

# Visualisierung von komplexen Produktstrukturen

## Studienarbeit

Abteilung Informatik  
Hochschule für Technik Rapperswil

Herbstsemester 2011

Autor(en): Christoph Rosenberger, Michael Steiner  
Betreuer: Dr. Daniel Keller  
Projektpartner: foryouandyourcustomers, Pfäffikon ZH  
Experte: Jonathan Möller

# Abstract

---

foryouandyourcustomers ist ein junges Unternehmen, welches sich im Bereich Multichannel stark gemacht hat. In diesem Fachgebiet gibt es aktuell nur Software, welche die Implementierung von Produktstrukturen unterstützt. Eine Software die das abstrakte modellieren von solchen Produktstrukturen ermöglicht und somit unabhängig von einer konkreten Implementierung ist, existiert noch nicht.

Wir haben die verschiedenen Aspekte solcher Projekte untersucht und eine solide Grundlage für die Umsetzung einer solchen Software (PSA) entworfen.

Mit PSA ermöglichen wir dem Entwickler durch geschickte Visualisierungen von Produktstrukturen, den Überblick über die komplexen Zusammenhänge dieser Strukturen zu behalten. Aber auch der Auftraggeber und dessen Mitarbeiter können so besser in das Projekt mit einbezogen werden. Dadurch kann dem Kunde und dem Endkunde eine einheitliche, ansprechende und eine technisch hoch stehende Lösung angeboten werden.

# Inhaltsverzeichnis

1	Management Summary .....	4
2	foryouandyourcustomers .....	5
3	Produktstrukturierung .....	6
3.1	Workproducts .....	6
3.2	Probleme bei der Produktstrukturierung .....	6
4	Ziele von PSA .....	8
4.1	Nutzen für foryouandyourcustomers .....	8
4.2	Nutzen für die IT-Abteilung des Kunden .....	8
4.3	Nutzen für die Mitarbeiter des Kunden .....	9
4.4	Nutzen für den Manager des Kunden .....	9
4.5	Nutzen für den Endkunden .....	9
5	Datenbasis .....	10
5.1	System und MetaInformation Layer .....	12
5.2	Touchpoints .....	16
5.3	User .....	16
5.4	Weitere Konzepte .....	17
5.5	Metadaten .....	17
6	Visualisierungen .....	19
6.1	Konzepte .....	19
7	Bedienkonzepte .....	22
7.1	Visualisierung, Navigation, Editieren .....	22
7.2	Variante: Verschiedene Tools für Visualisierung und Datenmanipulation .....	22
7.3	Variante: Total verschmolzen .....	23
7.4	Variante: Verschiedene Bereiche innerhalb des Fensters .....	23
7.5	Filtern von Daten .....	24
7.6	Automatisch generierte vs. manuell aufbereitete Visualisierungen .....	24
7.7	Perspektiven bezogene Präsentationen .....	25
7.8	Templates .....	25
7.9	UML Editor .....	26
8	Technologien .....	27
9	Bachelorarbeit .....	28
10	Prototyp .....	31
10.1	Klassendiagramm .....	32
10.2	Darstellungen .....	34

11	Alternative Lösungsansätze .....	39
11.1	3D .....	39
11.2	EMF .....	43
11.3	GMF, GEF, GEF3D .....	44
12	Glossar .....	45
13	Quellen .....	46
13.1	Bilder .....	46
13.2	Literaturverzeichnis .....	46
14	Abbildungsverzeichnis .....	47
15	Anhang .....	48
15.1	A: Fragenkatalog der Sitzung vom 06.12.2011 .....	48
15.2	B: Aufgabenstellung .....	55
15.3	C: Erklärung über die eigenständige Arbeit .....	56
15.4	D: Persönliche Berichte .....	58

## 1 Management Summary

Ausgangslage	foryouandyourcustomers, eine sehr junge Firma, ist auf dem Weg zum Marktleader im Bereich Multichannel Management und Produktstrukturierung. Eines ihrer Probleme ist die ausführliche und einheitliche Dokumentation ihrer Erkenntnisse während eines Produktstrukturierungs Projekts.
Vorgehen & Technologie	Wir haben verschiedene Ansätze für PSA (Product Structure Architect) erarbeitet. Die für gut befundenen haben wir in einem PHP-MySQL Prototypen umgesetzt.
Ergebnisse	In dieser Arbeit werden grundlegende Fragen beleuchtet wie ein solches Tool aussehen könnte in welchem eine Produktstrukturierung erarbeitet und dokumentiert werden kann.
Ausblick	Im Frühlingssemester 2012 wird in einer Bachelor Arbeit PSA umgesetzt werden.

## 2 foryouandyourcustomers

Über foryouandyourcustomers	foryouandyourcustomers ist ein junges Unternehmen, welches sich im Bereich Multichannel stark gemacht hat. Sie unterscheiden sich enorm von Firmen, welche einfache E-Commerce Lösungen anbieten. Zur Zeit zählt die Firma 13 Mitarbeiter und befindet sich in einer Hyperwachstumsphase. Sie hebt sich von anderen Firmen ab, weil sie die "Werbesicht" und das technische Know-How liefern. Bei foryouandyourcustomers ist Business und Technologie unter einem Dach.
Ziele	Das Ziel ist es, den Kunden sowie den Endkunden (Konsumenten) zu unterstützen. Gemeinsam mit dem Kunden untersucht man was für eine Lösung er benötigt, wie umfassend sein PIM sein soll, welche Produkte und welche Attribute darin verwaltet werden sollen, welche Kanäle sinnvoll sind und welche Auswirkungen dies auf die Datenhaltung hat. Des Weiteren wird analysiert, was schon vorhanden ist und wie man diese Daten weiterhin nutzen kann. Den Endkunden möchte man beim Finden oder dem Vergleich von Produkten unterstützen. Durch Schaffung von Klarheit und Einheitlichkeit auf allen Kanälen kann dies gewährleistet werden. Eine reine Informatik-Firma würde eine gute technische Lösung liefern und eine Werbefirma ein schönes und ansprechendes Design. foryouandyourcustomers bietet dem Kunden beides. Somit ist auch gewährleistet, dass sich zum Beispiel das Design einer Website technisch auch sinnvoll umsetzen lässt.
Ist - Zustand	Ein Product Information Management System ist sehr Komplex und umfangreich, deshalb gehen Details schnell vergessen. Zur Zeit wird dem durch ein möglichst kleines Team entgegengewirkt. Dies führt aber dazu, dass ein Mitarbeiter nicht einfach ausgewechselt werden kann. Wenn ein Mitarbeiter ausgewechselt würde, müsste man wieder von vorne beginnen. Auch ist es schwierig jemand neues einzuweisen, denn das komplette Wissen ist nur in den Köpfen der Mitarbeiter. Eine spätere Weiterentwicklung des Projektes ist somit auch nicht trivial. Das Datenmodell wird direkt im PCM designed.
Kontakt	<p><b>Daniel Keller</b>, CTO</p> <p><b>Jonathan Möller</b>, Beratung und Verkauf</p> <p><b>Atanas Aladjov</b>, Berater</p>

