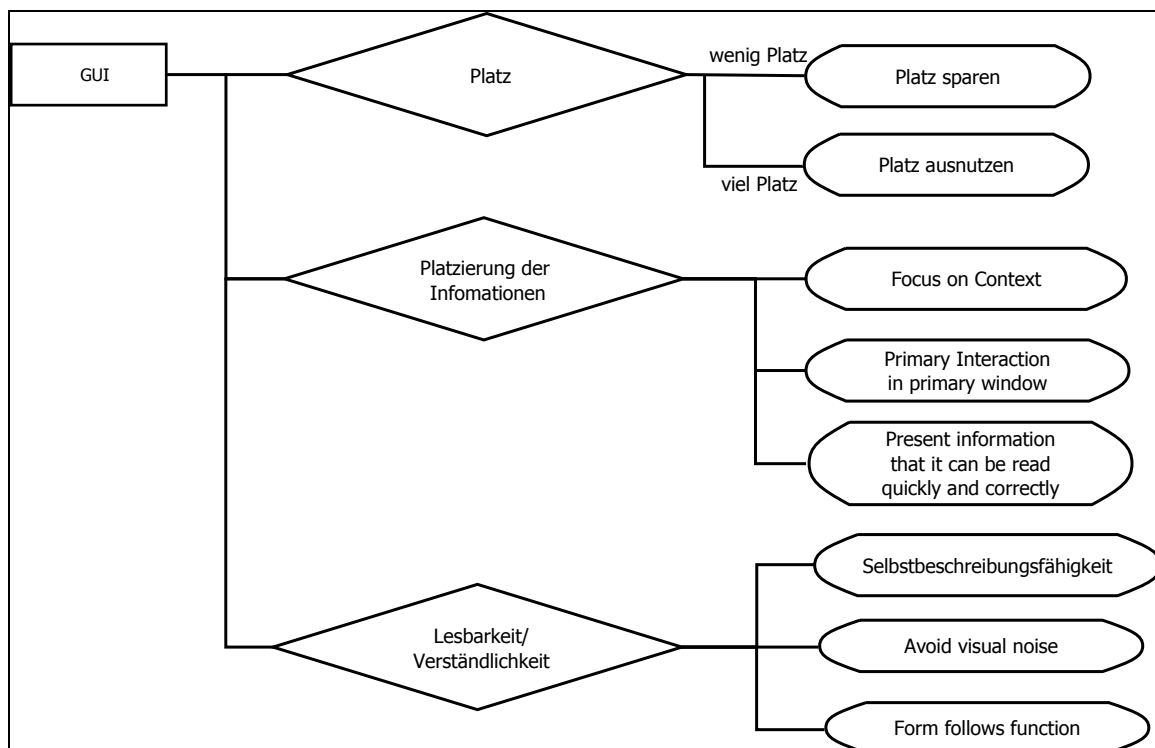


Abbildung 5: Domäneneigenschaften der Domäne „GUI“ mit den dazugehörigen Prinzipien

Auch für die anderen Domänen existieren entsprechende Domäneneigenschaften, welche mit unterstützenden Prinzipien verknüpft sind. Des Weiteren gibt es Interaktionseigenschaften, welche an die Interaktionen, d. h. an die gemeinsamen Phänomene, zwischen zwei Domänen geknüpft sind. Deren Struktur ist die gleiche wie die der Domäneneigenschaften, mit dem Unterschied, dass an der linken Seite statt einer Domäne zwei Domänen dargestellt sind.

2.2.3 Erweiterung der Problem Frames-Notation

Die aus den MCI-Entwurfsmustern erarbeiteten Prinzipien fließen in die Problem Frames-Modellierung ein. Um dieses visuell zu verdeutlichen, ist eine Modifikation der Problem Frames-Notation erforderlich. Die Notation der zusätzlichen Elemente entspricht der Notation der Domänen- und Interaktionseigenschaften (vgl. Abbildung 5). Die jeweils relevanten Eigenschaften werden direkt an die Domänen bzw. zwischen die Domänen gehängt. An diesen wiederum hängen die für die jeweilige Problemsituation passenden Prinzipien. Dabei müssen nur die Eigenschaften und Prinzipien dargestellt werden, welche gerade für das jeweilige Problem relevant sind.

Da diese Notationserweiterung den Umfang der Problem Frames-Diagramme erhöhen, sind die erweiterten Notationselemente in den nachfolgenden Abbildungen kenntlich gemacht worden, indem sie in einer anderen Farbe dargestellt sind. Die erweiterte Notation der Problem Frames wird in den folgenden Beispielen verwendet.

2.2.4 Zusammenführung der MCI-Prinzipien mit MCI-Entwurfsmustern

Die mit Hilfe von Domäneneigenschaften in die Problem Frames integrierten MCI-Prinzipien dienen dazu, um bei der späteren Modellierung jeweils zu den Prinzipien passende MCI-Entwurfsmuster zu erhalten. Um von Prinzipien zu Entwurfsmustern zu gelangen, muss der in Abschnitt 2.2.1 beschriebene Schritt (Herleitung von MCI-Prinzipien aus MCI-Entwurfsmustern) rückwärts gemacht werden. Die „passenden“ Entwurfsmuster sind genau die Entwurfsmuster, aus denen das Prinzip extrahiert wurde. Für diesen Zweck wurden die Entwurfsmuster der Sammlungen von Tidwell (2005), Welie (2008) sowie der Interface Design Pattern-Sammlung der Fachhochschule Potsdam (Patternbrowser, 2008) analysiert. Diesen Schritt werden wir im Folgenden als „Matching“ bezeichnen.

Dargestellt werden kann das „Matching“ durch eine einfache Tabelle (siehe Tabelle 1). Dabei stehen die Spalten für die im jeweiligen Problem Frame enthaltenen Prinzipien und die Zeilen für die Entwurfsmuster, die mindestens eins der Prinzipien umsetzen. Zellen, die zwischen Prinzipien und Entwurfsmustern stehen, enthalten ein „X“, falls das entsprechende Entwurfsmuster das durch die Spalte dargestellte Prinzip umsetzt.

Alle in der Tabelle enthaltenen Entwurfsmuster sind potenziellen Kandidaten bzw. Vorschläge für die Lösung des durch das Problem Frame modellierten Problems. Die endgültige Auswahl der Entwurfsmuster liegt beim Designer, ein hilfreicher Indikator für die Auswahl könnte aber z.B. die Anzahl der „X“ in der Zeile eines Entwurfsmusters sein.

