
Otto-von-Guericke Universität Magdeburg



Fakultät für Informatik

Arbeitsgruppe Wirtschaftsinformatik

Studienarbeit

**Visuelle Informationsverarbeitung in einem mit
Microsoft ® Sharepoint ® entwickelten
Wissensmanagementsystems mit Hilfe von
Microsoft ® Silverlight ®**

Themensteller: Dipl. Ing. Knut Köchli
Lintra Solutions GmbH

Betreuer : Prof. Dr. rer. pol. habil. Hans-Knud Arndt

vorgelegt von: Felix Keller

Abgabetermin: 19.10.2009

Inhaltsverzeichnis

ABBILDUNGSVERZEICHNIS.....	V
TABELLENVERZEICHNIS.....	VI
ABKÜRZUNGSVERZEICHNIS.....	VII
1. EINLEITUNG.....	8
1.1. MOTIVATION.....	8
1.2. ZIELSETZUNG.....	10
1.3. GLIEDERUNG.....	11
2. BEGRIFFE UND GRUNDLAGEN.....	11
2.1. WAS VERSTEHT MAN UNTER INFORMATIONSVISUALISIERUNG.....	11
2.2. ZIELE DER INFORMATIONSVISUALISIERUNG.....	12
2.3. ANFORDERUNGEN AN EIN INFORMATIONSVISUALISIERUNGSSYSTEM.....	13
2.3.1. BENUTZERBEREICH.....	14
2.3.2. INTERAKTION.....	15
2.3.3. FUNKTIONALITÄT.....	15
3. EINLEITUNG IN DIE ENTWICKELTE ANWENDUNG.....	17
3.1. KONTEXTUELLE EINORDNUNG DER ARBEIT.....	17
3.2. ANALYSE DER FUNKTIONALEN ANFORDERUNGEN.....	19
3.2.1. DARSTELLUNG DER ELEMENTE UND DEREN BEZIEHUNGEN.....	19
3.2.2. INTERAKTIVE DATENFILTERUNG.....	20
3.2.3. INNOVATIVE DATENMANIPULATION.....	21
3.3. DER SYSTEMENTWURF.....	22
3.3.1. DAS PROMESYS PORTAL.....	22
3.3.2. MICROSOFT® SILVERLIGHT®.....	23
4. IMPLEMENTIERUNG: PROMESYS VISUALIZER.....	26
4.1. AUFBAU DES IMPLEMENTIERTEN SYSTEMS.....	26
4.2. DIE SILVERLIGHTANWENDUNG.....	28

4.2.1.	DER GRAPH.....	28
	Der Layoutalgorithmus	28
	Webservices zur Kommunikation mit Sharepoint.....	30
	Die Nutzerinteraktion zur Analyse.....	31
	Die Nutzerinteraktion zum Modellieren	35
4.2.2.	DER HIERARCHIE EDITOR.....	39
4.2.3.	DIE BEZIEHUNGSMATRIX	40
5.	ZUSAMMENFASSUNG	42
A.	ANHANG	43
	A.1 KONFIGURATIONSDATEI IM XML-FORMAT.....	43
	LITERATURVERZEICHNIS.....	46

Abbildungsverzeichnis

Abbildung 1-1: Ausschnitt des Liniennetzes der MVB	8
Abbildung 2-1: Arbeitsbereiche der Informationsvisualisierung.....	12
Abbildung 2-2: Referenzmodell des klassischen Informationsvisualisierungsablaufes	13
Abbildung 3-1: kontextuelle Einordnung der Arbeit	17
Abbildung 4-1: Systemstruktur aus Nutzersicht	26
Abbildung 4-2: WebPartEditor zur initialen Konfiguration der Anwendung.....	27
Abbildung 4-3: Webpart zum Hosten der Silverlightanwendung.....	27
Abbildung 4-4: Die Silverlightanwendung im aktivem Tab „Graph“	28
Abbildung 4-5: Ein Webservice gestattet den Zugriff auf eine Anwendungskomponente unter Verwendung von gewöhnlichen Internettechniken.	30
Abbildung 4-6: überfüllter Graph, Zahlen zeigen jeweils den Zugang zur Filterungsalternative	31
Abbildung 4-7: aktualisierter Graph mit eingeblendeten Hierarchiebaum	32
Abbildung 4-8: eingeblendete verknüpfte Elemente.....	33
Abbildung 4-9: Ergebnis einer weiteren Auswahl im Modus „Verknüpfte Elemente“ zeigt transitive Beziehungen	34
Abbildung 4-10: Ad Hoc Anzeige der Kontextinformationen einer relationalen Beziehung..	35
Abbildung 4-11: Formular zum Erstellen neuer Elemente	36
Abbildung 4-12: Attributliste mit hinterlegten Attributen	37
Abbildung 4-13: Einträge in Hilfsliste zum Speichern der Attributwerte.....	37
Abbildung 4-14: Setzen neuer Verknüpfungen mit Drag and Drop	38
Abbildung 4-15: Mittels Drag and Drop lässt sich die Hierarchie verändern.....	40
Abbildung 4-16: Beziehungsmatrix zum Pflegen von Beziehungen	41

Tabellenverzeichnis

Tabelle 1-1: Liniennetz der MVB in tabellarischer Form.....	9
Tabelle 3-1: Komponenten der Silverlight ® -Architektur.....	25

Abkürzungsverzeichnis

MVB	Magdeburger Verkehrsbetriebe
Promesys	Prozesskettenorientiertes Regelkreismodell für ein nachhaltiges robustes Design mechatronischer Systeme
Str.	Straße
vgl.	vergleiche
WSS	Windows SharePoint Services

1. Einleitung

Informationsmanagementsysteme sind heutzutage aus Organisationen nicht mehr wegzudenken. Sämtliche Aspekte sind abhängig von einer effektiven und effizienten Informationsverarbeitung. Ein Ziel dabei ist es möglichst viel Mehrwert aus den erfassten Informationen in Form von Wissen zu generieren. Auf diese Aufgabe haben sich Wissensmanagementsysteme spezialisiert. Eine wichtige Frage besteht darin, wo und wann entsteht überhaupt Wissen? Dabei kommt man zu dem Ergebnis, dass Wissen dann entsteht, wenn Zusammenhänge innerhalb komplexer Systeme erfasst und verstanden werden, woraus eine Handlungsrelevanz gefolgert werden kann, welche ein wichtiges Kriterium in der Abgrenzung von Informationen und Wissen darstellt. Erfolgsfaktoren sind neben der Informationsaufbereitung, die Darstellung und die Selektivität der Informationen. Diese Aspekte werden innerhalb der Informationsvisualisierung betrachtet.

1.1. Motivation

Um die Motivation dieser Arbeit einzuleiten, soll ein Beispiel aus dem Alltag eingefügt werden. Die folgende Abbildung zeigt einen Ausschnitt aus dem Liniennetz der Magdeburger Verkehrsbetriebe.

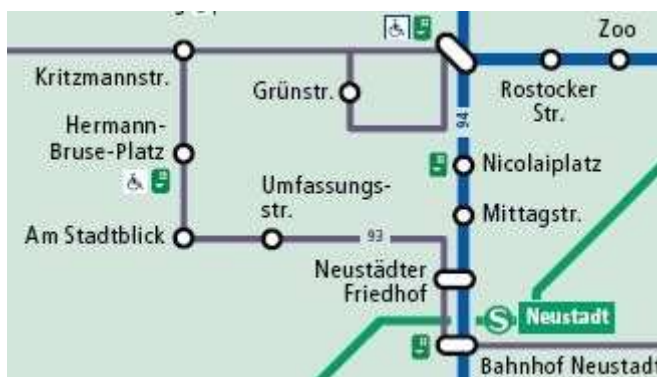


Abbildung 1-1: Ausschnitt des Liniennetzes der MVB

In der Abbildung sieht man die Namen der Haltestellen, welche als Daten betrachtet werden können. Da diese Daten in einen Kontext eingeordnet werden können, entstehen durch diese Daten Informationen. Bieten die Informationen die Möglichkeit Erkenntnisse beim Betrachter zu erzeugen, welche für diesen Handlungsrelevant werden, wird neues Wissen generiert.

Beschränken wir den Informationsgehalt darin, dass aufgezeigt wird, von welcher Haltestelle man andere Haltestellen erreichen kann, bildet die folgende Tabelle exakt die gleichen Informationen ab, wie der Ausschnitt des Fahrplans aus Abbildung 1-1.

ID	Haltestelle	Zugeordnete Haltestellen
Haltestelle_1	Kritzmannstr.	Hermann-Bruse-Platz, Grünestr., Rostockerstr.
Haltestelle_2	Hermann-Bruse-Platz	Kritzmannstr., Am Stadtblick
Haltestelle_3	Am Stadtblick	Umfassungsstr., Hermann-Bruse-Platz
Haltestelle_4	Umfassungsstr	Neustädter Friedhof, Am Stadtblick
Haltestelle_5	Bahnhof Neustadt	Agnetenstr., Neustädter Friedhof
Haltestelle_6	Mittagstr.	Bahnhof Neustadt, Nicolaiplatz
Haltestelle_7	Nicolaiplatz	Mittagstr, Rostockerstr.
Haltestelle_8	Grünstr.	Kritzmannstr, Rostockerstr.
Haltestelle_9	Rostockerstr	Grünstr., Zoo
Haltestelle_10	Zoo	Rostockerstr, Koferstr.

Tabelle 1-1: Liniennetz der MVB in tabellarischer Form

Obwohl die Informationen, welche sich in der Tabelle befinden vollständig und konsistent sind, ist es weitaus schwieriger daraus Wissen zu generieren als mit Hilfe des Liniennetzes. Der Betrachter wird deutlich länger brauchen die einzelnen Haltestellen in Zusammenhang zu setzen und damit seine Entscheidung über die richtige Route zu treffen als mit Hilfe der grafischen Darstellung.

Vor der Schwierigkeit mit Hilfe korrekt abgebildeter Informationen Wissen zu generieren stehen viele Informationssysteme. Bei steigender Anzahl an Daten, Abhängigkeiten und Wechselwirkungen können die abgebildeten Informationen schnell so komplex werden, dass es für den Nutzer immer schwieriger wird, einen Zugang zu den für ihn relevanten Informationen zu finden um neue handlungsentscheidende Erkenntnisse zu gewinnen. Innerhalb des Forschungs- und Entwicklungsprojektes Promesys ist ein auf Microsoft Sharepoint basierendes Wissensmanagementsystem entstanden. Promesys beinhaltet ein

Konzept zur strukturierten Erfassung aller Aspekte in komplexen mechatronischen Systemen entlang deren gesamten Lebenszyklusses. Dieses Erfassungskonzept wurde mit Hilfe des Wissensmanagementsystems Microsoft® Sharepoint® prototypisch im Verlauf des Entwicklungsprojektes umgesetzt. Es wurde erfolgreich ein bedienbares System erschaffen, mit dessen Hilfe alle Informationen verarbeitet werden können, die zur Strukturierung der Elemente der Systeme benötigt werden.

Damit war es möglich sämtliche Gliederungsaspekte und Interdependenzen der Elemente aufzunehmen und zu verwalten. Trotz der Erfassung der Abhängigkeiten der Elemente untereinander, stellte sich heraus, dass es für den Benutzer des Systems relativ schwierig war solche Abhängigkeiten und Zusammenhänge zu erkennen und zu kommunizieren. Ähnlich ergeht es dem Betrachter des Fahrplans in tabellarischer Form beim Finden der richtigen Route. Um das Ziel von Promesys: Die Erschaffung eines nachhaltigen, robusten Designs realisieren zu können, stellt diese Erkennung jedoch eine wichtige Voraussetzung dar. Eine interaktive Visualisierungskomponente sollte bei der Überwindung dieser Schwierigkeiten Unterstützung bieten und somit das Erschaffen neuen Wissens auf Grundlage der vorher erfassten Informationen effizienter machen.

1.2. Zielsetzung

Am Ende dieser Arbeit soll ein Informationsvisualisierungssystem entstanden sein, welches in das mit Sharepoint entwickelte Wissensmanagementsystem vollständig integriert ist. Dabei wird untersucht, ob die verschieden umgesetzten Visualisierungs- und Interaktionstechniken in der Lage sind, die Entwickler mechatronischer Systeme und alle in die Qualitätssicherung involvierten Personen bei der Erfassung der Komplexität und dem Modellieren solcher Systeme zu unterstützen. Das entwickelte Informationsvisualisierungssystem muss die Möglichkeit bieten die Struktur der mechatronischen Systeme, in Form von Klassifikationen, Abhängigkeiten und Wechselwirkungen der Elemente, aus denen ein System besteht, aufzuzeigen. Für den Benutzer müssen Interaktionsmöglichkeiten realisiert werden, um individuelle Sichten auf das System zu erstellen. Denn nur wenn man es schafft der Informationsflut mit Such- und Filtermechanismen zu begegnen, kann ein Mehrwert durch das zu entwickelnde Visualisierungssystem generiert werden.