### 3 Produktstrukturierung

#### 3.1 Workproducts

Begriffsdefinition	<p>Das erste Workproduct von foryouandyourcustomers ist eine detaillierte Analyse über den Ist-Zustand des Kunden. Es gilt zu dokumentieren welche Daten schon in welcher Form vorhanden sind.</p> <p>Ein weiterer wesentlicher Bestandteil ist die Namensgebung, also die Einigung mit dem Kunden über die Bedeutung von Begriffen wie SKU, Artikel oder Produkt.</p>
Produktstrukturierung	<p>Nun gilt es, die gewonnen Erkenntnisse in eine detaillierte und technisch konkrete Produktstruktur abzubilden. Ein Datenmodell mit allen Klassen, Attributen und Attributdefinitionen wird erstellt. Es werden Prozesse designed, welche beschreiben, wie das PIM produktiv unterhalten wird (z.B. Prozesse um Produktdaten aktuell halten).</p>
Planung für die Umsetzung	<p>Nach der Produktstrukturierung ist noch nichts implementiert. Es wurde ein Plan erstellt, der definiert wie was implementiert werden soll. Dies kann durch foryouandyourcustomers selbst oder eine andere Agentur (dies ist eher ein exotischer Fall) geschehen.</p>

#### 3.2 Probleme bei der Produktstrukturierung

Standard	<p>foryouandyourcustomers hat / verwendet noch keine standardisierte Produktstruktur. Sie sind somit abhängig von dem eingesetzten PCM.</p> <p>Es existiert keine gute Lösung um eine einheitliche und vollständige Dokumentation zu erstellen. Das Format einer solchen Beschreibung ist nicht klar, idealer Weise wird es daher in einem PCM "dokumentiert" / umgesetzt.</p> <p>Wenn eine gute Lösung für die Dokumentation existieren würde, könnte man aus Projekten Templates herleiten. Dies würde die Effizienz steigern und auch das Risiko, dass man etwas vernachlässigt, verringern.</p>
Umfang	<p>Eines der Hauptprobleme ist der Umfang eines solchen Projektes. Grössere Firmen können über 100'000 Produkte anbieten. Diese Produkte haben wiederum etliche Attribute. All diese Informationen zu erfassen und dabei keine zu vergessen ist extrem schwierig. Eine weitere Herausforderung ist es, das richtige Mass zwischen Detailliertheit und Kosteneffizienz zu finden. Jede Information, die man später speichern möchte muss erfasst werden und ist somit mit Kosten und Aufwand verbunden.</p> <p>Eine weitere wesentliche Frage ist, welche Daten überhaupt erfasst werden sollen.</p> <ul style="list-style-type: none"> <li>• Soll das Werbematerial auch im PIM erfasst werden?</li> <li>• Soll Migros auch die Klubschule im PIM erfassen, oder nur Artikel?</li> <li>• Soll man die Länge einer Stange beliebig wählen können</li> </ul>

oder sollen verschiedene Längen als Varianten modelliert werden?

#### Vorhandene Daten

Wenn ein solches Projekt gestartet wird, sind meistens schon Daten über Produkte, Bestellungen oder Kunden in verschiedenen Formen vorhanden. Es könnte zum Beispiel sein, dass alle Produkte in einem Umsystem erfasst wurden. Dieses Umsystem könnte ein Excel-Sheet oder ein ERP sein. Nun gilt es, diese Daten zu bewerten und gegebenen Falls zu verwenden.

Gewisse Details über ein Produkt wurden möglicherweise nicht in Computersystemen erfasst. Ca. 30% der Informationen über die Produkte sind nur in den Köpfen der Mitarbeiter gespeichert. foryouandyourcustomers spricht mit den einzelnen Mitarbeitern und dokumentiert ihr Wissen.

Es ist wichtig, sich einen guten Überblick über all dieses Know-how zu verschaffen. Sprich von wo oder von wem stammen diese Informationen wie wertvoll sind sie und in welcher Form sind sie vorhanden.

#### Mitarbeiter des Kunden

Oft haben Leute Angst vor neuen Computersystemen. Sie fühlen sich bedroht und fürchten sich davor, durch ein solches abgelöst zu werden. Es ist wichtig, diesen Personen klar zu machen, dass sie immer noch wichtig sind und gebraucht werden. Sie sollten die Vorteile eines PIMs verstehen und erkennen, dass sich für sie mehr oder weniger nur die Dateneingabe ändert.

#### Auftraggeber

Ein weiteres Problem ist es, dem Auftraggeber die Komplexität dieser Arbeit zu vermitteln. Wenn man eine richtig gute Lösung findet, ist diese meist relativ einfach zu verstehen, was es nicht leicht macht, den Aufwand zu begründen.

## 4 Ziele von PSA

PSA soll foryouandyourcustomers bei der Umsetzung ihrer Produktstrukturierungsmethode ([http://www.foryouandyourcustomers.com/wp-content/uploads/productsheet\\_productstructure\\_de.pdf](http://www.foryouandyourcustomers.com/wp-content/uploads/productsheet_productstructure_de.pdf)) unterstützen, insbesondere in den Punkten 2.3 und 2.4.