Prinzipien / Entwurfsmuster	Steuerbarkeit	Make actions reversible	Changes in midstream
Detail on Demand	X		
Slider	X		
Collapsible Panels	X		
Live Filter	X		
(Multi Level) Undo	X	X	X

Tabelle 1 Darstellung des "Matchings"

3 Anwendung des Ansatzes für die Modellierung eines Landkarten-Browsers

Ein wesentliches Ziel ist es, dass der Landkarten-Browser eine explorierende Art der Navigation ermöglichen soll, damit sich Nutzer der Plattform auf verschiedene Weise über die dort hinterlegten Inhalte informieren können. Durch die im letzten Abschnitt beschriebene Vorgehensweise ist für jedes Requirement ein Problem Frame-Diagramm und eine Tabelle mit referenzierten Entwurfsmustern erforderlich. Im Anwendungsszenario ALEXANDRIA des THESEUS-Programms, wurde diese Vorgehensweise auf alle Requirements des Landkartenbrowsers angewandt. Dieser Report soll jedoch vor allem das Vorgehen an sich verdeutlichen und erhebt keinen Anspruch auf eine vollständige Spezifikation. Daher erfolgt hier eine Auswahl, die das Vorgehen hinreichend demonstriert. Wir beschränken uns darauf, die Vorgehensweise an einem Beispiel deutlich zu machen, die weiteren Schritte werden verkürzt vorgestellt oder ausgelassen.

3.1 Identifizierte Tasks im Landkartenbrowser

Die folgende Auflistung definiert Aufgaben (Tasks) für den Landkarten-Browser, welche Nutzer potenziell auf dieser Plattform durchführen können. Diese Aufgaben sind das Ergebnis eines Brainstormings und bilden die Grundlage der konzeptionellen Entwicklung des Landkartenbrowsers.

1. Artikel und Bilder sollen interaktiv über die Landkarte exploriert werden können.
2. Inhalte sollen aus verschiedenen Blickrichtungen zugänglich gemacht werden.
3. Grenzverläufe bzw. deren Änderungen sollen nachvollzogen werden können.
4. Die Karte soll bei der Erstellung von Artikeln benutzen werden.
5. Es soll möglich sein, Zusammenhänge zwischen beliebigen Artikeln sichtbar zu machen.
6. Darstellungen komplexer Zusammenhänge (Bsp.: Landkarten „proportionalisieren“)

Die dargestellten Tasks sind nach Priorität geordnet. Sie werden im Folgenden kurz erläutert. Für die Realisierung wurden die Tasks 1 und 2 berücksichtigt, die anderen Tasks stellen Erweiterungsmöglichkeiten dar.

3.1.1 Task 1: Artikel und Bilder sollen interaktiv über die Landkarte exploriert werden können.

Artikel und Bilder sollen interaktiv exploriert werden können, was voraussetzt, dass sie zunächst einmal dargestellt werden. Je nach Art des Artikels kann dies durch ein entsprechendes Icon oder Schlagwort, welches an der verlinkten Position angezeigt

wird, geschehen. Damit ein Artikel auf der Landkarte angezeigt werden kann, benötigt er Verortungsinformationen (z. B. eine oder mehrere Koordinaten). Optional kann auch eine Fläche definiert werden, indem ein Bereich angegeben wird (z. B. mit einem Polygontool). Der Nutzer soll den Artikeltyp leicht erkennen können. Außerdem muss es dem Nutzer möglich sein, auch dann Artikel zu erkennen und auszuwählen, wenn diese auf sehr nahe beieinander liegenden Koordinaten verlinkt sind. Um leichtere Orientierung für den Nutzer zu schaffen, soll es eine Möglichkeit geben, die Auswahl der angezeigten Artikel individuell anzupassen.

3.1.2 Task 2: Inhalte aus verschiedenen „Richtungen“ zugänglich machen

Hier ist zunächst zu erklären was mit den „Richtungen“ gemeint ist. Je nachdem, aus welcher Perspektive ein Inhalt betrachtet wird, müssen verschiedene Interaktionsmöglichkeiten vorhanden sein, um zu den hinterlegten Inhalten zu gelangen. Folgende Beispiele verdeutlichen dieses:

- Mapping: Ort → Personen
Inhalte, die einem aktuellen Kartenausschnitt zugeordnet sind, sollen gleichzeitig verschiedenen historischen Persönlichkeiten zugeordnet sein.

Bsp.: Der Kartenausschnitt befindet sich über einer bestimmten Stadt und man kann so sehen, welche Personen mit diesem Ort verknüpft sind, weil diese dort in irgendeiner Weise gewirkt haben.
- Mapping: Person → Ort
Die umgekehrte Richtung: alle Inhalte zu einer bestimmten, aktuell ausgewählten Person, die sich auf einer Karte darstellen lassen anzeigen und alle andere Inhalte ausblenden.
- Ebenso ist dieses denkbar mit anderen Inhalten, z. B. Ereignis → Ort und Ort → Ereignisse.

3.1.3 Task 3: Grenzverläufe bzw. deren Änderungen sollen nachvollzogen werden können

Ein Anwendungsfall für historische Informationen in Bezug auf eine geografische Visualisierung ist, dem Nutzer das Nachvollziehen von Grenzverschiebungen zu ermöglichen. Voraussetzung hierfür ist, dass ein entsprechendes Kartenmaterial vorhanden ist oder dass der Nutzer dieses im Sinne des Web 2.0 als nutzergenerierten Inhalt selbst erstellen kann.

Ist solches Kartenmaterial vorhanden, lassen sich Sachverhalte darstellen wie:

- der Verlauf von Schlachten und Kriegen und die damit verbundenen Grenzverschiebungen,

- Völkerwanderungen,
- Veränderungen von Kultur- und Sprachräumen etc.

3.1.4 Task 4: Die Karte soll bei der Erstellung von Artikeln benutzen werden

Das bereits vorhandene Kartenmaterial kann auch für die Erstellung von Artikeln verwendet werden. Beispielsweise können Kartenausschnitte gewählt werden, um in Artikeln als Illustrationen zu dienen (ergänzend zu anderen Bildern). Dieses kann technisch durch die Auswahl zweier Koordinaten (linke obere und rechte untere Ecke) oder eines Gebietes, z. B. eines Landes (sofern in der Datenbank hinterlegt), geschehen.

3.1.5 Task 5: Es soll möglich sein, Zusammenhänge zwischen beliebigen Artikeln sichtbar zu machen.

Mit dieser Task ist folgende Idee gemeint: Das Anklicken zweier beliebiger Länder oder Orte führt zu einer Anfrage, die ausgibt, welche Beziehungen zwischen diesen beiden Entitäten bestehen.

Ebenso denkbar ist das Anklicken nur eines Landes oder Ortes, was zu einer Anfrage führt, die auflistet, welche weiteren Orte oder Länder hiermit in Beziehung stehen bzw. standen und welcher Natur diese Beziehung ist.