1.3. Gliederung

Nachdem in die Thematik motivierend eingeleitet wurde und die Formulierung der Zielsetzung abgeschlossen ist, wird zunächst auf allgemeine, grundlegende Aspekte der Informationsvisualisierung eingegangen. Dabei werden Begriffe definiert, um einen Einstieg in die Thematik zu gewähren. Darauf aufbauend soll auf den Zweck und die Motivation der Informationsvisualisierung eingegangen werden. Anschließend werden die Anforderungen an ein Informationsvisualisierungssystem erläutert. Im weiteren Verlauf spielt die Vorstellung des Informationsvisualisierungssystem eine wichtige Rolle, welches mit Hilfe von Microsoft® Silverlight® entwickelt wurden ist. Am Schluss wird die Arbeit zusammengefasst und eine Nachbetrachtung hinsichtlich der Zielsetzung und den Anforderungen an ein Informationsvisualisierungssystem durchgeführt.

2. Begriffe und Grundlagen

2.1. Was versteht man unter Informationsvisualisierung

Informationsvisualisierung ist ein Prozess der Transformation von Informationen in eine grafische Darstellungsform, um dem Nutzer ein Beobachten der Informationen zu ermöglichen. Informationsvisualisierung bezeichnet auch ein Forschungsgebiet, das sich mit den computer-unterstützten Methoden zur graphischen Darstellung großer Datenmengen auseinandersetzt, die nicht unmittelbar mit physikalischen Zuständen oder Vorgängen verknüpft sind. Die bildlichen Darstellungsmethoden sollen helfen, diese Daten auszuwerten und aus ihnen neue Erkenntnisse zu gewinnen, vgl. [Die07, S.1]

Allgemein unterscheidet man zwischen den Begriffen „Information“ und „Wissen“. So wird auch die Wissensvisualisierung selbst von der Informationsvisualisierung unterscheiden. Diese wird als ein Prozess der Wissensübermittlung, welcher sich graphischer Mittel jeglicher Art bedient, definiert, vgl. [Bur05]

Ein weiteres Tätigkeitsfeld der Informationsvisualisierung ist die Mensch-Computer-Interaktion, worunter hier Folgendes zu verstehen ist: Die Kognition wird mit Hilfe von computergestützten interaktiven Darstellungstechniken von abstrakten Daten erweitert.

Zusammenfassend versteht man heute unter Informationsvisualisierung alle Konzepte, Methoden und Werkzeuge zur visuellen Darstellung von Informationen aus Datenbanken, digitalen Bibliotheken oder anderen großen Dokumentsammlungen.

Eine wichtige Eigenschaft der Informationsvisualisierung ist ihre Interdisziplinarität. Die folgende Abbildung zeigt die verschiedenen Arbeitsbereiche, die für die Informationsvisualisierung von Bedeutung sind.

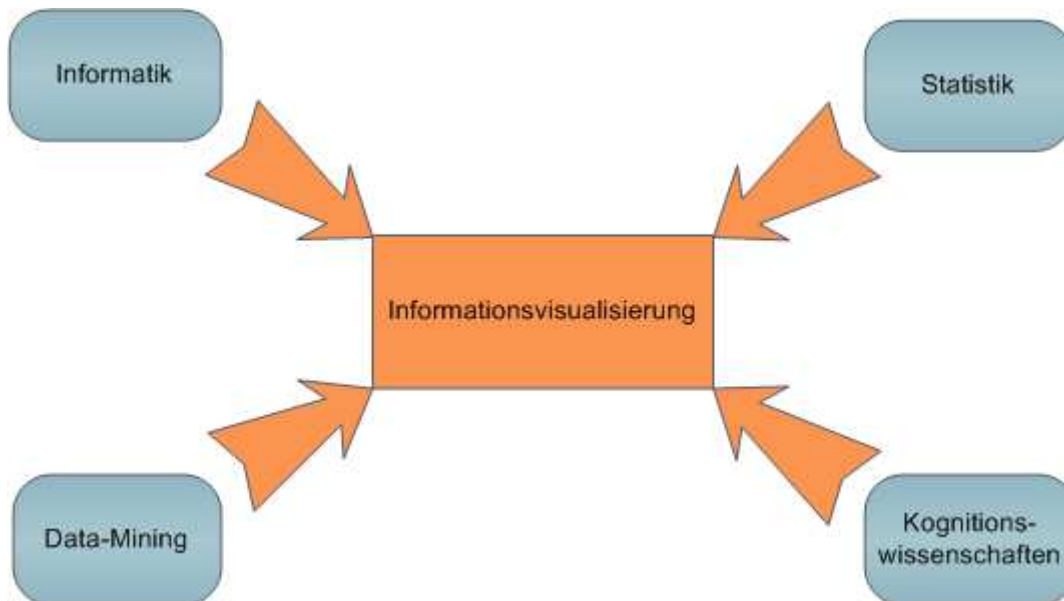


Abbildung 2-1: Arbeitsbereiche der Informationsvisualisierung

2.2. Ziele der Informationsvisualisierung

Primäres Ziel der Informationsvisualisierung ist es, abstrakte Informationen aus beliebigen, vorrangig textbasierten, Informationssystemen strukturell aufzubereiten und grafisch darzustellen. Dabei werden traditionelle Informationssysteme durch die visuelle Repräsentation von Informationen um folgende Aspekte erweitert:

- Das Auffinden spezifischer Informationsdomänen in großen Datenbeständen
- Das Erkennen von Relationen, Strukturen oder Trends in unstrukturiert erscheinenden Informationsmengen
- Verschiedene Sichten auf identische Datenbestände

Das folgende Referenzmodell stellt den klassischen Ablauf der Informationsvisualisierung dar.

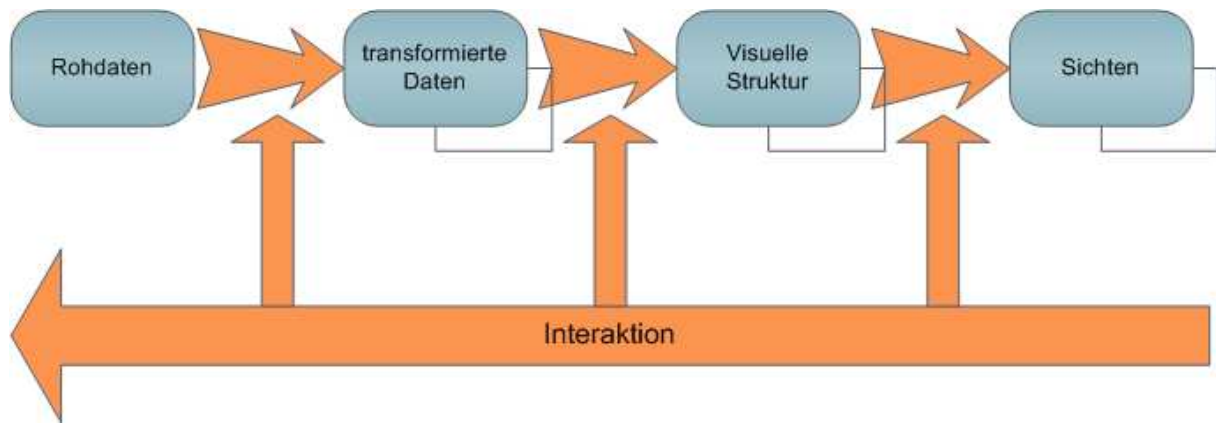


Abbildung 2-2: Referenzmodell des klassischen Informationsvisualisierungsablaufes

Das Modell geht von einem Bestand an Rohdaten aus, welche in einer proprietären Struktur vorliegen. Das können z.B. relationale Tabellen oder XML sein. Es kann sich dabei aber auch um völlig unstrukturierte Daten handeln. Um von diesen Daten zu einer Visualisierung zu gelangen, müssen zunächst diverse Transformationen mit den Daten durchgeführt werden. Einerseits ist es möglich, dass abgeleitete Größen (z.B. Aggregationen) dargestellt werden sollen, andererseits müssen die Daten normalisiert werden, um eine Abbildung auf visuelle Eigenschaften möglich zu machen. Das Ergebnis dieses ersten Schrittes ist eine Menge transformierter Daten in einheitlicher Struktur. Im nächsten Schritt werden nun diese Daten auf die gewünschte visuelle Struktur abgebildet (Mapping). Auf dieser Struktur kann nun eine Menge von benutzerspezifischen Sichten generiert werden, die dem Benutzer die Navigation durch die Darstellung erlaubt. Eine Interaktion des Benutzers kann sich dann auf mehrere Stufen des Abbildungsprozesses auswirken. Er kann seine Sicht auf die Daten anpassen, die visuelle Struktur ändern oder sogar auf die Datentransformation Einfluss nehmen. Die zyklischen Pfeile in der Abbildung verweisen auf die Tatsache, dass es sich hier um einen iterativen Prozess handelt, der zunächst mehrfach wiederholt werden kann, bevor der nächste Schritt durchgeführt wird.

2.3. Anforderungen an ein Informationsvisualisierungssystem

Informationsvisualisierungskomponenten müssen mehr leisten als nur bunte Bilder darstellen zu können. Um herauszufinden ob die Systeme den Benutzeranforderungen genügen, wurden Anforderungs- /Bewertungskataloge entwickelt. Barberena Najarro arbeitete in ihrer Dissertation Anforderungs- und Bewertungskriterien für Informationsvisualisierungssysteme aus, welche im folgenden Abschnitt erläutert werden, vgl. [Naj03]

Sie unterteilt ein Informationsvisualisierungssystem in 3 Komponenten:

- Benutzeroberfläche
- Interaktion
- Funktionalität

2.3.1. Benutzeroberfläche

Die Kriterien der Benutzeroberfläche befassen sich mit dem statischen Aufbau der Oberfläche. Dabei ist wichtig, dass der Benutzer die ausführbaren Aktionen schnell erkennen kann und nicht aufwendig erlernen muss

Strukturiertheit

Unter Strukturiertheit versteht man, dass der Nutzer schnell einen Überblick über die Gesamtheit der Visualisierung erhält, Zusammenhänge verstehen und damit die Informationen effektiv erschließen kann

Erkennbarkeit

Unter Erkennbarkeit werden folgende Fragen zusammengefasst:

- Wie sind die Elemente der Arbeitsoberfläche gestaltet, unterstützen oder beeinträchtigen sie die Wahrnehmung des Benutzers?
- Wie gut sind die Objekte auffindbar und selektierbar?
- Gibt die Visualisierung eine Rückmeldung, welches Objekt gerade betrachtet wird?

Lokalität

Die Ausführung einer Aktion und die Wahrnehmung durch den Benutzer sollten immer möglichst eng gekoppelt sein. Deshalb ist es wichtig, dass die häufigsten Funktionen der Informationsvisualisierung, unabhängig von der Position des Betrachters im direkten Zugriff erreichbar bleiben. Ergebnisse von Aktionen sollten möglichst unmittelbar am Ort der Ausführung erfolgen, damit der Benutzer sie durch seine Position und Blickrichtung wahrnehmen kann. Das Kriterium der Lokalität ist besonders in komplexen, dreidimensionalen Räumen von Bedeutung.

Orientierung

Damit der Benutzer nicht die Orientierung verliert, sollten z. B. Zielzustände, ausführbare Operationen oder erreichbare Positionen möglichst vollständig visualisiert werden. Dabei muss vor allem der aktuelle Ort- oder Zeitpunkt im Verhältnis zum Ganzen dargestellt werden.

2.3.2. Interaktion

Interaktion beinhaltet die Mechanismen, mit denen der User das System nach seinen Wünschen verändern bzw. seine Suche verfeinern kann.

Wechselwirkung

Häufig auftretende Interaktionsmechanismen in Informationsvisualisierungen sind:

- Rotation der Ansicht
- Veränderung des Kamerastandpunktes
- Zoomen in ein interessantes Gebiet hinein für Details oder aus einem Gebiet heraus für einen besseren Überblick
- Auswählen von Elementen, um zusätzliche Informationen zu erlangen
- Ausblenden von Elementen zur besseren Übersicht

Ausführungsminimalität

Die Interaktionsmechanismen sollten es dem Benutzer möglichst einfach und schnell erlauben ein bestimmtes Element in den Fokus der Betrachtung zu bringen. In diesem Zusammenhang ist zu untersuchen, ob die Aktionen dabei verständlich sind, sich eventuell gegenseitig bedingen, und ob sich mehrere Handlungssequenzen zu einer Aktion zusammenfassen lassen.

2.3.3. Funktionalität

Hier werden die Verfahren in Hinblick auf die gebotenen Möglichkeiten zur Untersuchung von Dokumentenmengen verglichen. Einerseits wird die grafische Darstellung des Informationsraumes, andererseits die Technik zur Bearbeitung der Elemente innerhalb des Informationsraumes betrachtet.

Darstellung

Bei großen Datenmengen kann ein Rechner noch immer an seine Grenzen stoßen. Daher ist es sinnvoll, die Ergebnisdatenmenge schon vor der grafischen Darstellung zu reduzieren, um die Ausgabegeschwindigkeit zu erhöhen.