### 4.1 Nutzen für foryouandyourcustomers

foryouandyourcustomers entwickelt Produktstrukturen. Hierzu werden sowohl die Produkte des Kunden analysiert, als auch seine bestehenden Informatiksysteme (ERP, PLS, MAM, CMS, ...). Das Resultat der Arbeit ist eine Produktstruktur (Datenmodell), welche als Single Point of Truth zentral für das darauf aufbauende PIM ist.

Dokumentieren	Das Ergebnis der Produktstrukturierung kann in einem einheitlichen Tool dokumentiert werden. Diese Dokumentation sollte möglichst vollständig sein: Vom Datenmodell des PIM, der Klassifizierung und Kategorisierung der Produkte, über die Umsysteme und Datenflüsse zum PIM bis hin zu den Channels und Touchpoints, welche den Endkunden zur Verfügung gestellt werden sollen. Auch die Prozesse und Arbeiten, welche laufen müssen, um dieses System zu unterhalten, sollen geplant werden können. Ausserdem sollen Informationen über den Produktstrukturierungsprozess selbst erfasst werden können.
Visualisieren	Die dokumentierten Resultate sollen in verschiedenen Perspektiven angezeigt werden. Dadurch sollen foryouandyourcustomers-Entwickler ihre Resultate erforschen können. PSA unterstützt sie beim Verstehen ihrer Arbeit und beim Erkennen von Mängeln und Problemen.
Know-how Transfer	Für die Modellierung stehen Templates und Listen von Standardwerten zur Verfügung. Hierdurch kann PSA genutzt werden, um Know-How innerhalb der Firma foryouandyourcustomers zu verbreiten.

### 4.2 Nutzen für die IT-Abteilung des Kunden

Der IT-Abteilung des foryouandyourcustomers-Kunden ist bei der Umsetzung der geplanten Lösung beteiligt und muss später das System unterhalten.

Dokumentieren	Die Produktstrukturierung ist vollständig dokumentiert. Somit sind die Grundlagen für die Umsetzung bekannt. Für die Wartung des Systems muss dessen Aufbau bekannt sein und verstanden werden. Hierzu dient eine vollständige, gut strukturierte Dokumentation.
Verständnis fördern	Verschiedene Aspekte der Produktstruktur können aus diversen Blickwinkeln betrachtet werden. Somit sollen die Probleme und Lösungen besser verstanden werden.

### 4.3 Nutzen für die Mitarbeiter des Kunden

Beim Kunden werden Mitarbeiter in verschiedenen Rollen mit der Multichannellösung arbeiten (z.B. Katalog-Designer, Übersetzer, ...)

Einbindung in Entwicklungsprozess	Durch geschickt gestaltete Perspektiven kann mit dem Mitarbeiter besprochen werden, ob alle für seine Arbeit nötigen Daten abgefragt bzw. eingegeben werden können. Es können Daten gefiltert werden, so dass man Visualisierungen mit den für den konkreten Mitarbeiter relevanten Daten erstellen kann.
Für neue Systeme gewinnen	Dem Mitarbeiter kann aufgezeigt werden, dass ihm durch die neue Multichannellösung nichts genommen wird: Es sind immer noch alle Daten und alt bekannten Speichersysteme (ERP, PLS, ...) vorhanden, nur der Zugriff auf seine Daten wird neu geregelt.

### 4.4 Nutzen für den Manager des Kunden

Der Manager des Kunden ist verantwortlich für das Budget. Er will sich sicher sein, dass seine Investitionen lohnend sind.

Verständnis dokumentieren	Durch die vollständige Dokumentation der Produktstrukturierung kann dem Manager gezeigt werden, dass man seine Firma und seine Bedürfnisse verstanden hat.
Zukunftssicherheit dokumentieren	Dem Manager kann aufgezeigt werden, wie durch die neuen Channels auch neue Daten, wie z.B. Community Contents, generiert und genutzt werden können. Er sieht, dass das System für die aktuellen Technologien und Trends gerüstet und für die Zukunft bereit ist.
Wertschätzung	Die neue Produktstruktur ist der Single Point of Truth im PIM. Der Manager muss verstehen, dass dieser sehr wichtig ist und gut durchdacht sein muss. Er sieht den Wert seiner Investition.

### 4.5 Nutzen für den Endkunden

PSA unterstützt foryouandyourcustomers bei der Entwicklung eines PIMs. Dem Endkunde (z.B. ein Konsument) werden so einheitliche und intuitive Touchpoints zur Verfügung gestellt und gut durchdachte Customer journeys ermöglicht.

## 5 Datenbasis

### Datenbasis

In diesem Kapitel möchten wir aufzeigen, wie die Daten von PSA strukturiert werden können. Wir gehen davon aus, dass im Grossen und Ganzen dieses Domainmodell verwendet werden kann. Detailprobleme können wir aber noch nicht lösen: Uns fehlt das nötige Wissen wie exakt die Modellierung aussehen soll. Bei einer stimmigen Grobstruktur sollte dies aber kein grösseres Problem sein; die Detailfragen müssen in den Iterationen geklärt werden, in welchen die Daten auch gebraucht werden.

### Layer

PSA benötigt Daten über diverse Konzepte unterschiedlicher Natur. Um die Komplexität der Datenbasis beherrschbar zu machen, haben wir sie in vier Layer Unterteilt.