3.1.6 Task 6: Darstellungen komplexer Zusammenhänge

Diese Task stellt eine Erweiterung des Landkarten-Browsers im Sinne der Informationsvisualisierung dar. Dabei spielen folgende Gedanken eine Rolle: Die Darstellung der Landkarten erfolgt nicht nach tatsächlicher Verteilung der Landmasse, sondern die Länder werden nach Quantitäten skaliert (z. B. Einkommen, Bevölkerungsanzahl, auch Anzahl der Artikel in der Datenbank). Voraussetzung hierfür ist, dass entsprechende Daten vorliegen.

3.2 Vorgehensweise zur Modellierung des Landkartenbrowsers

Um zunächst einen Überblick über den gesamten Kontext zu erhalten, beginnt unsere Modellierung mit der Erstellung eines Kontextdiagramms. Abbildung 6 zeigt ein Kontextdiagramm, in dem die betreffenden Domänen und ihre Beziehungen zueinander dargestellt sind. Die Domäne „Daten“ bildet hier einen Sonderfall. Sie enthält sowohl die Domäne „Landkarte“ als auch die Domäne „Entität“, wobei mit letzterer jeweils die Konzepte, zu denen Informationen in Alexandria existieren (Person, Ort, Ereignis, Werk, Organisation), gemeint sind. Je nach Betrachtungsart taucht die Domäne „Daten“ in der Modellierung sowohl als Ganzes als auch mit ihren jeweiligen Bestandteilen auf.

Für die Interaktion mit dem Nutzer gehen wir davon aus, dass die Domäne „Daten“ bereits existiert. Daher wird diese nicht gesondert markiert. Erstellt wird als zentrales

Produkt der Landkarten-Browser, der in diesem Fall die zu erstellende Maschine darstellt sowie die dazugehörige grafische Benutzungsoberfläche (*Graphical User Interface*, GUI). Daher werden die zugehörigen Domänen entsprechend als Maschinendomäne (zwei zusätzliche Längsbalken in Abbildung 6) und als „Entworfenene Domäne“ (ein zusätzlicher Längsbalken in Abbildung 6) gekennzeichnet.

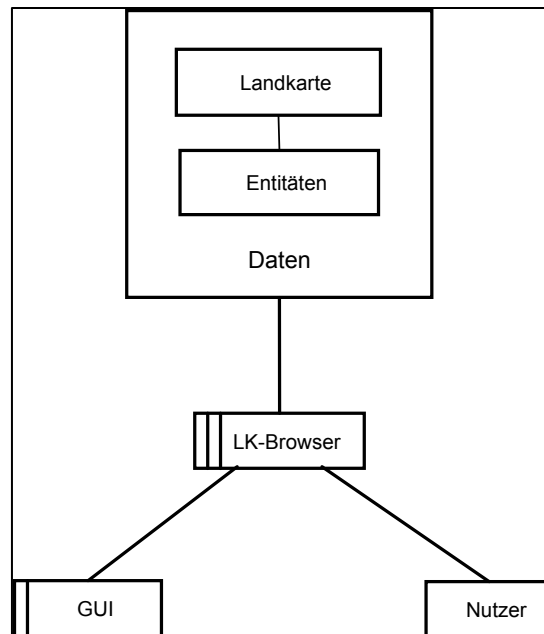


Abbildung 6: Kontextdiagramm mit den für die Modellierung relevanten Domänen

Im zweiten Schritt werden die weiter oben vorgestellten Tasks für die Herleitung von Requirements genutzt. Da die Anzahl aller abgeleiteten Requirements sehr groß ist, wird in diesem Report beispielhaft die Vorgehensweise am Requirement R3 stellvertretend für alle weiteren Requirements veranschaulicht. Die Vorgehensweise bei den weiteren Requirements erfolgt analog.

Das Requirement R3 besitzt mehrere Verfeinerungen, die das Requirement weiter detaillieren. Es bezieht sich auf die Modellierung von Task 2. Um einen Überblick über die Struktur des Requirements R3 und seine Detaillierungen zu bekommen, wird zunächst dieses Requirement im folgenden Unterabschnitt (in verkürzter Form) beschrieben.

3.2.1 Aufbau des Requirements R3

[R3] Es muss eine Möglichkeit für den Nutzer vorhanden sein, mit der Landkarte zu interagieren und sie anzupassen (z. B. verschieben, zoomen, ...).

R3.1 Die gesamte Karte sowie wichtige Metainformationen (Maßstab, Ausrichtung) müssen dem Nutzer angezeigt werden. Hierbei muss ein sinnvolles Maß für Abstraktion gefunden werden.

[...]

R3.2 Der Nutzer muss die Möglichkeit haben, den Kartenausschnitt anzupassen und angezeigte Entitäten auszuwählen.

[...]

R3.3 Auf dem gewählten Kartenausschnitt sollen die Inhalte, welche zu diesem Kartenausschnitt Bezug haben, dargestellt werden.

R3.3.1 Der Nutzer muss die Arten der Entitäten (Person, Gruppe, Werk,...) eindeutig und leicht voneinander unterscheiden können.

R3.3.2 Die Entitäten werden auf der Karte, entsprechend der Verortung, korrekt dargestellt.

R3.3.3 Es muss ersichtlich sein, wenn sich mehrere Entitäten auf ein und derselben Koordinate befinden.

R3.3.4 Der Nutzer soll sehen können, wo sich der aktuelle Kartenausschnitt auf der gesamten Karte befindet.

R3.3.5 Der Nutzer muss erkennen können, zu welchen weiteren Orten die gewählte Entität Beziehungen hat.

R3.3.6 Der Nutzer benötigt auch bei großen Mengen von Entitäten eine übersichtliche Darstellung auf der Karte.

3.2.2 Herleitung der Problem Frames

Aus dem Requirement R3 entstand das in Abbildung 7 dargestellte Problem Frame, welches die oberste Ebene des Requirements R3 beschreibt. Es ist eine Weiterentwicklung des Kontextdiagramms (Abbildung 6), in welchem nun das Requirement R3 hinzugefügt wurde. Die Domäne Daten wurde in diesem Fall als eine Domäne interpretiert. Noch nicht dargestellt sind in Abbildung 7 die gemeinsamen Phänomene.