Erweiterbarkeit

Verschiedene Informationsvisualisierungen benötigen unterschiedliche Darstellungsformen und auch unterschiedliche Benutzer haben individuelle Darstellungspräferenzen. Deshalb sollte das Informationsvisualisierungssystem Visualisierungen, Filter und Ausgabelemente erweiterbar sein

Details

Welche Möglichkeiten die Informationsvisualisierung zur genauen Betrachtung von Datenmengen bietet, wird in diesem Kriterium bewertet.

3. Einleitung in die entwickelte Anwendung

In diesem Kapitel soll in die Umsetzung der entwickelten Anwendung eingeführt werden. Dazu wird zunächst eine Übersicht aufgezeigt, in welchen Kontext sich die Arbeit einordnen lässt.

3.1. Kontextuelle Einordnung der Arbeit

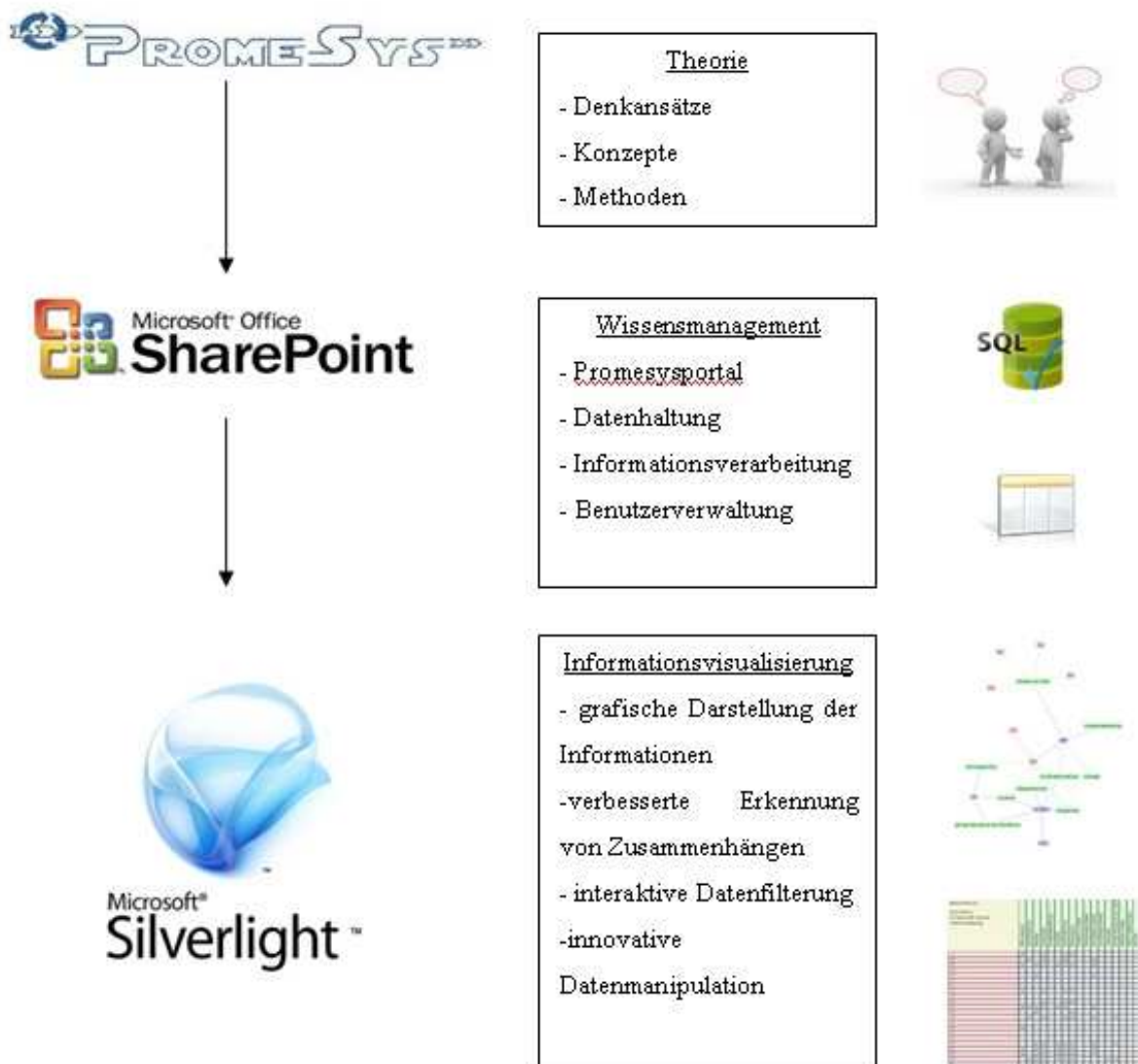


Abbildung 3-1: kontextuelle Einordnung der Arbeit

Mit Hilfe der oberen Grafik wird der Kontext dieser Arbeit erläutert. Wie bereits erwähnt, entstand die Entwicklung des Informationsvisualisierungssystems im Rahmen des Forschungs- und Entwicklungsprojektes Promesys, dessen Konzepte, Methoden und Denkweisen hier als Ausgangspunkt gesehen werden.

Dabei steht die Entwicklung und prototypische Realisierung eines ganzheitlichen Konzeptes zur Erhöhung der Zuverlässigkeit mechatronischer Systeme, insbesondere in den frühen Phasen des Lebenszyklusses, im Vordergrund. Hierbei kommt es darauf an, den durch die unterschiedlichen Teildisziplinen geprägten und mit zahlreichen Schnittstellen behafteten Designprozess der Systeme grundlegend zu verbessern, um deren Zuverlässigkeit signifikant zu erhöhen. Eine Kernaufgabe besteht darin, die zuverlässigkeitsrelevanten Anforderungen systematisch mit den beteiligten Komponenten, Funktionen und Prozessen zu verknüpfen, vgl [Bert08, S. 15]. Das Konzept und die methodischen Ansätze konnten vor Beginn dieser Arbeit in ein Wissensmanagementsystem auf Basis von Microsoft® Sharepoint® überführt werden. Dieses werde ich im Kapitel 3.3. näher beschreiben.

Innerhalb des entstandenen Wissensmanagementsystems, welches hier als Promesysportal bezeichnet wird, können alle benötigten Informationen verteilt verarbeitet werden. Außerdem bietet es die Möglichkeit Nutzerspezifische Sichten zu realisieren und Benutzer zu verwalten, welche am Design eines mechatronischen Systems und dessen Qualitätssicherung involviert sind.

Mit Hilfe des Promesysportals wird der Zugang zum Wissen, welches im Laufe des Lebenszyklusses eines mechatronischen Systems entsteht, jeder autorisierten Person ermöglicht. Im Laufe des Projektes wurde jedoch die Erkenntnis gewonnen, dass zwar die Anforderungen an Vollständigkeit, Genauigkeit und Widerspruchsfreiheit erfüllt werden können, es jedoch Verbesserungspotential in der Präsentation der Informationen gibt. Dieses begründet sich in sofern, dass es relativ aufwendig ist, komplexe semantische Zusammenhänge von Elementen zu erkennen und es unkomfortabel ist, solche inhaltlichen Interdependenzen anderen zu präsentieren. Grundsätzlich ist es möglich Verknüpfungen und Beziehungen von Elementen innerhalb des Portals zu pflegen und nachzuvollziehen. Um dies jedoch zu erreichen, kann eine Traversierung durch eine Vielzahl von Listen und Elementen nötig sein. Eine Übersicht über vielschichtige Zusammenhänge ist dabei nur schwer zu erreichen.

An diese Stelle tritt das Informationsvisualisierungssystem. Ziel eines solchen Systems ist es, dem Nutzer den Zugang zu den Informationen zu erleichtern.

Wichtig dabei sind zunächst eine verbesserte Präsentation der Elemente und deren Beziehungen untereinander. Daneben soll eine interaktive Datenfilterung das Analysieren der Informationen erleichtern. Eine weitere Anforderung wird das komfortable und intuitive Modellieren von mechatronischen Systemen darstellen.

Zur Umsetzung des Visualisierungssystems wird Microsoft® Silverlight® verwendet:

Silverlight ® ist eine browser- und plattformunabhängige Implementierung auf Basis des .NET Framework, mit dem RIA (Rich Internet Application) - Anwendungen für das Web erstellt werden können. Der englische Begriff "rich" steht hierbei für die reichhaltigen Möglichkeiten an Benutzerinteraktionen (bspw. Drag and Drop) und dem vielseitigen Einsatz von Medien innerhalb der Anwendung (bspw. Videos, Animationen). RIA - Anwendungen sind clientbasierte Anwendungen. Dadurch ist es möglich komplexe Berechnungen durchzuführen, die auf dem PC des Nutzers ausgeführt werden und nicht den Server belasten. Ein weiterer Vorteil der Clientanwendung sind kurze Antwortzeiten bei Nutzerinteraktionen, da keine Kommunikation zum Server stattfinden muss, wenn die Anwendung einmal auf den Client geladen wurde. Somit wird die Server- und Netzwerklast durch die Clientarchitektur reduziert. Sollte es jedoch nötig sein, dass die Anwendung während ihrer Laufzeit mit dem Server kommunizieren muss, können dafür Webservices verwendet werden. Nähere Erläuterungen zu dieser Kommunikationsform und welche Rolle sie innerhalb des entwickelten Informationsvisualisierungssystems, wird in einem späteren Abschnitt erläutert. Im Kapitel 3.3 wird ein Überblick über die Architektur Silverlights ® gegeben..

3.2. Analyse der funktionalen Anforderungen

Zu Beginn der Entwicklung muss zunächst geklärt werden, welche Funktionen sich aus den Anforderungen ableiten lassen.

3.2.1. Darstellung der Elemente und deren Beziehungen

Bevor mit der Funktionsanalyse begonnen werden kann, ist es zunächst wichtig, die Abgrenzung von Systemelementen zu Elementen zu verstehen. Die Systemelemente im Promesysansatz sind wie bereits erwähnt Anforderungen, Funktionen, Komponenten, Prozesse, Prüfungen und Probleme. Jedes Systemelement wird innerhalb des Portals durch eine Sharepointliste repräsentiert. Als Elemente verstehen wir Einträge innerhalb dieser Listen, welche Instanzen der Systemelemente widerspiegeln.

Eine Kernanforderung besteht darin, alle ins Portal eingepflegten Elemente und deren Beziehungen untereinander darzustellen. Betrachtet man die Elemente unabhängig voneinander, also ohne die Beziehungen zu untersuchen, muss der Nutzer zunächst in der Lage sein, relativ schnell zu erkennen, zu welchem Systemelement ein Element gehört, sprich die Visualisierung der Klassifikation Erreicht werden soll diese Anforderung durch die Darstellung der Elemente mit unterschiedlichen Farben und Formen, welche zwischen

Elementen eines Systemelementes identisch sind und sich zu Elementen anderer Systemelemente unterscheiden. Vier unterschiedliche geometrische Formen sollen zur visuellen Klassifikation verwendet werden: Rechteck, Ellipse, abgerundetes Rechteck und Hexagon. Ein Element welches durch eine geometrische Form dargestellt ist, wird im Folgenden als Knoten bezeichnet. Da ein Knoten nur die Möglichkeit bietet Kurzbezeichnung oder Titel des Elementes zu beinhalten, ist es notwendig, zu jedem Knoten bei Bedarf alle Informationen des Elementes in einem Kontextmenü anzuzeigen.

Wenn wir uns die Darstellung der Beziehungen der Elemente untereinander anschauen, müssen wir zwei Arten von Beziehungen unterscheiden: Hierarchische Beziehungen und relationale Beziehungen. Jedem Element wird ein Elternelement zugeordnet, so dass für das gesamte Systemelement ein Hierarchiebaum aufgespannt werden kann. Um diese Art der Beziehung darzustellen soll eine Baumvisualisierung umgesetzt werden, die es dem Nutzer erlaubt, einen Überblick über die hierarchischen Beziehungen der Systemelemente eines mechatronischen Systems zu gewinnen.

Die zweite Art der Beziehungen, die relationale Beziehung, geht über die Grenzen der Systemelemente hinaus, d.h. Elemente aus verschiedenen Systemelementen, können semantisch miteinander verknüpft sein. In der Anwendung ist es also erforderlich, dass Elemente, welche über geometrische Formen dargestellt werden, mit Hilfe von Kanten verbunden werden. Die Gesamtheit der Verknüpfungen stellt dann ein Netz aus verbundenen Elementen dar. Da es das Portal vorsieht, relationale Beziehungen mit Kontextinformationen zu versehen, ist es erforderlich, dass sich der Nutzer bei Bedarf diese Informationen über die Kanten, wie bei einem Knoten über ein Kontextmenü, anzeigen lassen kann.