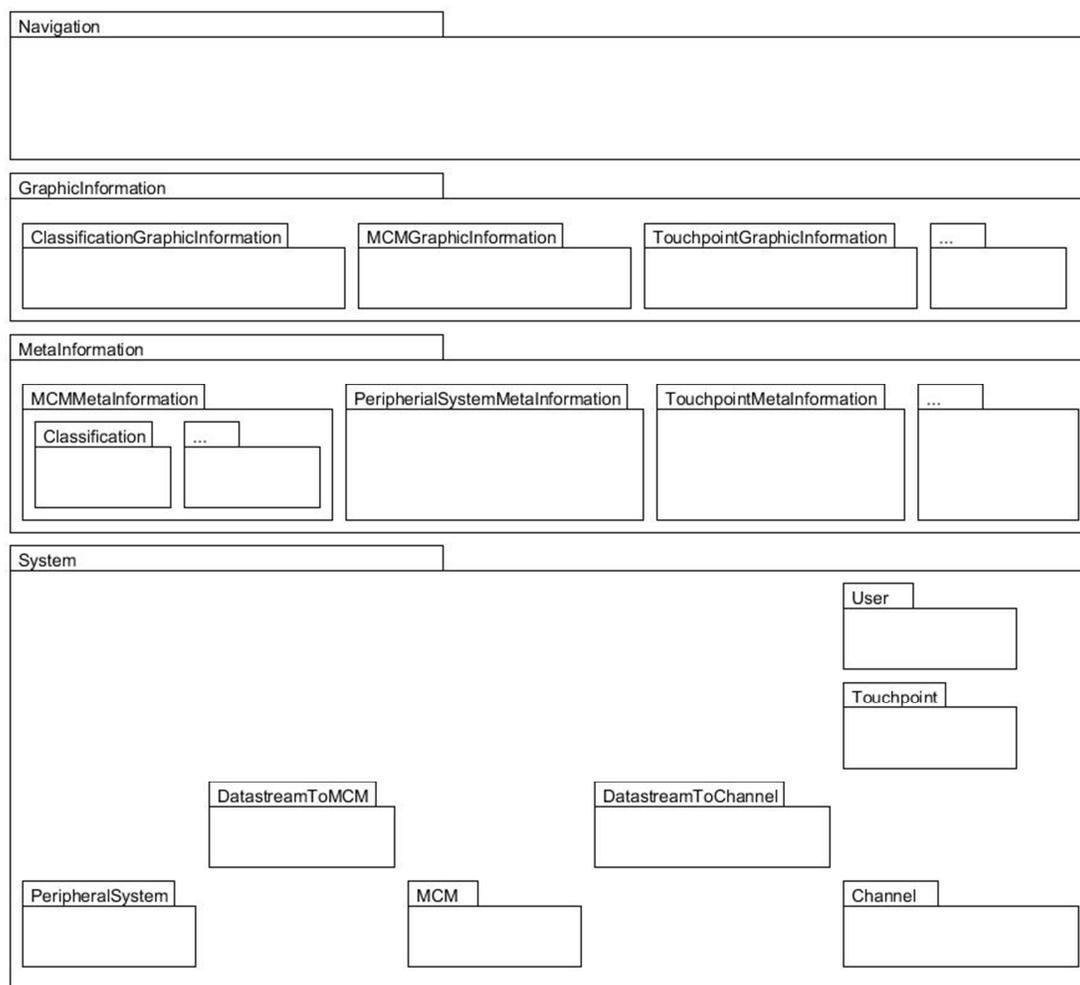


Abbildung 1 Layer der Datenbasis

### System

Im System Layer wird das System welches umgesetzt werden soll abstrahiert. Dies beinhaltet die Teilsysteme von den Umsystemen über das MCM bis hin zu den Channels, die Datenströme zwischen den Teilsystemen, die Touchpoints und die User.

MetaInformation	Für Daten im unteren Layer werden diverse Metainformationen erfasst. So möchte man z.B. von Attributen wissen, wie häufig sie in den bestehenden Umsystemen abgefüllt sind. Oder die Klassifikation, eine spezielle Sicht auf die Modellierung der Produktattribute.
GraphicInformation	Im GraphicInformation Layer werden alle Daten gespeichert, welche für die Visualisierungen gebraucht werden.
Navigation	Im Navigation Layer werden diejenigen Daten gespeichert, welche für die Navigation benötigt werden. So könnte es z.B. verschieden detaillierte Ansichten eines Systems geben, so dass man Zoomen könnte.