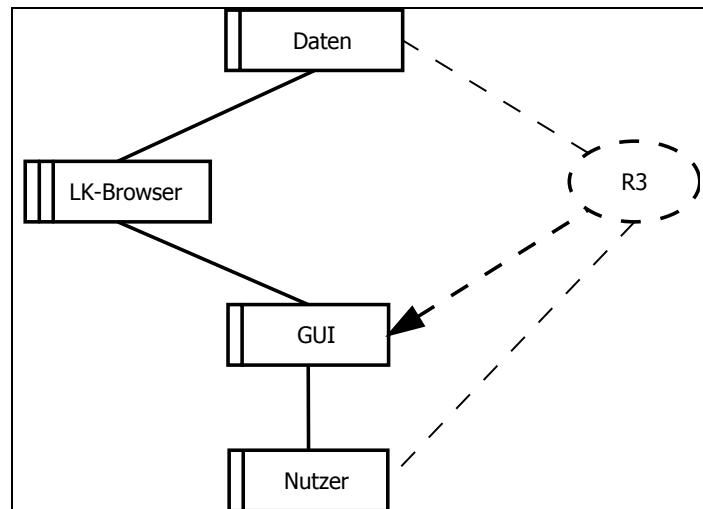


Abbildung 7 Problem Frame zu R3 (ohne gemeinsame Phänomene)

3.2.3 Erweiterung der Problem Frames um MCI-Prinzipien und Matching mit MCI-Entwurfsmustern

Im nächsten Schritt wird zu jedem Requirement eine Modellierung mittels Problem Frames durchgeführt, in welche anschließend die passenden Prinzipien integriert werden. Die Abbildungen Abbildung 8 und Abbildung 9 zeigen die Problem Frames zu zwei ausgewählten Detaillierungen des Requirements R3.

R3.3.1

Der Nutzer muss die Arten der Entitäten (Person, Gruppe, Werk,...) eindeutig und leicht voneinander unterscheiden können.

Das Problem Frame in Abbildung 8 modelliert die Detaillierung R3.3.1 des Requirements R3. Diese bezieht sich auf die unterschiedliche Darstellung der einzelnen Entitätentypen. Für die Darstellung sind die Domänen GUI als grafische Benutzerschnittstelle, und Nutzer als Bediener bzw. in diesem Fall Betrachter, von Bedeutung. Den beiden Domänen wurden über die Domäneneigenschaften (siehe Abschnitt 2.2.2) Prinzipien angehängt.

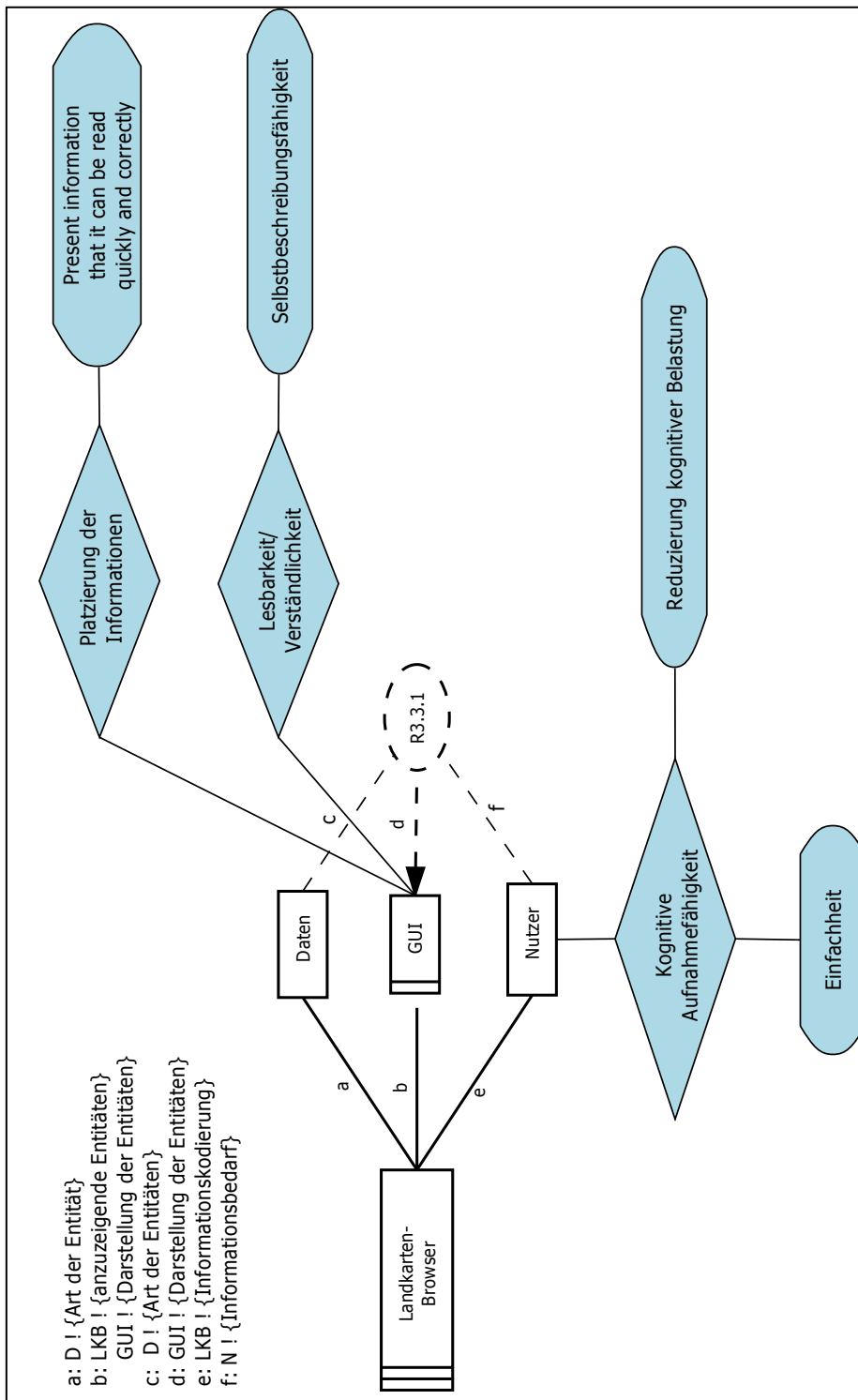


Abbildung 8: Problem Frame zu R3.3.1 inklusive Eigenschaften und Prinzipien

Für die Domäne GUI sind im Kontext des Requirements R3.3.1 die Domäneneigenschaften „Platzierung der Informationen“ sowie „Lesbarkeit/Verständlichkeit“ relevant, da es um das leichte und korrekte Unterscheiden der einzelnen Entitäten geht. Als wichtige Domäneneigenschaft des Nutzers wurde die „Kognitive Aufnahmefähigkeit“ identifiziert, da eine eindeutige und leichte Unterscheidung auch abhängig von der kognitiven Belastung des Nutzers ist.

Durch die Domäneneigenschaften gelangt man an die zu den jeweiligen Eigenschaften zugehörigen Prinzipien entsprechend des in Abbildung 5 dargestellten Schemas. Solche Schemata wurden zu mehreren regelmäßig auftauchenden typischen Domänen sowie zu deren Interaktionen untereinander im Vorfeld erstellt. Für jedes zur jeweiligen Domäneneigenschaft ausgewählte Prinzip wird nun beurteilt, ob es in dem aktuellen Kontext relevant ist oder nicht.