3.2.2. Interaktive Datenfilterung

Neben der Darstellung des semantischen Netzes, ist es wichtig, dass der Nutzer mit diesem auch interagieren kann, um eine individuelle Sicht auf die Daten zu erhalten, wodurch der Zugang zu den Informationen erleichtert werden soll. Neben dem besseren Auffinden einzelner Elemente, werden mit Hilfe der Benutzerinteraktion, einzelne Teilnetze aufgebaut, deren Ausgangspunkte selektierte Elemente sind. Für die Funktionalität der Anwendung bedeutet dies zunächst, dass verschiedene Suchmöglichkeiten umgesetzt werden müssen. Eine Möglichkeit für den Nutzer muss darin bestehen, sich über den Hierarchiebaum des Systemelementes zu den gewünschten Elementen „durchzuhangeln“ und innerhalb des Baumes Elemente zu selektieren. Eine Alternative dazu ist ein Suchformular, mit welchem der Nutzer durch Eingabe einer Zeichenfolge Elemente finden kann. Die anschließende

Selektion der Elemente erfolgt manuell im Hierarchiebaum und im Elementnetz selbst oder wird automatisch nach Auffinden über das Suchformular ausgeführt und dem Nutzer zu erkennen gegeben. Nach der Selektion können alle selektierten oder nicht selektierten Elemente ausgeblendet werden, so dass die Knoten- und Kantenanzahl verringert wird und eine neue Datensicht entsteht.

Weiterhin wird durch Nutzerinteraktion angestrebt, dass man sich ausgehend von einem bestimmten Element alle mit diesem verbundenen Elemente anzeigen lassen kann. Dies bietet die Möglichkeit der semantischen Traversierung ausgehend von einem gewählten Element. Somit sollen neben hierarchischen - und relationalen - auch transitive Beziehungen schnell aufgelöst und nachvollzogen werden können. Durch Umsetzung dieser Funktion wird es möglich sein, multidimensionale Anfragen komfortabel aufzulösen, bei denen man mit den herkömmlichen Listenformularen schnell an seine Grenze stößt.

3.2.3. Innovative Datenmanipulation

Da es sich für den Nutzer als relativ aufwendig erwiesen hat die Daten ausschließlich über Listenformulare zu manipulieren, ist eine weitere Anforderung an das System alternative, innovative Bearbeitungsmöglichkeiten zu bieten. Zum Erstellen neuer Elemente müssen neue Knoten innerhalb des Systems angelegt werden und sämtliche Eigenschaften hinterlegt werden. Darüber hinaus muss ein bestehendes Element über den korrespondierenden Knoten veränderbar sein. Zum Erstellen neuer Beziehungen bietet es sich an, Knoten durch Kanten miteinander zu verbinden. Im gleichen Arbeitsschritt soll es dann möglich sein der neuen Relation Kontextinformationen anzuhängen. Geplant ist es, das Erstellen neuer Kanten mit Hilfe von Drag and Drop zu realisieren. Dabei wählt der Nutzer den Quellknoten aus und zeichnet von diesem eine Verbindung zum Zielknoten. Der Arbeitsschritt soll nach Eingabe der Kontextinformationen über ein zusätzliches Eingabeformular abgeschlossen werden. Neben dem kreieren neuer Beziehungen über das Zeichnen von Kanten, muss es dem Nutzer möglich sein, Beziehungen über ein Matrixformular zu setzen. Hintergrund dieser Anforderung ist es, innerhalb kürzester Zeit eine Vielzahl an Beziehungen zu setzen. Eine solche Matrix würde sich neben der Manipulation natürlich auch zur übersichtlichen Darstellung von Interdependenzen der Elemente eignen. Eine weitere Art der Datenbearbeitung ergibt sich aus dem spezifischen Zusammenhang zwischen den Systemelementen und Probleme. Um das Design eines mechatronischen Systems robuster zu gestalten, ist es essentiell, Probleme der bestehenden Systemelemente zu speichern. Diese werden wie auch Funktionen, Anforderungen, Komponenten und Prozesse als Einträge in

einer Sharepointliste abgelegt und können über Nachschlagfelder mit den Systemelementen verknüpft werden. Der Wunsch der Projektpartner ist es, die Beziehungen zwischen Systemelementen und Problemen mit Hilfe eines Quadrantenformulars zu bearbeiten. Über Drag and Drop sollen Elemente schnell und intuitiv zu Problemen zugeordnet werden. Die Hierarchie innerhalb eines Systemelements ist bei der Zuordnung unbedingt zu beachten und muss bei der Umsetzung der Nutzerschnittstelle berücksichtigt werden, d.h. wenn der Nutzer ein Element zu einem Problem hinzufügt, müssen automatisch alle Unterelemente rekursiv mit dem gleichen Problem in Beziehung gebracht werden.

3.3. Der Systementwurf

In diesem Abschnitt wird der Entwurf des entwickelten Systems vorgestellt. Dabei werden sowohl vorhandene Komponenten, wie dem auf Windows® Sharepoint® Services entwickelten Promesysportal und die Silverlight® Architektur, als auch die zu entwickelnden Komponenten vorgestellt und miteinander in Zusammenhang gebracht.

3.3.1. Das Promesysportal

Da das Promesysportal auf Microsoft® Windows® Sharepoint®, im Folgenden nur noch als Sharepoint bezeichnet, entwickelt wurde, wird es eine kurze Übersicht über diese Plattform geben.

Die SharePoint Technologie unterstützt sämtliche Aspekte des Umgangs mit Informationen (Speichern, Bereitstellen, Suchen / Finden, Integration und Konsolidierung. Um funktionale Erweiterungen zu realisieren, können Webparts verwendet werden. Das hier zu implementierende Informationsvisualisierungssystem wird Teil eines Webparts sein (siehe Kapitel 3.3.3). Die aktuelle SharePoint-Generation grenzt sich in zwei Produkte ab, einerseits den Windows® Sharepoint® Services 3.0 (WSS) und dem Microsoft® Office Sharepoint® Server 2007 (MOSS). Die WSS 3.0 sind eine kostenfreie Ergänzung der Microsoft® Windows-Serverplattform® (Windows 2003 und spätere) und stellen grundlegende Funktionen bereit, um Informationsverwaltung - und Zusammenarbeitslösungen zu erstellen. Zur Umsetzung des Promesysportals wurde ausschließlich die WSS 3.0 verwendet, da es als ausreichend erachtet wurde, dass diese zur Optimierung der Zusammenarbeit eines überschaubaren Teams genügen. Für einen unternehmensweiten Einsatz, bei dem die Informationsbereitstellung für einen großen Benutzerkreis realisiert werden muss, wird das

Potential des MOSS interessant sein. Insbesondere bei der Integration von Daten aus Fremdsystemen (z.B. Warenwirtschaft, Buchhaltung, allgemeine externe Quellen), sollte man sich für die Variante des Office SharePoint Servers entscheiden. Zusammenfassend kann man sagen, dass Microsoft ® Windows ® Sharepoint ® Services 3.0 eine leistungsfähige und komplexe Technologie des Windows Server ist, welche ein integriertes Portfolio von Diensten für die Kommunikation und Zusammenarbeit anbietet mit dem Ziel, Menschen, Informationen, Prozesse und Systeme zu integrieren. Nachdem ein kurzer Einstieg in SharePoint gegeben wurde, steht im Folgenden die konzeptionelle Umsetzung des Promesys Datenmodells im Fokus der Betrachtung. Jedes Element wird nur einmal modelliert und als Objekt in der Datenbank bzw. auf dem SharePoint Server gespeichert. Jede Komponente, jede Anforderung, jede Funktion, jeder Prozess, jede Prüfung und jedes Problem eines mechatronischen Systems wird genau einmal beschrieben, unabhängig der Anzahl der Verwendung. Jedes dieser Elemente findet sich im Portal als ein Listeneintrag wieder, vgl. [Bert04, S. 108] Über Listenfelder können die Eigenschaften der Elemente gespeichert werden (vergleichbar mit Attributen in relationalen Datenbanken). Für die Darstellung semantischer Beziehungen sind besonders Nachschlagefelder interessant. In diesen Feldern werden andere Elemente referenziert, welche aus der gleichen oder einer andere Listen stammen können. Damit wird es möglich sowohl hierarchische als auch relationale Beziehungen abzubilden. SharePoint-Listen und deren Einträge bilden die zentrale Datenquelle für das Informationsvisualisierungssystem. Lese- und Schreibetechniken zur Erfassung und Manipulation der Daten stellen eine Grundvoraussetzung bei der angestrebten visuellen Darstellung und Modellierung mechatronischer Systeme dar. In Kapitel 4.2.1. werden diese Techniken näher erläutert. Zur Entwicklung des Visualisierungssystems, welches auch als eine alternative Benutzerschnittstelle innerhalb des Portals betrachtet werden kann, die es dem Nutzer erlaubt, einen innovativen Zugang zu den Informationen, sowohl in der Darstellung als auch in der Modellierung, zu erhalten, wird die Entwicklungsplattform Microsoft ® Silverlight ® verwendet .

3.3.2. Microsoft ® Silverlight ®

Silverlight ® dient nicht nur zur ansprechenden Darstellung von Web- und Medieninhalten, sondern liefert auch eine leistungsstarke Entwicklungsplattform zur Umsetzung portierbarer, plattformübergreifender vernetzter Anwendungen, welche verschiedene Daten, Dienste und Technologien integrieren. Besonders im Bereich der Benutzerfreundlichkeit bietet Silverlight ® im Vergleich zu herkömmlichen Webanwendungen ein hohes Verbesserungspotential.

Die Silverlight ® -Plattform setzt sich aus zwei Hauptteilen sowie einer Installations- und Aktualisierungskomponente zusammen, Die folgende Tabelle stellt diese Zusammenstellung dar und beschreibt die einzelnen Komponenten.

Komponente	Beschreibung
Kernpräsentationsframework	Dienste und Komponenten, die auf die Benutzeroberfläche und die Benutzerinteraktion ausgerichtet sind. Dazu gehören Benutzereingaben, einfache Benutzeroberflächen-Steuererelemente für den Einsatz in Webanwendungen, Medienwiedergabe, Verwaltung digitaler Rechte, Datenbindung sowie Präsentationsfeatures wie Vektorgrafiken, Text, Animation und Bilder. Des Weiteren ist die Extensible Application Markup Language (XAML) zum Angeben von Layouts enthalten.
.NET Framework für Silverlight	Ein Ausschnitt des .NET Frameworks mit Komponenten und Bibliotheken. Wie der Datenintegration, erweiterbaren Windows-Steuererelementen, Netzwerkfunktionen, Basisklassenbibliotheken, Garbage Collection und Common Language Runtime (CLR).
Installations- und Aktualisierungsprogramm	Ein Installations- und Aktualisierungssteuerelement, das die Installation der Anwendung für Erstbenutzer vereinfacht und anschließend automatische Updates zur Verfügung stellt.

Tabelle 3-1: Komponenten der Silverlight ® -Architektur

4. Implementierung: Promesys Visualizer

4.1. Aufbau des implementierten Systems

Das folgende Modell gibt einen Überblick über die Strukturierung des Systems aus der Sicht des Anwenders.

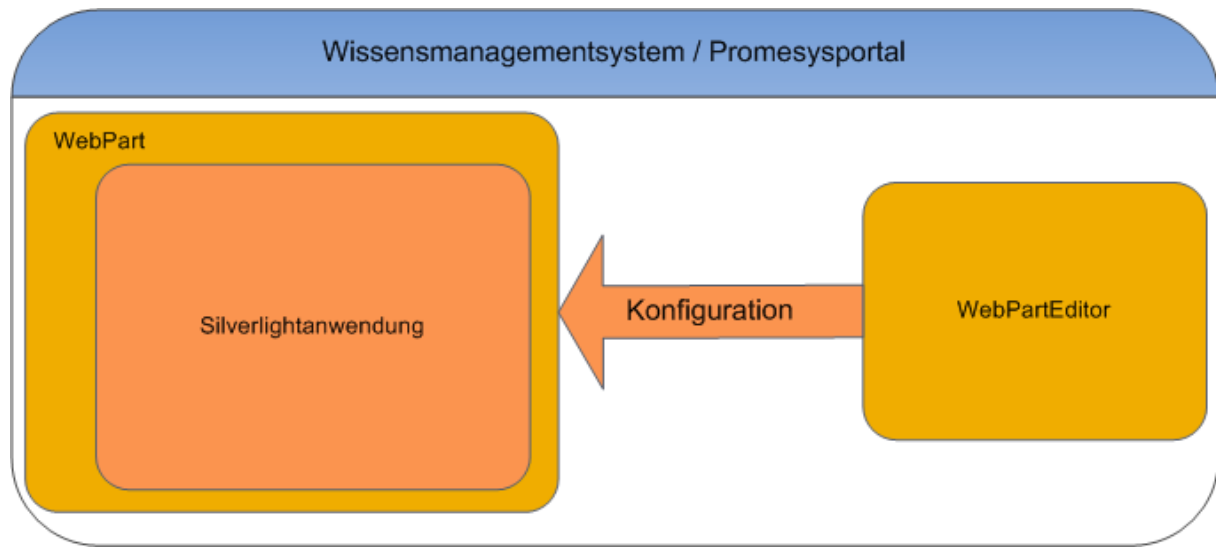


Abbildung 4-1: Systemstruktur aus Nutzersicht

Eine Anforderung bei der Entwicklung war es, die Inputdaten, welche in Form von gefüllten Sharepointlisten vorliegen, vor Laufzeit der Silverlightanwendung, anpassbar zu machen. Dazu wird ein Sharepoint WebPartEditor verwendet, mit Hilfe dessen persistente Parameter eines Webparts gesetzt werden können. Die Anpassung beinhaltet zunächst die Auswahl der Listen, aus denen die Daten gelesen werden, eine Festlegung des Feldes, welches für die Beschriftung eines Knotens verantwortlich ist und die Wahl von Form und Farbe eines dargestellten Elementes. Weiterhin wählt der Nutzer das Feld aus, auf dessen Basis die Eltern-Kind-Beziehung abgebildet ist, welche zur Generierung der Hierarchie innerhalb eines Systemelementes benötigt wird. Die Nachschlagfelder, welche die Basis der relationalen Beziehungen sind, werden automatisch ermittelt. Die gesamte initiale Konfiguration wird in Form eines XML Fragmentes strukturiert und nach Ende der Laufzeit dauerhaft innerhalb des Sharepointsystems gespeichert. Die folgende Abbildung zeigt die Oberfläche des

entwickelten Konfigurationsteils innerhalb des WebPartEditors.