### 5.1 System und MetaInformation Layer

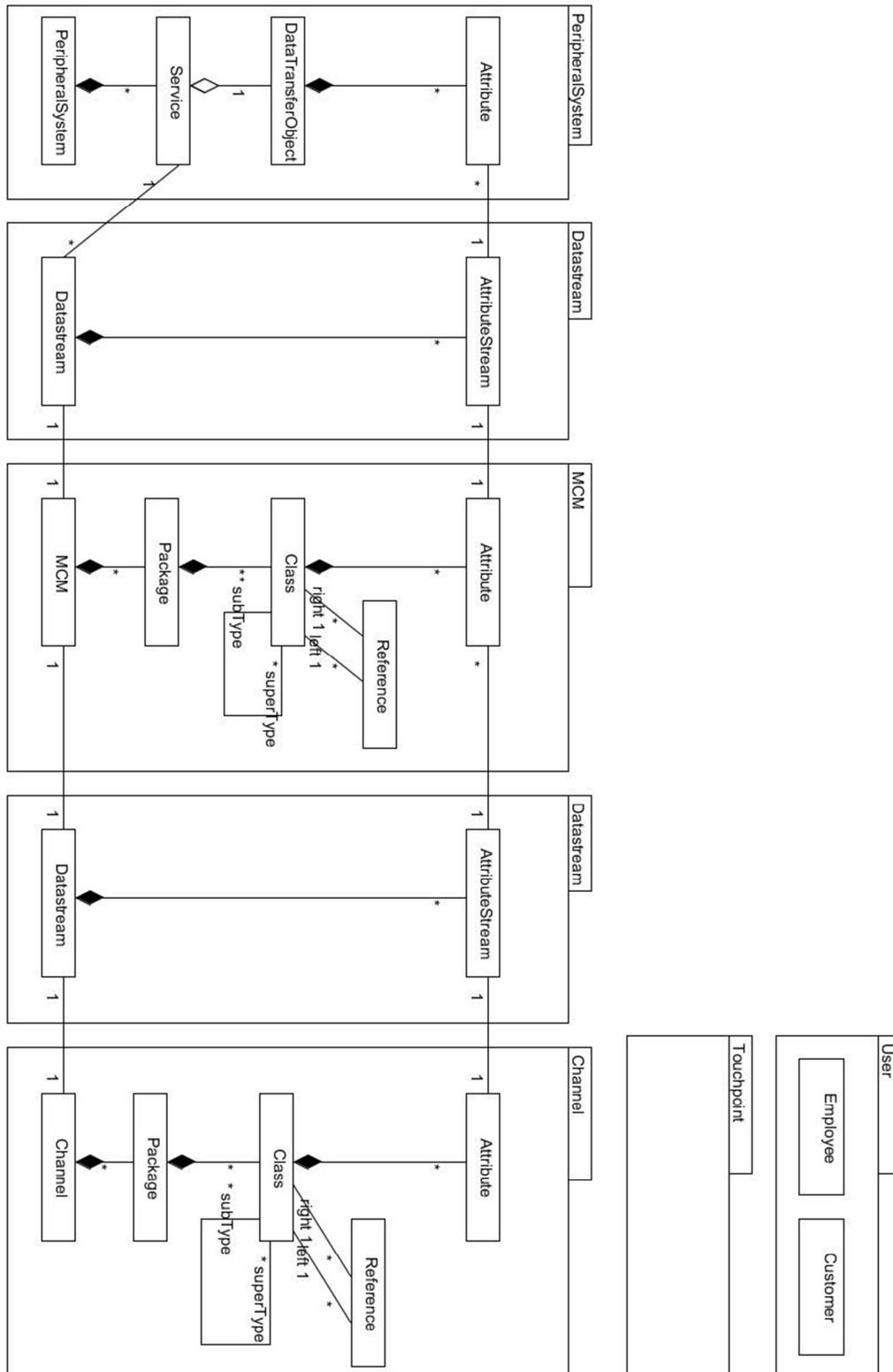


Abbildung 2 Einfache Modellierung des System Layer

### 5.1.1 PeripheralSystem

#### Service

Die Umsysteme können als Services abstrahiert werden. Zu prüfen gilt, ob die Services zustandslos modelliert werden können. Folgende Punkte muss man sicherlich modellieren können:

- Welche Abfragen (Services) gibt es?
- Welche Inputparameter brauchen diese Services?
- Welche Return Werte geben diese Services?

Keine Rolle dürfte hingegen die interne Struktur der Umsysteme spielen.

#### Metainformationen

Folgende Fragen sollten mit den Metainformationen beantwortet werden können:

- Welche der Systeme sind neu, welche gibt es bereits?
- Welche Services der Systeme sind neu, welche gibt es bereits?
- Welche Änderungen müssen an den Systemen vorgenommen werden?
- Wie viele Datensätze sind bei einer Anfrage zu erwarten?
- Sind die Daten über die Services hinweg normalisiert?
- Wie schnell sind die Abfragen?

### 5.1.2 DatastreamToMCM

#### Anknüpfung der Umsysteme an MCM

Im Package DatastreamToMCM geht es darum aufzuzeigen, wie das MCM an die Umsysteme angeknüpft ist, d.h. welche Attribute des MCM aus welchen Umsystemen befüllt werden, aber auch umgekehrt, welche Daten aus dem MCM zurück in die Umsysteme gespielt werden.

#### Prozesse

Eine offene Frage ist ob auch die Prozesse welche für den Datenstrom verantwortlich sind, modelliert werden müssen, oder ob das MCM selbst sich alle Informationen besorgt.

#### Metainformationen

Durch die Metainformationen sollten folgende Fragen geklärt werden:

- Werden die Daten referenziert oder kopiert?
- Findet die Synchronisation automatisch oder manuell statt?
- Wie häufig wird der Service aufgerufen?
- Werden die Werte eins zu eins übernommen, oder müssen sie transformiert werden?
- Wie sind die Systeme miteinander verbunden?
  - Geschwindigkeit der Leitung
  - Bandbreite
  - Ausfallsicherheit

### 5.1.3 MCM

#### Innere Struktur

Das MCM soll als Single Point Of Access alle nötigen Informationen für die Channels zur Verfügung stellen. Im Wesentlichen abstrahieren wir das MCM als Datenmodell. Dieses wiederum modellieren wir als (einfaches) UML-Klassendiagramm. Wir sehen vor, dass die verschiedenen Bereiche des MCM, d.h. PIM, Kampagne, CRM, etc. als

Packages modelliert werden.

## UML- Klassendiagramm

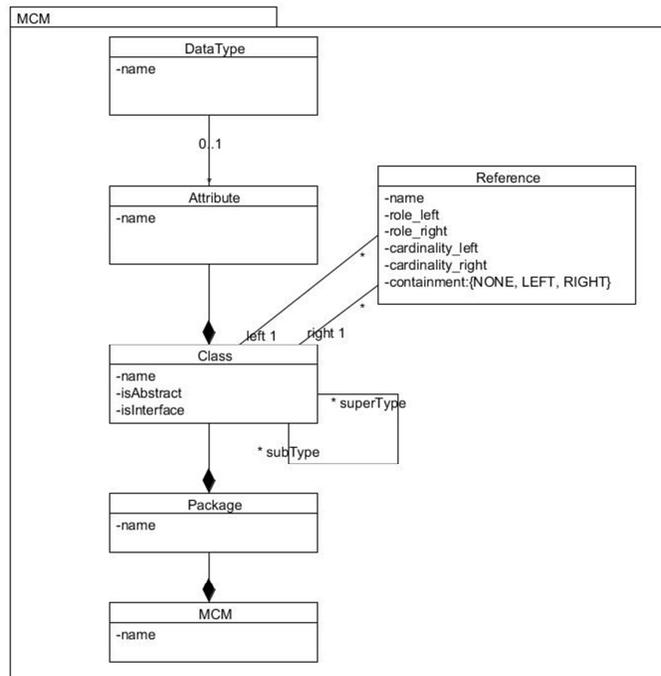


Abbildung 3 UML Klassendiagramm Meta Modell

Wir haben ein relativ einfaches Metamodell entwickelt. Vom Umfang her haben wir uns am eCore Modell von EMF orientiert. Wir gehen davon aus, dass dies in der Praxis genügend Modellierungsmöglichkeiten bietet.

## DataType

Fraglich ist, wie weit die Modellierung von Datentypen gehen sollte. Dies könnte in C-artiger Weise für fast jedes Attribut einen eigenen Datentyp vorsehen; diese könnten dann nicht nur mit einem eigenen Namen und Basisdatentyp, sondern auch mit Constraints wie Wertebereich modelliert werden. Die einfachere Variante ist, dass die Datentypen einfach ein String-Attribut der Attribut-Klasse sind. Allfällige weitere Informationen wie Wertebereiche könnten dann als Meta-Attribute erfasst werden.