Beispiel aus Abbildung 8:

Domäneneigenschaft an Domäne GUI: Lesbarkeit/Verständlichkeit

in Frage kommende MCI-Prinzipien entsprechend Abbildung 5: Selbstbeschreibungsfähigkeit; Avoid visual noise; Form follows function

nach Vergleich mit dem aktuellem Kontext ausgewähltes MCI-Prinzip: Selbstbeschreibungsfähigkeit.

Nach Feststellung der relevanten Prinzipien werden diese im nächsten Schritt, dem „Matching“, mit Entwurfsmustern in Verbindung gebracht (siehe Tabelle 2: Matching zwischen Prinzipien und Entwurfsmustern zu R3.3.1).

Durch die Integration der Prinzipien in die Problem Frames ist es gelungen, den Design Space für Entwurfsmuster zur Lösung dieses Problems von ursprünglich über 100 möglichen Entwurfsmustern in diesem Fall auf 16 potenzielle Kandidaten zu begrenzen. In Tabelle 2 und Tabelle 3 olivgrün hinterlegt sind dabei diejenigen Entwurfsmuster, welche im Verlauf der Umsetzung des Landkarten-Browsers (siehe Kapitel 4) tatsächlich zur Anwendung kommen. Dabei muss es sich nicht immer zwangsläufig um das Entwurfsmuster mit den meisten Übereinstimmungen zu den identifizierten Prinzipien handeln (obgleich dies natürlich ein Indikator für die Eignung eines Entwurfsmusters darstellt), sondern es hängt in letzter Instanz von der Beurteilung des Designers ab, welche Entwurfsmuster aus der Tabelle geeignet erscheinen.

Entwurfsmuster zu R3.3.1

Prinzipien / Entwurfsmuster	Einfachheit	Reduzierung kognitiver Belastung	Selbstbeschreibungsfähigkeit	Present information so that it can be read quickly and correctly
Symbol Coding	X	X	X	X
Clustering	X	X		X
Data Tips		X	X	X
Detail on Demand		X		X
Zoom	X		X	
Slider	X		X	
Simple Zoom	X		X	
Scale			X	X
Panning	X			
Overview + Detail		X	X	X
Icon Menu	X	X	X	X
Paging	X	X	X	X
Collapsible Panels			X	
Liquid Layout				X
Spotlight Effect	X		X	X
Compass Rose			X	X

Tabelle 2: Matching zwischen Prinzipien und Entwurfsmustern zu R3.3.1

R3.3.3

Es muss dem Nutzer ersichtlich sein, wenn sich mehrere Entitäten auf ein und derselben Koordinate befinden.

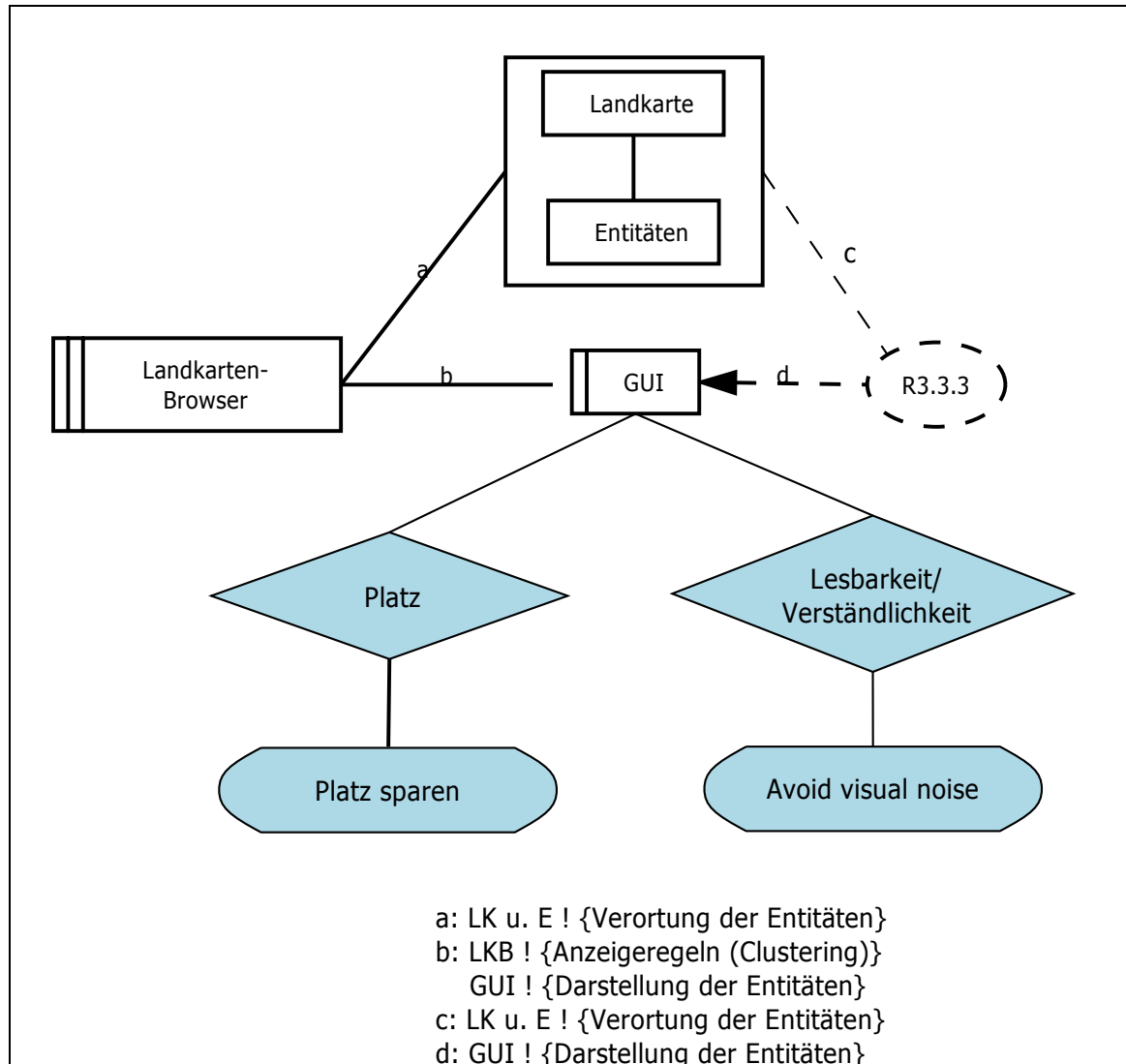


Abbildung 9: Problem Frame zu R3.3.3 inklusive Eigenschaften und Prinzipien

Beim Requirement R3.3.3 steht die Darstellung und somit die Domäne GUI im Vordergrund. Die Eigenschaften „Platz“ und „Lesbarkeit/Verständlichkeit“ erweisen sich hier als relevant. Entsprechend dieser Eigenschaften sind die dazu gehörenden Prinzipien „Platz sparen“ sowie „Avoid visual noise and clutter“ in das Problem Frame (Abbildung 9) integriert worden.