Abbildung 4-2: WebPartEditor zur initialen Konfiguration der Anwendung

Nachdem die initiale Konfiguration, welche in der Regel nur beim ersten Starten der Anwendung festgelegt werden muss, abgeschlossen ist, können zur Laufzeit des Webparts, aber noch vor dem Starten der Silverlightanwendung, weitere Eingabeparameter gesetzt werden. Die nächste Abbildung zeigt die Benutzeroberfläche des Webparts, in welchem dann per Knopfdruck („Draw Graph“) die Silverlightanwendung gestartet wird. Vor dem Starten kann der Nutzer die Elemente, welche in der Silverlightanwendung angezeigt werden, über die Sharepointsicht filtern und bestimmte Inhaltstypen ausblenden.

PromesysVisualizer3

Visualizer (Attribute), HierarchyViewer, HierarchyEditor, Probleme.



Abbildung 4-3: Webpart zum Hosten der Silverlightanwendung

Nachdem sowohl die initiale Konfiguration abgeschlossen ist und die Eingabeparameter festgelegt wurden, startet der Nutzer die Visualisierungsoberfläche, die als Silverlightanwendung implementiert wurde und den Hauptteil der praktischen Arbeit ausmacht. In den folgenden Kapiteln wird auf Konzepte bei der Umsetzung sowohl aus Entwickler- als auch aus Nutzersicht eingegangen. Aus Sicht des Entwicklers ist zunächst der Layoutalgorithmus zum Zeichnen des semantischen Netzes interessant. Wesentliche Silverlightkonzepte und Steuerelemente sowie die Verwendung von Webservices zur Kommunikation mit dem Promesysportal spielen bei der Entwicklung eine große Rolle. Aus Nutzersicht soll auf die Bedienung der Anwendung eingegangen werden. Dazu gehören Erläuterungen zum Umgang mit dem System und Intentionen der Funktionen.

zur Physik sind, um die Knoten eines Graphen zu positionieren. Dabei wird der Graph als ein System von Körpern betrachtet, auf welche Kräfte einwirken. Der Algorithmus sucht dabei nach einer Konfiguration, in der die Gesamtsumme der Kräfte, die auf die Knoten bzw. Körper wirken, minimiert wird.

Der Spring Embedder Algorithmus verwendet eine Kombination aus Federn und elektrischen Kräften. Dabei repräsentieren die Federn die Kanten des Graphen. Die Knoten werden als sich elektrisch abstoßende Körper modelliert.

Sei $p_v = (x_v, y_v)$ der Vektor zum Knoten $v \in V$. Mit $\|p_v - p_u\|$ wird die Länge des Differenzvektors $p_v - p_u$ (der euklidische Abstand der Punkte (x_v, y_v) und (x_u, y_u)) bezeichnet. Außerdem ist $\overrightarrow{p_u p_v}$ der Einheitsvektor $\frac{p_v - p_u}{\|p_v - p_u\|}$, der von p_u nach p_v zeigt.

Die abstoßende Kraft wird mit folgender Formel berechnet:

$$f_0(p_u, p_v) = \frac{c_c}{\|p_v - p_u\|^2} * \overrightarrow{p_u p_v}$$

wobei c_c eine Konstante ist, die angibt wie stark sich zwei Knoten abstoßen.

Die Federkraft lässt sich mit der Gleichung

$$f_1(p_u, p_v) = -c_s (\|p_v - p_u\| - l_{uv}) * \overrightarrow{p_u p_v}$$

berechnen, wobei die Konstante c_s die Federkraft und l_{uv} die natürliche Länge der Feder angibt.

Die Kraft f_0 in x-Richtung lautet:

$$F_{0_x} = \frac{c_c}{(p_{v_x} - p_{u_x})^2 + (p_{v_y} - p_{u_y})^2} * \frac{p_{v_x} - p_{u_x}}{\sqrt{(p_{v_x} - p_{u_x})^2 + (p_{v_y} - p_{u_y})^2}}$$

Analog lassen sich die Werte für F_{0_y} , F_{1_x} und F_{1_y} berechnen.

Die Kraft, die auf einen Knoten $u \in V$ einwirkt ergibt sich aus folgender Formel:

$$F(u) = \sum_{(u,v) \in V \times V} f_0(u,v) + \sum_{(u,v) \in E} f_1(u,v)$$

Die Berechnung des Layouts erfolgt dann nach folgendem Algorithmus: Es wirkt auf alle Knoten eine Kraft, welche sich nicht im „Kräftegleichgewicht“ befinden. Damit das System „entspannt“ wird, werden die Knoten iterativ in Richtung der auf sie einwirkenden Kraft verschoben. Zum Zeitpunkt t wirkt auf einen Knoten v die Kraft $F_t(v)$

.Nach Berechnung aller Kräfte $F_t(v)$ für alle $v \in V$ zum Zeitpunkt t , wird jeder Knoten v um $\delta * F_t(v)$ verschoben. Die Konstante $\delta (0 < \delta < 1)$ verhindert, dass Knoten übermäßig weit verschoben werden.

Webservices zur Kommunikation mit Sharepoint

Zur Kommunikation zwischen dem Frontend, der Silverlightanwendung und dem Backend, den Windows® Sharepoint® Services werden Webservices eingesetzt. Diese ermöglichen sowohl Lese – als auch Schreibeoperationen auf den Sharepointlisten, welche von der Clientanwendung über die grafische Benutzeroberfläche ausgelöst werden. Der Begriff Webservice wird folgendermaßen definiert: Ein Webservice ist eine über ein Netzwerk zugängliche Schnittstelle zu Anwendungsfunktionen, die mit Hilfe von Standardtechniken des Internets realisiert wird, vgl. [SnTiKu03] Abbildung 4-5 skizziert diese Technologie.

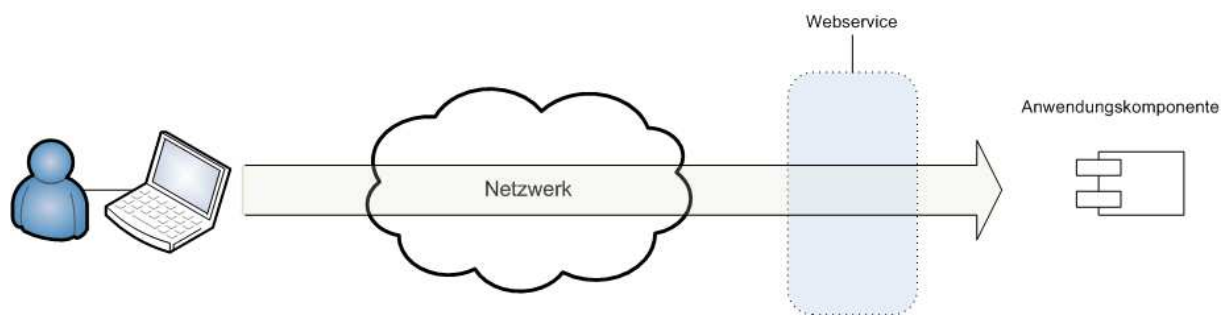


Abbildung 4-5: Ein Webservice gestattet den Zugriff auf eine Anwendungskomponente unter Verwendung von gewöhnlichen Internettechniken.

Wird mit einer Anwendung mit Hilfe einer Kombination von Protokollen wie HTTP, XML oder SMTP über ein Netzwerk kommuniziert, spricht man von einem Webservice. Wie in Abbildung 4-5 dargestellt, kann ein Webservice als Schnittstelle zwischen einer Anwendungskomponente und einem Benutzer beschrieben werden. Um die Plattform –und Programmiersprachen-spezifischen Einzelheiten vom eigentlichen Anwendungsaufwurf zu trennen, wirkt sich die Schnittstelle als Abstraktionsschicht aus. Solch eine standardisierte Schicht hat den Vorteil, dass jede Sprache, die Webservices unterstützt, auf die Funktionen der Anwendung zugreifen kann. Übertragen auf unser System kann als Benutzer die Silverlightanwendung betrachtet werden, die auf einem Client ausgeführt wird. Die WSS stellen dabei die Anwendungskomponente dar. Die Funktionen, die der Client benötigt, sind Lese – und Schreibeoperationen auf Sharepointlisten. Zur Ausführung dieser Funktionen wird der Webservice „Lists.asmx“ verwendet.

Die Nutzerinteraktion zur Analyse

Im Hauptfenster des aktiven Tabs „Graph“ (siehe Abbildung 4-6) beobachtet der Nutzer zunächst die Ausrichtung von Knoten und Kanten auf Basis des Layoutalgorithmus. Manuel kann der Algorithmus nun über eine Schaltfläche gestoppt werden. Nachdem die Abbruchkriterien erreicht sind (siehe Layoutalgorithmus) stellt sich auch automatisch ein Ende der Positionsaktualisierung der Knoten ein.

Der Nutzer hat zwar die Möglichkeit vor dem Starten der Anwendung eine individuelle Selektion der zu visualisierenden Elemente vorzunehmen, jedoch ist es nicht ungewöhnlich, dass durch die hohe Anzahl an angezeigten Elementen und deren Verbindungen die Übersicht über das System verloren geht. Grund dafür kann sein, dass der Nutzer durch fehlendes Wissen nicht in der Lage ist, eine Vorauswahl zu treffen, auf diese Vorauswahl bewusst verzichtet oder die Anzahl der nicht gewählten Elemente sehr gering ist. Der Mehrwert eines überfüllten Netzes für den Anwender bewegt sich dann gegen Null.

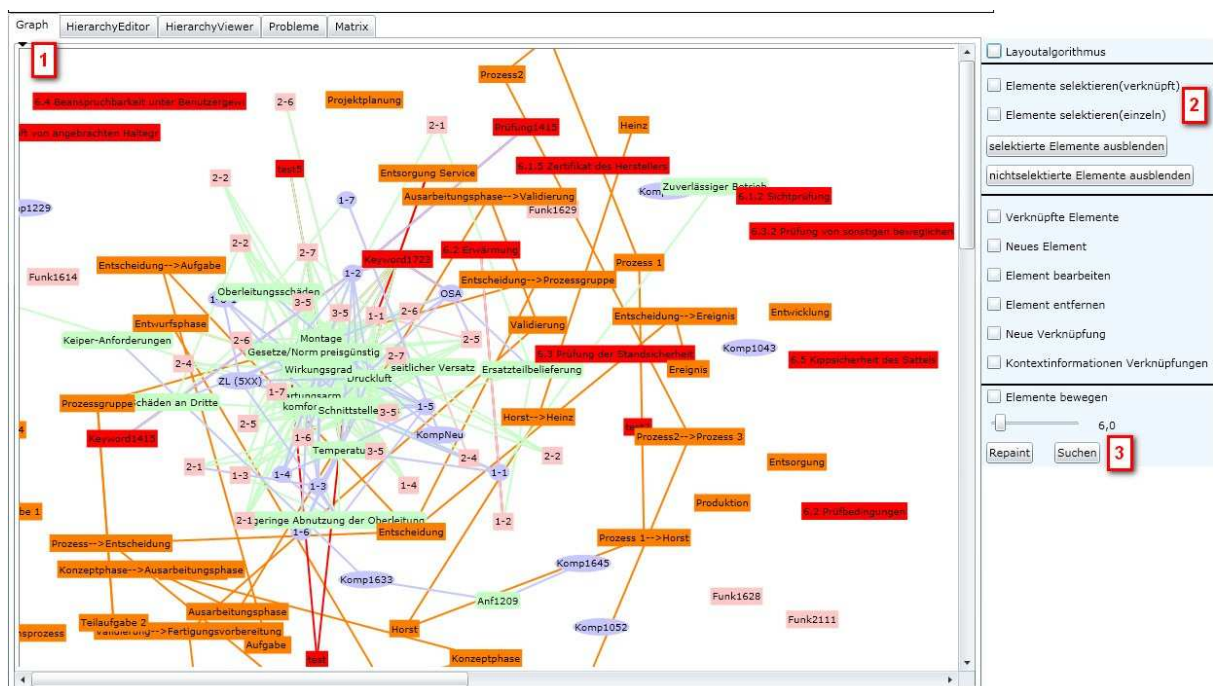


Abbildung 4-6: überfüllter Graph, Zahlen zeigen jeweils den Zugang zur Filterungsalternative

Es ist demnach unumgänglich, dass die Anwendung dem Nutzer verschiedene Alternativen bietet auch während der Laufzeit mit dem Graphen zu interagieren, um die Übersichtlichkeit wiederherzustellen, damit aus den gespeicherten Informationen mit Hilfe der Visualisierung Wissen generiert werden kann. Zunächst bietet die Anwendung die Möglichkeit, die

Systemelemente hierarchisch zu filtern. Dazu kann ein Hierarchiebaum über Schaltfläche 1 (siehe Abbildung 4-7) eingeblendet werden. Checkboxes an den einzelnen Knoten dienen der Selektion und Deselektion der Elemente. Die folgende Abbildung zeigt den eingeblendeten Hierarchiebaum, in dem ausschließlich 6 Elemente selektiert wurden. Über die Schaltfläche "Repaint" kann dann der Graph hinsichtlich der durchgeführten Selektion / Deselektion aktualisiert werden. In der Abbildung sieht man dann nur 6 Elemente.