## Verhalten gegenüber Channel

Vertieft abzuklären ist, ob aus Sicht der Channels die Modellierung des MCM als Datenmodell günstig ist. Dies würde bedeuten, dass die Channels direkten Zugriff auf dieses Datenmodell haben. Dies entspricht auch dem, was wir bisher verstanden haben. Vorstellbar wäre aber auch, dass sich das MCM als Service Anbieter präsentiert, analog den Umsystemen zum MCM.

## Attribut Metaattribute

Mitarbeiter:

- Wer ist die Ansprechperson für das Attribut?
- Welche Mitarbeiter können die Werte für dieses Attribut liefern (inital und im laufenden Betrieb)?

Qualität:

- In welcher Qualität liegen die Daten vor? Wie zuverlässig

sind die Werte?

- Wie häufig ist das Attribut in den alten Systemen abgefüllt?
- Wie häufig sollte das Attribut neu abgefüllt sein?

Wert:

- Wie teuer ist der Unterhalt eines Attributes (initial und im laufenden Betrieb)?
- Ist das Attribut internationalisierbar?
- Wie häufig muss der Wert angepasst werden?
- Wie wichtig ist das Attribut? Wieviel Wert generiert es? Von wem werden die Informationen gebraucht?
- Ist das Attribut ein Informations Objekt oder ein gestaltendes Objekt? Oder Beides?

Technisch:

- Welchen Wertebereich kann das Attribut annehmen
- Weitere Details zum Datentyp (z.B. Codierung wie UTF-8)
- Einheit (unit, z.B. Kilogramm)

Class Meta Attribute

Natürlich können auch Klassen über Meta Attribute weiter beschrieben werden. Folgende Meta Attribute haben wir bisher identifiziert:

- Wieviele Entitäten sind zu erwarten?
- Ist die Klasse vom PCM (z.B. Hybris) vorgegeben?

#### 5.1.4 DatastreamToChannel

Offene Fragen

Bevor die Datenströme zu den Channels sinnvoll modelliert werden können, müssen die offenen Fragen zu den Channels geklärt werden.

#### 5.1.5 Channel

Offene Fragen

Für die Modellierung der Channels gibt es noch weitreichende offene Fragen:

- Haben die Channels eine eigene Datenstruktur oder benutzen sie dieselbe wie das MCM? (Wir haben das so verstanden, dass genau die einheitliche und vollständige Datenstruktur die Idee des MCM ist. Fraglich wäre noch, ob eins zu eins dieselbe Struktur verwendet wird, oder ob nicht gebrauchte Daten weggelassen werden.)
- Haben die Channels eine eigene Datenhaltung (wo sie z.B. Werte cachen können) oder greifen sie immer über das MCM auf die Daten zu?
- Können die Channels eigene Werte speichern, welche nicht ins MCM zurückgespielt werden? (z.B. spezielle Preise für einzelne Channels, temporäre Texte für Aktionen, ...)

## 5.2 Touchpoints

Wage Vorstellungen

Über die Touchpoints haben wir erst sehr wenig erfahren; entsprechend haben wir uns auch noch keine konkreteren Gedanken darüber gemacht, wie sie modelliert werden könnten.

Mögliche UML Diagramme

Wir gehen davon aus, dass sie, wie auch die Umsysteme, als Services modelliert werden können. Für zustandslose Services müsste man das Klassendiagramm und Use Case Diagramm (für die Übersicht) prüfen. Für zustandsbehaftete Services wären ausserdem das Zustandsdiagramm, das Sequenzdiagramm oder auch das Aktivitätsdiagramm zu prüfen.

Modellierung als Landkarte

Unabhängig von UML könnten die Touchpoints auch als Graf modelliert werden: Die Touchpoints wären die Knoten und die einzelnen Schritte der Customer Journeys wären die Kanten. Bei den Knoten könnte man angeben, welche Informationen als Input gebraucht werden und welche als Output dem Kunden gegeben werden. Somit liesse sich auch prüfen, ob der Customer Journey überhaupt funktioniert, oder ob der Kunde nicht alle benötigten Informationen zu den Touchpoints mitbringt.

## 5.3 User

Verschiedene Rollen

Es gibt verschiedene User welche in verschiedenen Rollen an das System herantreten. Es gilt abzuklären, welche Rollen relevant sind für die Produktstrukturierung und was für die Rollen speziell modelliert werden soll.

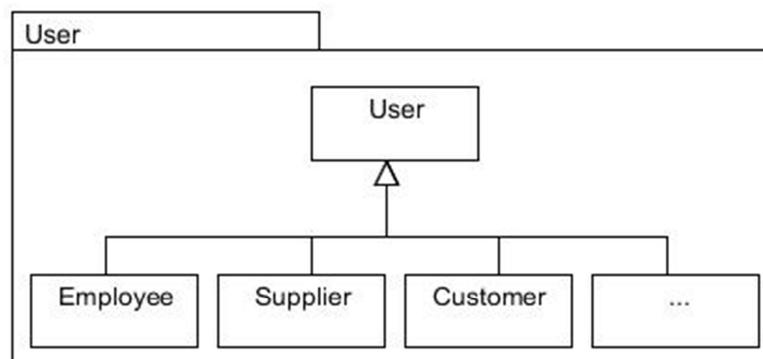


Abbildung 4 Mögliche User des Systems

Abhängigkeiten

Es ist abzuklären, welche Abhängigkeiten zum Rest des Systems modelliert werden müssen. Können die User alle ihre Use Cases über die Touchpoints erledigen? Müssen für Prozesse die beteiligten User referenziert werden? Brauchen gewisse User direkten Zugriff auf die Umsysteme oder das MCM?

Als Ausgangspunkt einer solchen Modellierung drängt sich natürlich ein Use Case Diagramm auf. Möglich wäre auch das Modellieren von Zugriffsrechten: Welcher Use Case muss auf welchen Systemen welche CRUD-Operationen (Create, Read, Update, Delete) durchführen, bzw. welche Services nutzen.