Diese beiden Prinzipien enthalten Referenzen auf mehrere Entwurfsmuster. Beim Matching (Tabelle 3) wurden mehrere geeignete Entwurfsmuster identifiziert, von denen für das Design übernommen wurden (Clustering; Data Tips; Detail on Demand und Paging).

Entwurfsmuster zu R3.3.3

Prinzipien / Entwurfsmuster	Platz sparen	Avoid visual noise and clutter
Symbol Coding	X	
Clustering	X	X
Data Tips	X	X
Detail on Demand	X	X
Overview + Detail	X	
Icon Menu	X	
Paging	X	X
Liquid Layout	X	X
Live Filter		X

Tabelle 3: Matching zwischen Prinzipien und Entwurfsmustern zu R3.3.3

4 Umsetzung der Konzepte in das User Interface

Nachdem im letzten Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** die User Requirements inklusive einer Modellierung mit Problem Frames und Integration von MCI-Prinzipien entwickelt worden sind, befasst sich dieser Abschnitt mit der konkreten Umsetzung der mittels dieses Ansatzes gefundenen Lösungen in Form von MCI-Entwurfsmustern. Dabei wurden die in Tabelle 2 und Tabelle 3 identifizierten Entwurfsmuster für die Umsetzung herangezogen, um für die mit den Problem Frames modellierten Probleme Lösungsansätze zu erarbeiten. Für das hierdurch entwickelte Konzept wurde prototypisch ein Demonstrator auf Basis von HTML und JavaScript erstellt, mit dem das Konzept einerseits erprobt werden kann, und andererseits die entwickelten Ideen plastisch nachvollziehbar gemacht werden.

Die hier vorgestellte Umsetzung des Konzeptes beschreibt die Umsetzung der im letzten Abschnitt 3.2 identifizierten Entwurfsmuster. Im Rahmen der Modellierung des Landkartenbrowsers im THESEUS-Programm wurden weitere Requirements mit entsprechenden Problem Frames und daraus resultierenden Entwurfsmustern modelliert sowie ein Konzept zur Umsetzung erarbeitet, das analog der hier beschriebenen Konzepte entwickelt wurde.

4.1 Darstellung der Entitäten als Symbole

Die verschiedenen Entitäten der Datendomäne werden durch die Symbole in Abbildung 10 verkörpert. Die Entitätentypen sowie die Symbole, welche diese repräsentieren (mit Ausnahme des „Sammelicon“-Symbols), wurden vom Projektpartner neofonie übernommen, so dass eine durchgehend einheitliche Symbolcodierung gewährleistet ist. Somit wurde das Entwurfsmuster „Symbol Coding“², (vgl. Tabelle 2: Matching zwischen Prinzipien und Entwurfsmustern zu R3.3.1) hier konsequent und konsistent angewandt.

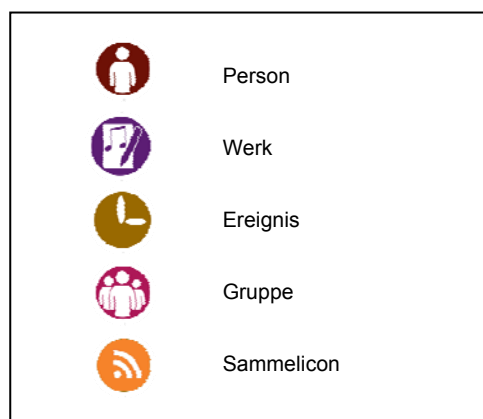


Abbildung 10: Symbol Coding für die Entitäten auf Alexandria

Abbildung 11³ zeigt einen Screenshot, der verdeutlicht, wie das Symbol Coding im Landkartenbrowser umgesetzt wurde. Die verschiedenen Symbole sind rot eingekreist (rote Kreise dienen nur zur Illustration und erscheinen nicht im Landkartenbrowser).

² http://www.patternbrowser.org/code/pattern/pattern_anzeigen.php?4,295,17,0,0,316

³ Kartenmaterial der Abbildungen 11-17 Daten von OpenStreetMap - Veröffentlicht unter CC-BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0/>)

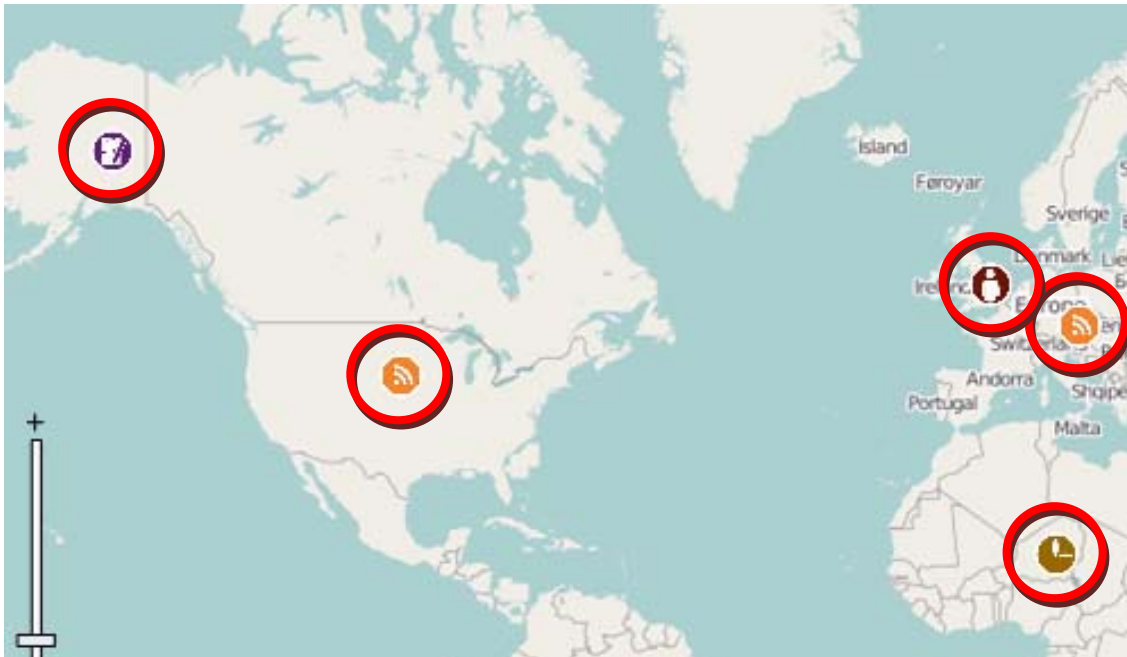


Abbildung 11: Umsetzung von Symbol Coding im Landkartenbrowser

4.2 Darstellung von verdichteten Entitäten

Die Entitäten werden auf der Karte entsprechend ihrer Verortung angezeigt. Dabei sind die Entitäten in der Regel nicht gleichmäßig über die Karte verteilt, sondern es gibt Bereiche, in denen sehr wenige Entitäten dargestellt werden, ebenso wie Bereiche, die sehr viele Entitäten enthalten. Infolgedessen kann das Problem entstehen, dass sich Entitäten überlappen und im schlimmsten Fall sogar komplett überdecken. Dieses Phänomen ist durchaus üblich, etwa wenn zwei Ereignisse auf die gleiche Stadt verortet sind, ohne dass diese genaue Geokoordinaten enthalten.