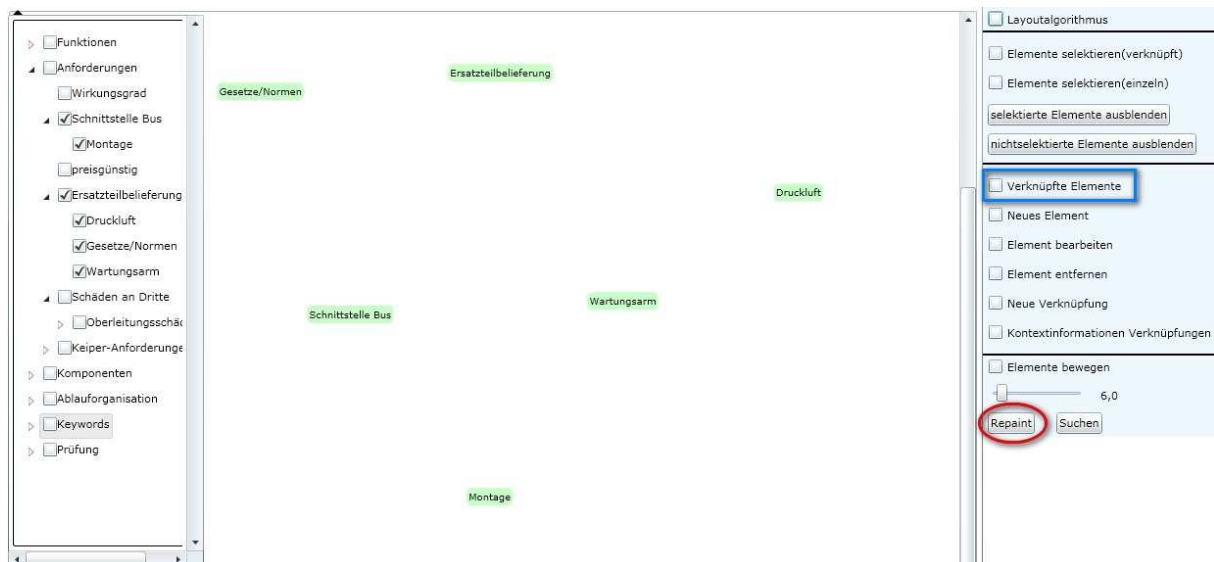


Abbildung 4-7: aktualisierter Graph mit eingeblendeten Hierarchiebaum

Ziel bei dieser Herangehensweise ist es, dass der Nutzer die Möglichkeit hat, die Visualisierung auf bestimmte Gruppen und Module zu beschränken. Neben der Hierarchischen Filterung besteht die Option das visualisierte System syntaktisch zu Filtern. Dazu wird das Suchformular verwendet, welches über die Schaltfläche „Suchen“ (siehe Abbildung 4-6, Punkt 3) aufgerufen wird.

Mit diesem Formular können Elemente selektiert werden, welche die vom Benutzer eingegebene Bezeichnung innerhalb des Titels, der Kurzbezeichnung oder der Beschreibung besitzen.

Die Dritte Möglichkeit Elemente zu selektieren ist die freie manuelle Auswahl innerhalb des Graphen. Elemente können dabei einzeln ausgewählt oder aber zusammen mit ihren verknüpften Elementen selektiert werden. Eine Mehrfachauswahl ist dabei über die STRG-Taste zu erreichen. Wie bei der Selektion durch das Suchformular können nun wahlweise alle selektierten oder alle nicht selektierten Elemente ausgeblendet werden. (siehe Abbildung 4-6, Punkt 3) Neben der Filterung kann sich der Nutzer gezielt Informationen zu den bestimmten Elementen in einem Kontextmenü anzeigen lassen. Zu diesen Informationen gehören die Eigenschaften des Elementes, wie beispielsweise die Beschreibung, physische oder monetäre

Eigenschaften. Eine Kernaufgabe in einem semantischen Netz besteht darin Beziehungen zu anderen Elementen zu entdecken. Diese Beziehungen sind bei der Anzeige vollständiger Systeme zwar vorhanden, jedoch durch eine zu hohe Anzahl an Knoten und Kanten für den Nutzer nicht greifbar. Die Anwendung bietet im Modus „verknüpfte Elemente“ (siehe Abbildung 4-7, blauer Rahmen) die Funktion, dass alle Elemente angezeigt werden, die mit dem gewählten Element verbunden sind und damit wieder sichtbar werden, wenn Sie auf Grund der vorher durchgeführten Filterung ausgeblendet wurden. Damit steht dem Anwender ein Werkzeug zur Verfügung mit dem er ausgehend von einem Element alle direkten und transitiven Beziehungen bis zu einer beliebigen Tiefe auflösen kann. Wechselwirkungen und Kausalketten in mechatronischen Systemen können somit deutlich besser aufgedeckt werden als über Sharepointlisten. Abbildung 4-8 und Abbildung 4-9 zeigen die Ergebnisse einer exemplarischen Interaktion im Modus „Verknüpfte Elemente“. Die jeweils eingerahmten Knoten wurden ausgewählt, was zum Einblenden der verknüpften Knoten führt. Diese waren Bestandteil der exemplarischen Filterung (Siehe Abbildung 4-8)

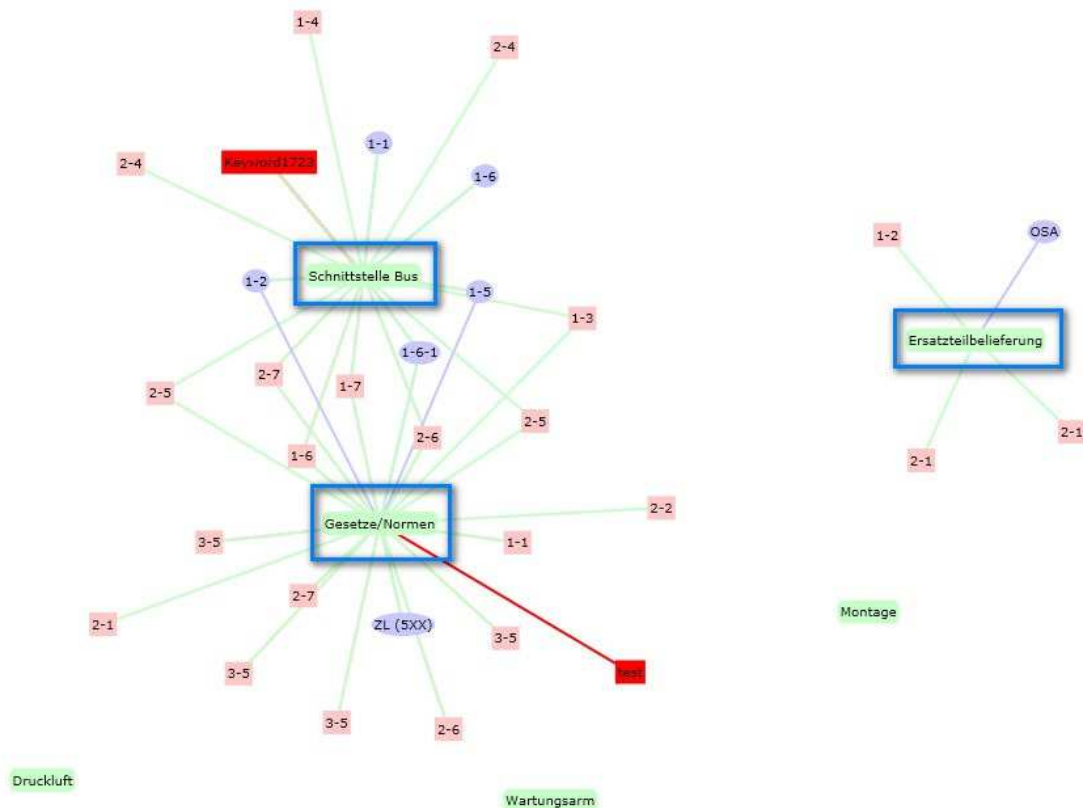


Abbildung 4-8: eingblendete verknüpfte Elemente

Im nächsten Schritt wurden die Elemente „OSA“ und „2-2“ ausgewählt. In Abbildung 4-9 ist das Resultat der Auswahl zu sehen.

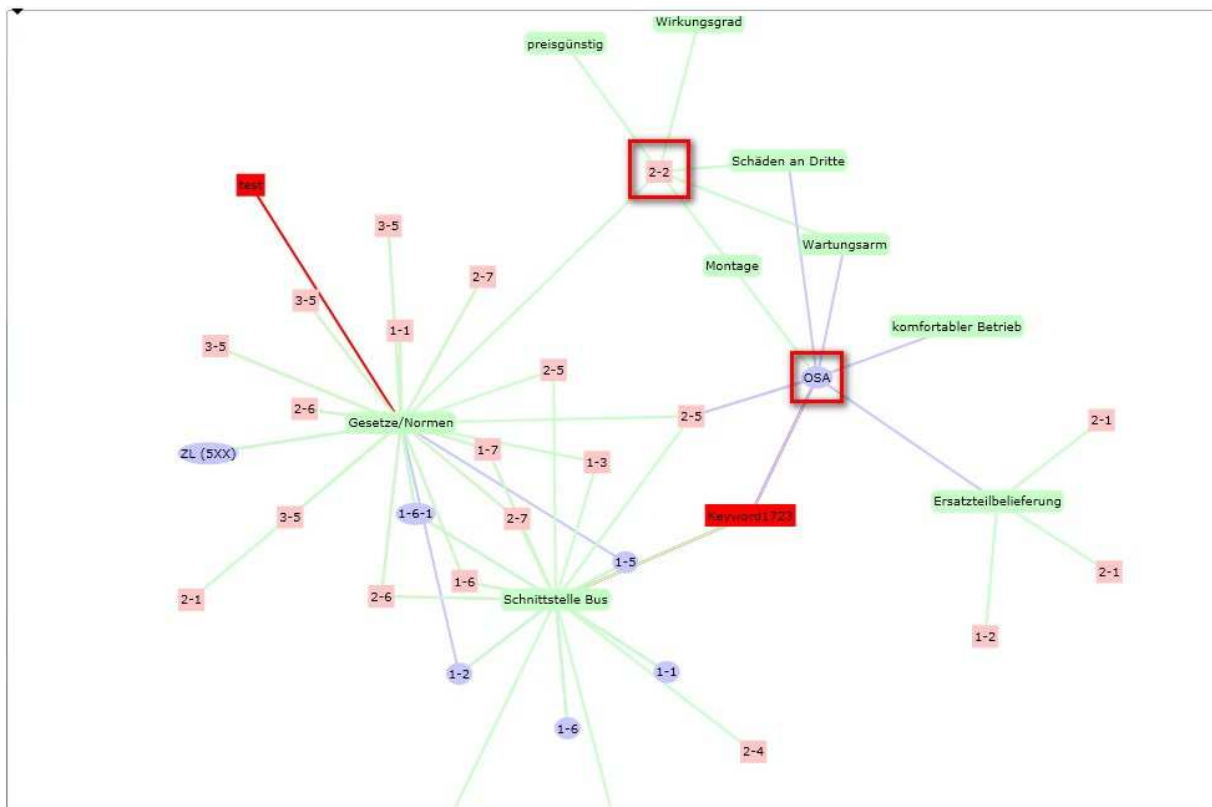


Abbildung 4-9: Ergebnis einer weiteren Auswahl im Modus „Verknüpfte Elemente“ zeigt transitive Beziehungen

Ein weiterer Aspekt des Promesyskonzeptes ist die Tatsache, dass relationale Beziehungen mit Kontextinformationen versehen werden können, so dass nicht nur die Information abgelegt ist, dass zwei Elemente miteinander in Beziehung stehen, sondern auch in welchem Kontext diese Relation entstanden ist. Wie bereits erwähnt, werden Relationen in Form von Kanten innerhalb des Graphen visualisiert. Daher bietet es sich an, die zugehörigen Kontextinformationen auch in diesen abzulegen. Da ein permanentes Anzeigen der Beschreibung der Beziehung aus Platzgründen nicht in Frage kommt, müssen sie ad hoc angezeigt werden. Dazu wurde eine Funktion implementiert, welche die Kontextinformationen, beim Bewegen der Maus über eine Kante anzeigt.