## 5.4 Weitere Konzepte

- Weitere Konzepte Folgende Konzepte können wir uns vorstellen, dass sie ebenfalls modelliert werden könnten:
- Software Prozesse: Welche Software Prozesse gibt es? Mit welchen Verantwortungen? Z.B. für die Datenströme.
  - Business Prozesse: Welche Business Prozesse gibt es? Wer sind die beteiligten User? Wie sehen sie aus?
  - Hardware: Welche Systeme und Prozesse laufen auf welcher Hardware? Über welche Netzwerke sind die Computer miteinander verbunden? Die Hardware würde in einem eigenen Layer (unterhalb des System Layers) modelliert.

## 5.5 Metadaten

- Unterteilung Die Metadaten kann man grundsätzlich in drei Bereiche Aufteilen:
- Informationen zum Entwicklungsprozess
    - Kommentare
    - Wer ist Ansprechperson / hat vertieftes Know-How zu diesem Eintrag
    - ...
  - Informationen zur Einführung des Systems
    - Wer kann den initialen Wert eines Attributes liefern
    - Wieviel kostet die initiale Recherche eines Attributes
    - ...
  - Informationen zum Unterhalt des Systems
    - Welcher Mitarbeiter ist für ein Attribut verantwortlich
    - Wieviel kostet der Unterhalt eines Attributes
    - ...

## Modellierung

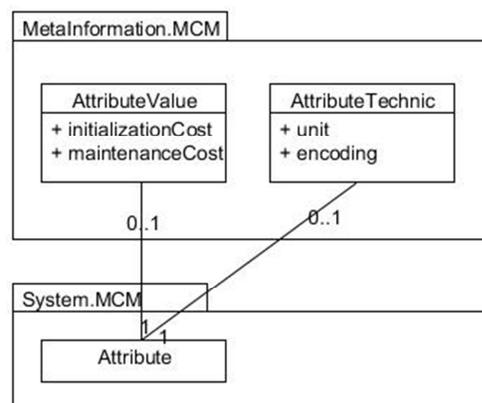


Abbildung 5 Beispiel von Metadaten

Die meisten Metadaten haben eine eindeutige Zuordnung zu einer Klasse aus dem System Layer. Es drängt sich natürlich die Frage auf, ob man diese Metaattribute nicht gleich in der System-Klasse einfügen soll. Für eine saubere Strukturierung ist es natürlich schöner, wenn diese beiden Layer getrennt sind: Im

System Layer diejenigen Infos, welche für die reale, technische Umsetzung benötigt werden, und im Meta Layer zusätzliche Infos.

## 6 Visualisierungen

Wir haben uns diverse Visualisierungs-Möglichkeiten zu verschiedenen Problemen überlegt.

### 6.1 Konzepte

Übersicht

Das PIM steht in der Mitte. Links sind alle alten, schon vorhandenen Systeme (Umsysteme) aufgelistet, rechts die sogenannten Kanäle wie zum Beispiel ein Webshop. Mit einem Klick auf ein System, gelangt man auf dessen Detailansicht.

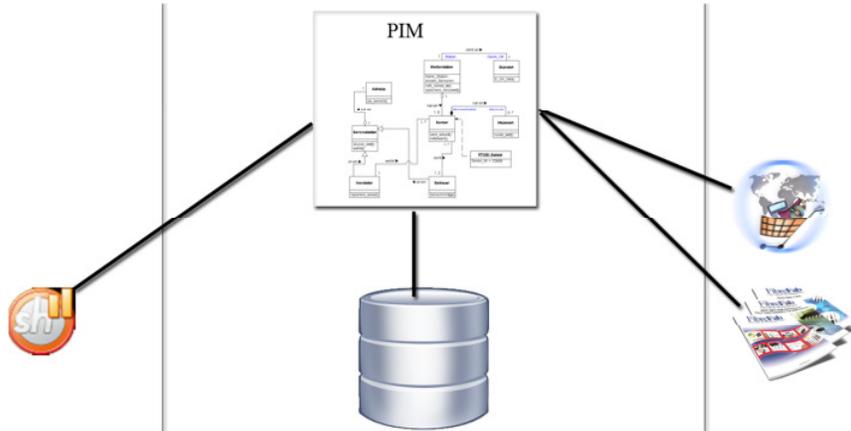


Abbildung 6 PIM Übersicht

Links - Rechts

Dies ist eines der wichtigsten Konzepte. Mit diesem ist es möglich, Systeme miteinander zu vergleichen.

#### Attribute vs. Kanäle

Hier werden die Attribute eines Systems den Kanälen gegenübergestellt. Für jede Verbindung / Datenstrom werden diverse Metadaten gespeichert wie etwa die Wichtigkeit einer solchen Verbindung, die Richtung (bi- oder unidirektional) oder die Kosten. Mit Hilfe von geeigneten Filtern ist es möglich nur die Daten anzuzeigen, welche im Moment von Interesse sind.