Zur Umgehung dieses Problems wird an dieser Stelle Gebrauch vom Entwurfsmuster „Clustering“⁴ entsprechend des Requirements R3.3.3 gemacht. Mehrere gedrängte Icons an gleicher Stelle werden in einem Sammelicon zusammengefasst (vgl. Abbildung 12), welches beim Herüberfahren mit der Maus Data Tips⁵ mit Detailinformationen anzeigt (vgl. Abbildung 13).

⁴ http://www.patternbrowser.org/code/pattern/pattern_anzeigen.php?4,273,17,0,0,294

⁵ http://www.patternbrowser.org/code/pattern/pattern_anzeigen.php?4,237,17,0,0,258



Abbildung 12: Clustering zum Anzeigen verdichteter Informationen



Abbildung 13: Data Tips mit näheren Informationen zu den unter einem Sammelicon zusammengefassten Entitäten

Durch Anklicken des Sammelicons erhält man weitere Informationen zu den einzelnen Entitäten (vgl. Abbildung 14). Hierbei handelt es sich um eine Umsetzung des aus dem Requirement R3.3.3 hergeleiteten Entwurfsmusters „Detail on

Demand“.⁶ Man erhält zu den verschiedenen Entitäten neben dem Titel der jeweils angezeigten Entität ein Vorschaubild mit einem kurzen einleitenden Text zur Entität. Durch Anklicken des Titels gelangt man zur Entitätenseite.



Abbildung 14: Details on Demand nach dem Anklicken des Sammelicons

Da unter einem Sammelicon mehrere Entitäten gleichzeitig gebündelt sind, sind weitere Interaktionsmöglichkeiten notwendig, um alle Entitäten auswählen zu können. Das Entwurfsmuster „Paging“⁷ löst in diesem Fall das Problem (vgl. Abbildung 15). Der Nutzer sieht, wie viele weitere Entitäten vorhanden sind und kann zwischen diesen blättern. Darüber hinaus wird dadurch weiterhin Platz gespart, da auch in dieser Ansicht nicht mehrere Entitäten gleichzeitig angezeigt werden müssen.

Als weitere Möglichkeit, um die verschiedenen Entitäten anzuzeigen, dient entsprechend des Vorschlags in Requirement R3.3.1 ein Icon Menu⁸, mit dem es möglich wird, die Informationen auch nach Entitätentyp zu bündeln (vgl. Abbildung 16). Wenn es zu einem Entitätentyp keine anzeigbaren Inhalte an dieser Stelle gibt, wird das entsprechende Icon ausgegraut dargestellt. Der gerade ausgewählte Entitätentyp wird mit einem gelben Kreis markiert.

⁶ http://patternbrowser.org/code/pattern/pattern_anzeigen.php?4,285,17,0,0,306

⁷ <http://www.welie.com/patterns/showPattern.php?patternID=paging>

⁸ <http://www.welie.com/patterns/showPattern.php?patternID=image-menu>



Abbildung 15: Paging zum Betrachten weiterer Entitäten



Abbildung 16: Icon Menu zum Wechsel zwischen verschiedenen Entitätentypen

4.3 Umgang mit Platzbeschränkung

Der Landkartenbrowser ist ein interaktives Element unter mehreren auf einer Entitätenseite von Alexandria. Entsprechend gering ist der Platz, der zur Verfügung steht. Aus diesem Grunde wurden verschiedene Überlegungen zur Platzoptimierung angestellt. Beide der folgenden Ansätze versuchen, den verfügbaren Platz, soweit möglich, auszunutzen. Die Collapsible Pannels⁹ (vgl. Abbildung 17) dienen dazu, den für den Landkarten-Browser vorhandenen Platz bei Bedarf frei zu geben und

⁹ <http://www.welie.com/patterns/showPattern.php?patternID=collapsible-panels>

Anmerkung: Dieses Entwurfsmuster ging aus einem anderen nicht in diesem Report beschriebenen Requirement mit entsprechendem Matching hervor.

ebenso für den Landkarten-Browser Platz zu schaffen, falls andere unbenutzte Elemente auf der Entitätenseite vorhanden sind. Sie sind ein sehr flexibles Instrument, weil man durch einfachen Klick jederzeit die größere Ansicht zurückerhält.

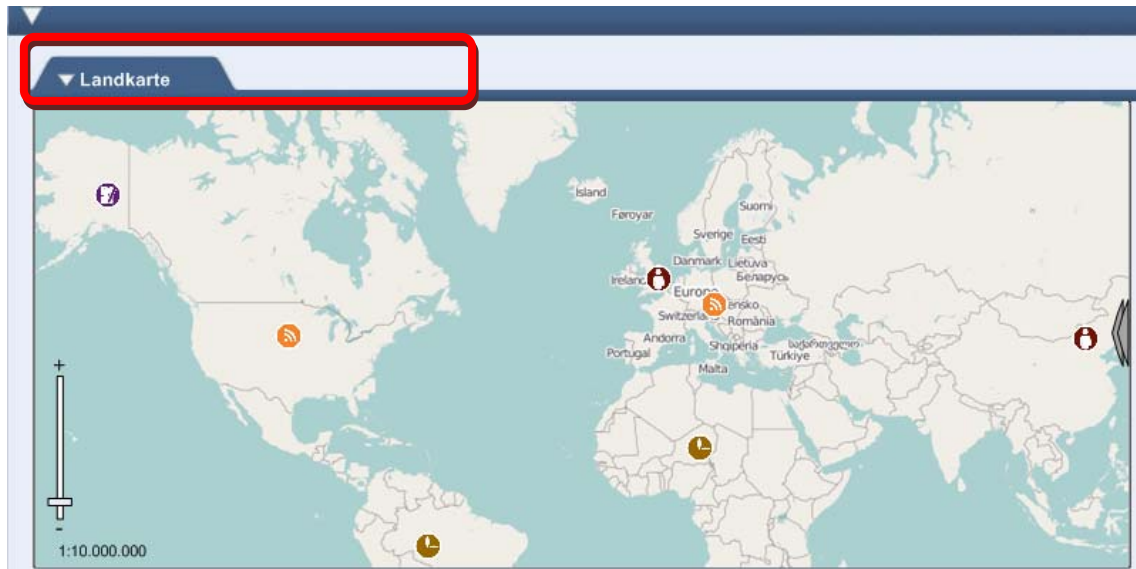


Abbildung 17: Collapsible Panels zum Ein- und Ausklappen von Interaktionselementen

Über ein Liquid Layout¹⁰ kann der Seiteninhalt jederzeit angepasst werden, d. h. wenn die Seite z. B. vergrößert wird, wird auch die Landkarte größer und wenn die Seite verkleinert wird, passt sich entsprechend auch der Landkarten-Browser an den zur Verfügung stehenden Platz an.