Die folgende Abbildung veranschaulicht exemplarisch diese Funktionalität.

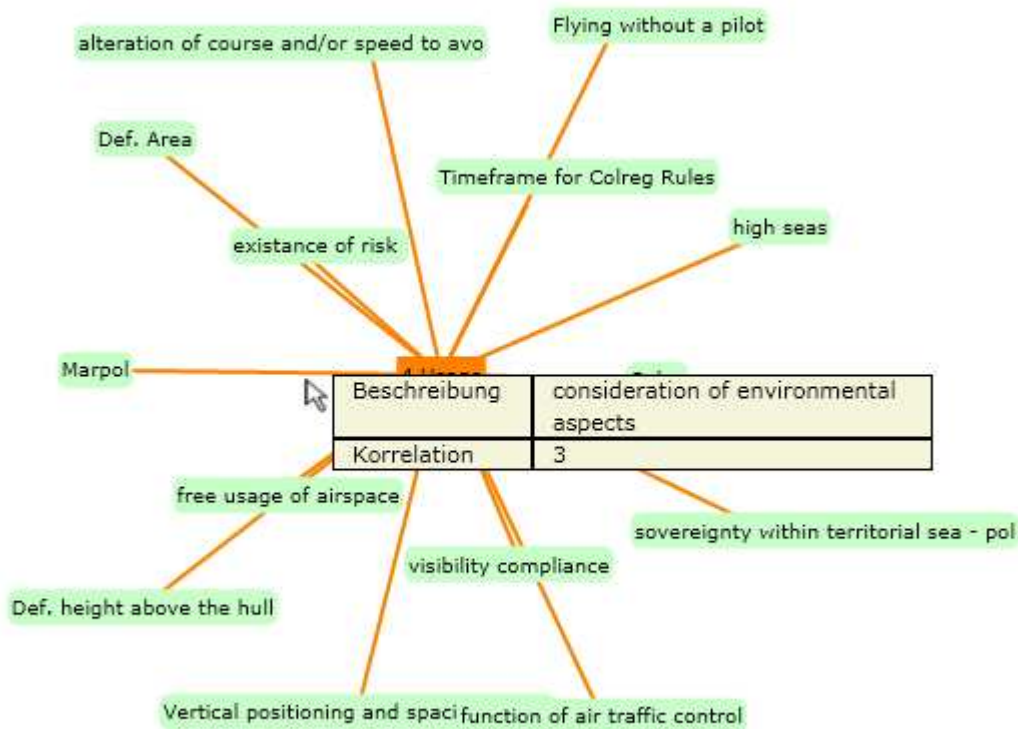


Abbildung 4-10: Ad Hoc Anzeige der Kontextinformationen einer relationalen Beziehung

Die Nutzerinteraktion zum Modellieren

Nicht nur die Analyse mechatronischer Systeme sollte mit Hilfe von Visualisierungstechniken umgesetzt werden, sondern auch das Modellieren. Aus Datensicht werden beim Modellieren Daten manipuliert, sprich neue Datensätze hinzugefügt, bestehende verändert oder gelöscht. Auf den Promesyskontext bezogen bedeutet so eine Manipulation das Hinzufügen, Ändern und Entfernen von Elementen, sowie das Setzen neuer, mit Kontextinformationen versehene, Beziehungen. Es wird nun vorgestellt, wie diese Anforderungen in der Silverlightanwendung umgesetzt werden konnten. Dabei wird beim Aspekt des Erstellens und Ändern von Elementen ein Konzept zur Verwendung unabhängiger Attribute erläutert, welches die Möglichkeit bietet, Elemente flexibel zu attributieren. Zum Hinzufügen eines neuen Elementes in das System, wählt der Nutzer zunächst den Modus „Neues Element“ und klickt auf eine freie Stelle, an der das Element platziert werden soll.

Wie in der folgenden Abbildung zu sehen ist, erscheint dann ein Formular zum Erstellen des neuen Elementes.

List	Komponenten ▾
Titel	Komponente 1052
Kurzbezeichnung	Komp1052
Attribute	Höhe ▾ Add
Farbe	blau
Breite	1,5 m
Gewicht	12,4 Kg
Cancel Save	

Abbildung 4-11: Formular zum Erstellen neuer Elemente

Mit Hilfe des blaugerahmten Dropdownmenüs wählt der Anwender zunächst das Systemelement. Darauf hin wird das Formular dynamisch an die Felder dessen angepasst. Über das zweite Dropdownmenü (roter Rahmen) können nun die im System hinterlegten Attribute hinzugefügt und deren Werte gesetzt werden. Das Verwenden von Attributen, die unabhängig des Systemelementes sind, wurde aus Datensicht folgendermaßen umgesetzt: Der ursprüngliche Ansatz Elemente mit Attributen zu versehen, war es die Attribute über Felder in der Sharepointliste abzubilden. Dies bedeutet jedoch, dass alle Elemente eines Systemelementes, also alle Einträge in einer Liste dieselben Attribute besitzen. Um die Komplexität von mechatronischen Systemen besser abzubilden, muss jedoch die Flexibilität gegeben sein, Elemente einer Liste unabhängig zu attributieren. Dafür wurde im Sharepoint eine Liste „Attribute“ angelegt, in der sämtliche Attribute hinterlegt und ergänzt werden können. In der folgenden Abbildung ist diese Liste exemplarisch gefüllt.

Attribute

Attribute der Systemelemente

Neu ▾ Aktionen ▾ Einstellungen ▾
Titel
Farbe
Länge
Breite
Höhe
Gewicht
Kraft
Kosten
Dauer
Spannung
Stromstärke

Abbildung 4-12: Attributliste mit hinterlegten Attributen

Um nun einzelne Elemente mit diesen Attributen zu verbinden wurde eine Nachschlagebeziehung zur Liste „Attribute“ erstellt. Damit kann zwar die Beziehung eines Elementes zu bestimmten Attributen abgebildet werden, jedoch fehlen noch die zugehörigen Attributwerte. Zur Realisierung muss eine Hilfsliste angelegt werden, deren Einträge eine Nachschlagebeziehung zum jeweiligen Element und zum jeweiligen Attribut herstellt und in einem weiteren Feld, den Wert des Attributes speichert. Auf der kommenden Abbildung sieht man exemplarisch zwei Einträge innerhalb dieser Hilfsliste, welche abbildet, das Komponente1052 Kosten von 52 EUR und eine Kraft von 52 N besitzt.

Komponente 1052	Kosten	52 EUR
Komponente 1052	Kraft	52 N

Abbildung 4-13: Einträge in Hilfsliste zum Speichern der Attributwerte

Zur Bearbeitung vorhandener Elemente, wählt der Nutzer den Modus „Element bearbeiten“. Per Klick auf einen Knoten können dann sämtliche Felder und Attribute des abgebildeten Elementes modifiziert werden.

Zum Herstellen von relationalen Beziehungen bietet der Tab „Graph“ die Möglichkeit über das Erzeugen neuer Kanten zwischen zwei Knoten, die durch diese Knoten abgebildeten Elemente miteinander zu verknüpfen. Da der Anspruch besteht innovative Interaktionsformen einzusetzen, wurde diese Funktion mittels „Drag and Drop“ umgesetzt. Dies erlaubt eine

intuitivere und schnellere Modellierung. Die folgende Abbildung zeigt exemplarisch, wie eine Anforderung mit einer Komponente verknüpft werden kann.



Abbildung 4-14: Setzen neuer Verknüpfungen mit Drag and Drop

Nachdem in den Modus „Neue Verknüpfung“ gewechselt wurde, wird zunächst der Quellknoten gewählt. Durch Bewegen der Maus bei gedrückter Maustaste sieht man wie eine neue Kante aus dem Quellelement (siehe Abbildung 4-14, blauer Rahmen) entsteht. Zum Wählen des Zielelementes (siehe Abbildung 4-14, roter Rahmen) muss die Maustaste über dem gewünschten Knoten losgelassen werden. Die beiden Knoten sind dann miteinander verknüpft und es erscheint ein Kontextmenü zur Eingabe der Kontextinformationen zu dieser relationalen Beziehung.

An dieser Stelle soll erläutert werden, wie innerhalb des Wissensmanagementsystems das Hinterlegen von Kontextinformationen zu relationalen Beziehungen datentechnisch abgebildet ist. Jede Beziehung, die aufgrund von Nachschlagebeziehungen entstanden ist, wird in einer Hilfsliste gespeichert. Diese besitzt neben den Feldern des Quell –und Zielelementes andere beliebige Felder, welche Eigenschaften der Beziehung beinhalten können. Mit der umgesetzten Modellierungsform ist es nun deutlich komfortabler neue Beziehungen anzulegen und diese im gleichen Arbeitsschritt zu attributieren als es die Sharepointlistenformulare erlauben, bei denen der Nutzer die zu verknüpfenden Elemente aus Listen wählen und zur Attributierung anschließend zu einem Listeneintrag in einer weiteren Liste navigieren muss.

Die umgesetzten Nutzerinteraktionen zur Modellierung mit Interaktionsformen wie „Drag and Drop“, die für Rich Internet Applications typisch sind, zeichnen sich durch eine deutlich

schnellere und komfortablere Modellierung der mechatronischen Systeme aus. In der umgesetzten Anwendung behält der Nutzer bei jedem Modellierungsschritt die Übersicht über relevante Teile des Systems, die direkt oder indirekt durch den jeweiligen Schritt tangiert werden. Durch die verbesserte Repräsentation der Semantik des Systems, können während des Modellierens neue semantische Erkenntnisse in Form von Abhängigkeiten der Elemente generiert werden.

Die Modellierung der hierarchischen Beziehungen mit Hilfe der entwickelten Anwendung, welche im folgenden Abschnitt vorgestellt wird, kann die Aussage, dass durch die Visualisierung der Informationen des mechatronischen Systems, neue semantische Erkenntnisse gewonnen werden können, zusätzlich bestätigen.

4.2.2. Der Hierarchie Editor

Wie bereits erwähnt, bestehen zwischen Elementen eines Systemelementes hierarchische Beziehungen. Vor Umsetzung der Anwendung waren diese Beziehungen schon im, mit Sharepoint entwickelten, Wissensmanagementsystem vorhanden. Über eine Nachschlagebeziehung ist für jedes Element das Elternelement hinterlegt, wodurch die gesamte Hierarchie eines Systemelementes abgebildet wird. Zur Erfassung dieser hierarchischen Abhängigkeiten bei Betrachtung der Listen, ist jedoch ein hohes Maß an Vorstellungskraft von Nöten. An diesem Beispiel kann man erneut erkennen, dass auch semantisch korrekt abgebildete Informationen durch fehlende visuelle Darstellung kaum Wissen generieren können. Hinsichtlich dieses Problems bestand eine Anforderung an die Visualisierungsanwendung darin, die hierarchischen Strukturen der Systemelemente grafisch abzubilden. Weiterhin sollte es in dieser Abbildung möglich sein, die Baumstruktur zu modifizieren und auf Wunsch zu speichern. Die nächste Abbildung zeigt die Hierarchie eines Systemelementes.

Mit Hilfe von „Drag and Drop“ ist ein Umhängen einzelner Knoten oder ganzer Unterbäume realisierbar.

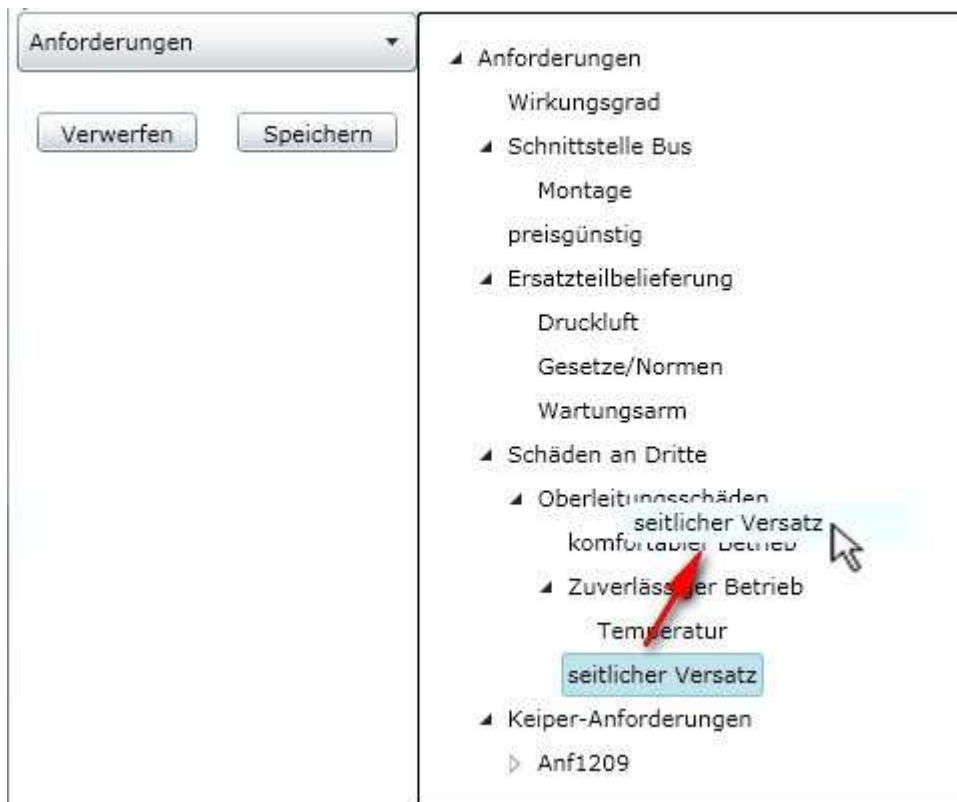


Abbildung 4-15: Mittels Drag and Drop lässt sich die Hierarchie verändern

Der Nutzer behält während der Modifikation den kompletten Überblick über die Hierarchie und kann sie durch ansprechende Darstellung deutlich besser geistig erfassen als bei der Listenansicht. Auf Wunsch kann diese Veränderung in Form eines Updates des Listenelementes im System abgespeichert werden.