5 Fazit

In diesem Report wurde eine integrative Herangehensweise beschrieben, die ausgehend von der Erstellung der Requirements über die Modellierung mittels Problem Frames bis hin zur Einbettung von geeigneten MCI-Entwurfsmustern sowie deren Umsetzung den vollständigen Konzeptionsweg eines Landkartenbrowsers beschreibt, welcher beispielhaft an zwei Problemdiagrammen demonstriert wurde. Diese Herangehensweise ermöglicht es, einerseits die Probleme strukturiert zu erfassen, andererseits passende Entwurfsmuster zu den Problemen zu erhalten, welche dann entsprechend umgesetzt werden können. Der Ansatz ist durchweg musterbasiert und leistet einen wesentlichen Beitrag, um den Auswahlprozess geeigneter Entwurfsmuster systematisch in die Modellierung zu integrieren. Durch Einbeziehung des aktuellen Nutzungskontextes und Einbettung geeigneter Prinzipien in den Problem Frames-Ansatz kann die Anzahl der in Frage kommenden Entwurfsmuster drastisch reduziert werden, was den Auswahlprozess deutlich erleichtert. Dennoch bleibt dem GUI-Designer genügend Freiraum, um Design-

¹⁰ http://www.designinginterfaces.com/Liquid_Layout

Entscheidungen zu treffen. Mittels des hier beschriebenen Prozesses wird es verschiedenen Beteiligten (Requirements Engineer, GUI-Designer, usw.) erleichtert, methodisch die Probleme des User Interface Designs anzugehen und unter Berücksichtigung der Requirements geeignete Lösungsansätze zu entwickeln.

Dieser innerhalb einer laufenden Dissertation entwickelte Ansatz versteht sich als unterstützendes Werkzeug zur GUI-Entwicklung und wurde im Rahmen des THESEUS-Programms im Anwendungsfall ALEXANDRIA erfolgreich erprobt. Die dadurch gesammelten Ergebnisse fließen derzeit in die Weiterentwicklung des Ansatzes ein. Insgesamt entsteht somit ein Beitrag, wertvolles MCI-Erfahrungswissen, welches nicht immer strukturiert vorliegt, methodisch in den Entwicklungsprozess von User Interfaces einfließen zu lassen.

Literatur

- Brown, J. R.; Cunningham, S. (1989): Programming the User Interface -Principles and Examples. New York, USA: John Wiley & Sons Inc.
- Constantine, L. L.; Lockwood, L. A. D. (2002): Instructive Interaction: Making Innovative Interfaces Self-Teaching. Constantine & Lockwood, Ltd. (1(3)).- Technischer Bericht, Expanded original preprint of paper appearing in User Experience
- Cooper, A.; Reimann, R.; Cronin, D. (2007): About Face 3: The Essentials of Interaction Design. 3. Auflage. Indianapolis, USA: John Wiley & Sons Inc.
- DIN EN ISO 9241-11 (1998): Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on usability. International Organisation for Standardization
- Jackson, M. (2001): Problem Frames. Analyzing and structuring software Development problems. Boston, USA: Addison-Wesley
- Mahemoff, M. J.; Johnston, L. J. (1998a): Pattern Languages for Usability: An Investigation of Alternative Approaches. In Tanaka, J. (Hrsg.): Asia-Pacific Conference on Human Computer Interaction (APCHI) 98 Proceedings. IEEE Computer Society, S. 25-31
- Mahemoff, M. J.; Johnston, L. J. (1998b): Principles for a Usability-Oriented Pattern Language. In OZCHI '98: Proceedings of the Australasian Conference on Computer Human Interaction. Washington, DC, USA: IEEE Computer Society, ISBN 0-8186-9206-5, S. 132
- Mayhew, D. J. (1992): Principles and Guidelines in Software User Interface Design. Englewood Cliffs, New Jersey: Prentice Hall PTR
- Oviatt, S. (2006): Human-centered design meets cognitive load theory: Designing interfaces that help people think. In MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia. New York, NY, USA: ACM, S. 871-880

- Patternbrowser (2008) <http://www.patternbrowser.org/>, zuletzt eingesehen am 22.06.2010
- Reeves, L. M.; LAI, J.; Larson, J. A.; Oviatt, S.; Balaji, T. S.; Buisine, S.; Collings, P.; Cohen, P.; Kraal, B.; Martin, J.-C.; MCTaer, M.; RAMAN, T.; Stanney, K. M.; Su, H.; Wang, Q. Y. (2004): Guidelines for Multimodal user interface design. *Commun. ACM*, 47, Nr. 1, S. 57-59
- Smith, C. U.; Williams, L. G. (2002): *Performance Solutions - A Practical Guide to Creating Responsive, Scalable Software*. New York, USA: Addison-Wesley Professional
- Specker, M.; Düchting, M. (2009): Musterbasiertes Usability Engineering am Beispiel eines Landkartenbrowsers. In *Mensch & Computer 2009: Grenzenlos frei!?*, H. Wandtke, S. Kain & D. Struve (Hrsg.), Oldenbourg Verlag, München
- Specker, M.; Pöhler, I.; Krebs, A. (2011a): Ein Konzept zum Explorieren von Inhalten in einem interaktiven Timeline-Browser. In F. Rammig, W. Kern: *C-LAB-Report*, tba.
- Specker, M.; Pöhler, I.; Krebs, A. (2011b): Use Case ALEXANDRIA, Work Package – D.ALEX.3.2, Version 2.0, Siemens IT Solutions and Services GmbH
- Specker, M.; Wentzlaff I. (2006): Pattern-Based Development of User-Friendly Web Applications. In: *MDWE 2006, Workshop zur Konferenz ICWE 2006*, Palo Alto. ACM, Digital Library
- Specker, M.; Wentzlaff I. (2007): Exploring Usability Needs by Human Computer Interaction Patterns. In *TAMODIA'2007, 6th International workshop on TAsk MOdels and DIAGrams*, November 7-9, 2007, Tou-louse, France
- Tidwell, J. (2005): *Designing Interfaces*. O'Reilly Media, Inc., Sebastopol, CA., teilweise online unter <http://designinginterfaces.com/>, zuletzt eingesehen am 22.06.2010
- Welie (2010) <http://www.welie.com/>, zuletzt eingesehen am 22.06.2010