4.2.3. Die Beziehungsmatrix

Ein weiteres Werkzeug zum Modellieren von Beziehungen wird im folgenden Abschnitt präsentiert: Die Beziehungsmatrix. Im Vergleich zum Setzen der Beziehungen durch das Zeichnen von Kanten, beschränkt sich das Werkzeug innerhalb eines Modellierungsschrittes nur auf 2 Systemelemente. Der Vorteil besteht jedoch darin, dass die Übersicht noch einmal verbessert und die Schnelligkeit der Interaktion erhöht wird. Der Nutzer verliert dabei nie die Hierarchie der betrachtenden Systemelemente aus den Augen. Des Weiteren können Kontextinformationen zu den Beziehungen jederzeit angezeigt werden. Die folgende

Abbildung dient zur Demonstration der Funktionen der entwickelten Beziehungsmatrix.

	Flüssigkeit (Tinte) übertragen	Flüssigkeit leiten	Flüssigkeit speichern	Gehäuseteile verbinden	Feder gegen Kopfstück abstützen	Minenposition fixieren	Minenposition umschalten	Feder gegen Mine abstützen	Funktionen Kugelschreiber	Minenposition eingefahren fixieren	Minenposition ausgefahren fixieren	Umschalten durch Benutzer ermöglichen	Reibschlussverbindung herste
Beschreibung													
Durch Speichern der Flüssigkeit wird Wirtschaftlichkeit begünstigt													
Korrelation													
6													
Anforderungen Kugelschreiber	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Design	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Neue	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Einsatzbedingungen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fertigung /Montage	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Flexibilität	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kosten	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Leistung / Funktion	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sicherheit/ Überwachung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steuerkonzept	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technik	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Umwelt / Recycling/ Demontage	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wartung / Instandhaltung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wirtschaftlichkeit	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zuverlässigkeit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rahmenbedingungen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entwicklung	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Konstruktion	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Abbildung 4-16: Beziehungsmatrix zum Pflegen von Beziehungen

Über Dropdownmenüs müssen zunächst die zu Verknüpfenden Systemelemente ausgewählt werden (roter und grüner Rahmen). In diesem Beispiel Funktionen und Anforderungen. Der Hierarchiebaum der Elemente wird nach der Auswahl angezeigt. Würde man nach der Selektion der jeweiligen Systemelemente die Matrix mit allen Elementen füllen, also sowohl alle Funktionen als auch alle Anforderungen anzeigen, bestünde die Gefahr, dass die Achsen der Matrix aus unübersichtlich langen Listen bestehen, in denen der Nutzer nur schwer die für ihn relevanten Elemente findet. Daher bestimmt dieser selbst aus welchen Elementen die Matrix aufgebaut werden soll. Mit Hilfe von "Drag and Drop" (roter und grüner Pfeil), füllt er die Matrix mit Elementen. Dabei werden die Unterbäume natürlich berücksichtigt, so dass er komfortabel nur bestimmte modulare Gruppen aus den Systemen für die Verknüpfung betrachten kann. Der innere Bereich der Matrix visualisiert dann, ob Elemente miteinander in Beziehung stehen. Dies wird über eine selektierte Checkbox der korrespondierenden Elemente deutlich. Da aber auch für die Beziehungsmatrix der Anspruch gilt, nicht nur zu zeigen, dass Elemente relational in Beziehung stehen, sondern auch in welchem Kontext, kann der Nutzer diese Informationen für jede Beziehung abrufen. Dazu bewegt der die Maus

auf eine Checkbox (blauer Rahmen). Die Kontextinformationen der korrespondierenden Relation werden dann angezeigt (gelber Rahmen).

5. Zusammenfassung

Dass vollständige und konsistente Informationen bei mangelnder Darstellung das Generieren von Wissen verhindern können, kommt vor allem bei großen, komplexen und vernetzten Datenbeständen zum Tragen. Geeignete Informationsvisualisierungstechniken können beim Verstehen der Informationen ein hohes Maß an Unterstützung bieten. Ziel der hier entwickelten Anwendung war es, Nutzer eines Wissensmanagementsystems mit verschiedenen Werkzeugen zu versorgen, die Sie beim Beherrschen der Informationsflut sowohl beim Analysieren als auch beim Modellieren unterstützen. Das mit Microsoft[®] Silverlight[®] entwickelte Programm konnte vollständig in Microsoft[®] Sharepoint[®] integriert werden. Dabei wurde ein System erschaffen, welches sich an einen Anforderungskatalog an ein Visualisierungssystem, welches aus den Komponenten Benutzeroberfläche, Interaktion und Funktionalität besteht, entwickelt. Die Anwendung erlaubt dem Nutzer Zusammenhänge und Abhängigkeiten besser zu verstehen. Die Elemente der Oberfläche konnten angemessen gestaltet werden, um die Wahrnehmung des Nutzers positiv zu unterstützen. Durch Such- und Filtermechanismen sind sowohl Objekte gut auffindbar als auch selektierbar. Die Interaktion mit dem Informationsraum erlaubt eine Anpassung der Sicht auf die Informationen und damit einen verbesserten, problembezogenen Zugang zu den Daten. Verschiedene Bearbeitungstechniken ermöglichen es die Elemente des Informationsraums dauerhaft zu modifizieren. Zur Erhöhung der Ausgabegeschwindigkeit wurden Möglichkeiten geschaffen, die Ergebnisdatenmenge vor der grafischen Darstellung, zu reduzieren. Des Weiteren realisiert eine Konfigurationskomponente eine Anpassung der Darstellung an persönliche Nutzerpräferenzen

Durch Umsetzung einer Beziehungsmatrix konnte weiterhin das Potential von Visualisierungstechniken und innovativen Interaktionsformen gezeigt werden. Der Nutzer bekommt ein Werkzeug an die Hand, mit dem er in der Lage ist komplexe Strukturen innerhalb mechatronischer Systeme zu verstehen und sie mit Hilfe seines Expertenwissens so zu modellieren, dass das Design des Produktes robuster gestaltet wird, welches positiven Einfluss auf alle Phasen des Produktlebenszyklusses implizieren kann.

A. Anhang

A.1 Konfigurationsdatei im XML-Format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<SharepointLists>
<SharepointList id='b0740369-e740-42dc-b6b2-2212262ef44e' typeId='1' visible='True'
dispField='Kurzbezeichnung' shape='Hexagon' color='255,0,0'>
<ConnectionFields>
<ConnectionField internalName='Zugeordnete_x0020_Anforderungen'
lookupList='{5a691a8f-8be6-40fd-b21e-54c21575e00d}' />
<ConnectionField internalName='Zugeordnete_x0020_Komponenten'
lookupList='{b0740369-e740-42dc-b6b2-2212262ef44e}' />
<ConnectionField internalName='Zugeordnete_x0020_Funktionen' lookupList='{54ee7bd1-
f02a-43ea-8e72-4ba715f820e1}' />
</ConnectionFields>
<AttributeFields>
<AttributeField internalName='Title' displayName='Titel' />
<AttributeField internalName='File_x0020_Type' displayName='Dateityp' />
<AttributeField internalName='Kurzbezeichnung' displayName='Kurzbezeichnung' />
<AttributeField internalName='Dokument_x0020_upload' displayName='Dokument upload'
/>
<AttributeField internalName='ContentType' displayName='Inhaltstyp' />
<AttributeField internalName='_CopySource' displayName='Kopiequelle' />
<AttributeField internalName='_UIVersionString' displayName='Version' />
<AttributeField internalName='ID' displayName='ID' />
</AttributeFields>
</SharepointList>
<SharepointList id='5a691a8f-8be6-40fd-b21e-54c21575e00d' typeId='2' visible='True'
dispField='Kurzbezeichnung' shape='Ellipse' color='0,0,255'>
<ConnectionFields><ConnectionField internalName='Zugeordnete_x0020_Funktionen'
lookupList='{54ee7bd1-f02a-43ea-8e72-4ba715f820e1}' />
```

```
<ConnectionField internalName='Bestandteil_x0020_von' lookupList='{5a691a8f-8be6-40fd-
b21e-54c21575e00d}' />
</ConnectionFields><AttributeFields><AttributeField internalName='Title'
displayName='Titel' />
<AttributeField internalName='File_x0020_Type' displayName='Dateityp' />
<AttributeField internalName='Kurzbezeichnung' displayName='Kurzbezeichnung' />
<AttributeField internalName='Dokument_x0020_upload' displayName='Dokument upload'
/>
<AttributeField internalName='ContentType' displayName='Inhaltstyp' />
<AttributeField internalName='_CopySource' displayName='Kopiequelle' />
<AttributeField internalName='_UIVersionString' displayName='Version' />
<AttributeField internalName='ID' displayName='ID' />
</AttributeFields>
</SharepointList>
<SharepointList id='54ee7bd1-f02a-43ea-8e72-4ba715f820e1' typeId='3' visible='True'
dispField='Kurzbezeichnung' shape='rounded Rectangle' color='0,255,0'>
<ConnectionFields><ConnectionField internalName='Zugeordnete_x0020_Anforderung'
lookupList='{5a691a8f-8be6-40fd-b21e-54c21575e00d}' />
<ConnectionField internalName='Zugeordnete_x0020_Komponenten'
lookupList='{b0740369-e740-42dc-b6b2-2212262ef44e}' />
</ConnectionFields><AttributeFields><AttributeField internalName='Title'
displayName='Titel' />
<AttributeField internalName='File_x0020_Type' displayName='Dateityp' />
<AttributeField internalName='Kurzbezeichnung' displayName='Kurzbezeichnung' />
<AttributeField internalName='ContentType' displayName='Inhaltstyp' />
<AttributeField internalName='_CopySource' displayName='Kopiequelle' />
<AttributeField internalName='_UIVersionString' displayName='Version' />
<AttributeField internalName='ID' displayName='ID' />
</AttributeFields>
</SharepointList>
<SharepointList id='7aa0ca02-b782-4b31-a52c-132d86f3e925' typeId='4' visible='True'
dispField='Stadtname' shape='Hexagon' color='255,255,0'>
<ConnectionFields />
<AttributeFields>
```

```
<AttributeField internalName='Title' displayName='Titel' />
<AttributeField internalName='File_x0020_Type' displayName='Dateityp' />
<AttributeField internalName='Stadtname' displayName='Stadtname' />
<AttributeField internalName='Land' displayName='Land' />
<AttributeField internalName='ContentType' displayName='Inhaltstyp' />
<AttributeField internalName='_CopySource' displayName='Kopiequelle' />
<AttributeField internalName='_UIVersionString' displayName='Version' />
<AttributeField internalName='ID' displayName='ID' />
</AttributeFields>
</SharepointList>
</SharepointLists>
```

Literaturverzeichnis

- SchuMü2000 **Schumann, H.; Müller, W.**
Visualisierung - Grundlagen und allgemeine Methoden, Springer, 2000
- CaMackSchn99 **Card, S.; Mackinlay J; Schneidemann, B.**
Information Visualization - Using Vision to think
- Bert08 **Bertram, S.**
Entwicklung eines produktlebenszyklusgeleitenden
Wissensmanagementsystems in Microsoft ® Windows ® Sharepoint ®
für ein nachhaltiges robustes Design mechatronischer Systeme
- Däß99 **Däßler, R.**
Informationsvisualisierung - Stand, Kritik und Perspektiven
<http://fabdp.fh-potsdam.de/infoviz/paper/InfoVis99.pdf>
16.09.2009
- Bur05 **Burkhard, R.**
Wissensvisualisierung wird Unternehmenskulturen nachhaltig
verändern
<http://www.semanticweb.at/index.php?id=1&subid=36&action=resource&item=22>
28.09.2009
- Die07 **Diehl, S.**
Software Visualization: Visualizing the structure, behaviour and
Evolution of Software, Springer, 2007
- Naj03 **Najarro, S.**
Visualisierung von Informationsräumen
<http://www.db-thueringen.de/servlets/DerivateServlet/Derivate-2531/ilm1-2003000144.pdf>
-

SnTiKu03

Snell, S. ; Tidwell, R. ; Kulchenko, C..

Webserviceprogrammierung mit SOAP, Springer, 2003