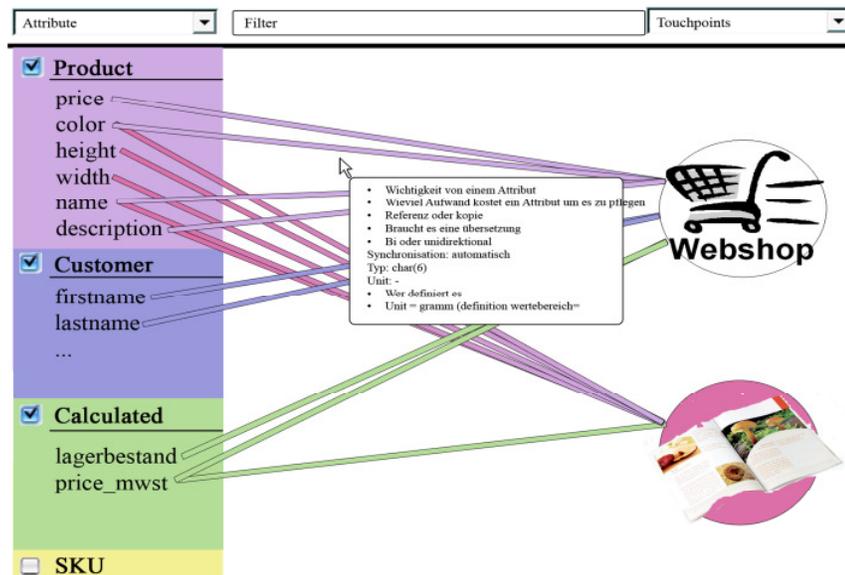


Abbildung 7 Gegenüberstellung Attribute und Kanäle

## Attribute vs. Kanal

Diese Ansicht eignet sich um mit einem Kanalgestalter (zum Beispiel einem Webdesigner) zu sprechen. Auf der rechten Seite kann ein Prototyp eines Kanals angezeigt werden. So kann man diskutieren welche Attribute man wo anzeigen möchte und welche Informationen überhaupt nötig sind. Auch Darstellungsinformationen, die eventuell in anderen Kanälen nicht benötigt werden, können so schnell erkannt werden.

The screenshot shows a real estate listing interface. On the left, a list of attributes for 'Kauf Haus' is shown, including 'Gemeinde (PLZ)', 'Strasse', 'Hausnummer', 'Lage', 'Zusatnd', 'Baujahr', 'Wohnfläche', 'Terrassenfläche', 'Kellerfläche', 'Anzahl Wohnräume', 'Schlafzimmer Anzahl', 'Anzahl WC', 'Anzahl Nasszellen', 'Hobbyraum', 'Gartenanteil', 'Heizungstyp (Zentralheizung | Ele...', 'Anzahl Aussenparkplätze', 'Anzahl Garagenplätze', 'Rollstuhlgängig', 'Verfügbar ab', 'Kaufpreis', 'Anzahl Stockwerke', and 'Objektyp'. On the right, the 'Übersicht' section displays details for a property: 'Im Schilf', '8807 Freienbach', 'Seeanstoss', 'Villa', 'Anzahl Zimmer: 12', 'Anzahl Etagen: 3', 'Wohnfläche: 1'000 m2', 'Nutzfläche: 1'500 m2', 'Grundstückfläche: 3'800 m2', 'Raumhöhe: 3 m', 'Baujahr: 2012', 'Kubatur: 6'000 m3', and 'Verfügbar: 30.11.2012'. The 'Verkaufspreis: auf Anfrage' and 'Lage' sections are also visible. A blue line connects the 'Rollstuhlgängig' attribute to the 'Kubatur: 6'000 m3' value.

Abbildung 8 Gegenüberstellung Attribute und Kanal

## Attribute

Das Finden der richtigen Attribute ist eine zentrale Aufgabe in der Produktstrukturierung. Doch welches sind die richtigen Attribute? Eine Reihe von Metaattributen soll bei dieser Entscheidung helfen. Hier werden alle Metaattribute aller Attribute aufgelistet. Man sieht zum Beispiel auf einen Blick wieviel Aufwand es bedeuten würde, wenn man ein entsprechendes Attribut für jedes Produkt ausfüllen würde.

<b>baujahr</b>	Wichtigkeit: <b>Sehr wichtig</b> Fusion: <b>Kopie/Bidirektional</b> Datentyp: <b>Jahre</b>	Aufwand: <b>0.1h</b> Synchronisation: <b>manuell</b>	Berechnet: <b>✓</b> C: <b>Content</b>
<b>Adresse</b>	Wichtigkeit: <b>Sehr wichtig</b> Fusion: <b>Kopie/Bidirektional</b> Datentyp: <b>Adresse</b>	Aufwand: <b>0.1h</b> Synchronisation: <b>manuell</b>	Berechnet: <b>✓</b> C: <b>Content</b>
	<div style="border: 1px solid black; padding: 5px;">           Hausnr: Haur            Strasse: Strasse            PLZ: PLZ            Typ: int            Range: 0000-9999         </div>		
<b>Objektyp</b>	Wichtigkeit: <b>Sehr wichtig</b> Fusion: <b>Kopie/Bidirektional</b> Datentyp: <b>Objektyp</b>	Aufwand: <b>0.1h</b> Synchronisation: <b>manuell</b>	Berechnet: <b>✗</b> C: <b>Content</b>
	Typ: Enum [Haus, Wohnung]		
<b>UnterObjektyp</b>	Wichtigkeit: <b>Sehr wichtig</b> Fusion: <b>Kopie/Bidirektional</b> Datentyp: <b>UnterObjektyp</b>	Aufwand: <b>0.1h</b> Synchronisation: <b>manuell</b>	Berechnet: <b>✗</b> C: <b>Content</b>
	? Enum ist abhängig von Objektyp wert		

Abbildung 9 Auflistung Metaattribute

## Statistik

Diese Ansicht ermöglicht es die Metaattribute einzelner Attribute zusammengefasst anzuzeigen. Man sieht zum Beispiel wieviele Attribute internationalisierbar sind oder sein müssen.

## Internationalisierbar:



## Aufwand:

Einmaliger Aufwand: 10h  
Aufwand / Woche: 20h

## Die drei C:



Abbildung 10 Statistik

## 7 Bedienkonzepte

### 7.1 Visualisierung, Navigation, Editieren

Problem	Es geht um eine grundsätzliche Frage des Userinterface Designs: Wie sollen die Visualisierung des Modells, die Editiermöglichkeiten und die Navigation durch die verschiedenen Visualisierungs- und Editiermöglichkeiten zusammen spielen.
Zielkonflikt	<ul style="list-style-type: none"> <li>• <b>Einheitliches und Einfaches UI:</b> Wir möchten ein einfaches, intuitiv bedienbares Userinterface, welches gleichzeitig viele Möglichkeiten bietet. Dies bedingt, dass die Bedienkonzepte durch das ganze Programm hinweg einheitlich gestaltet sind und nicht bei jeder Darstellung einzeln gelernt werden müssen.</li> <li>• <b>Erweiterbar:</b> Die Entwicklung von PSA wird in grossem Masse iterativ verlaufen; es werden mit grosser Sicherheit während der Entwicklung neue Ideen auftauchen für die problemgerechte Visualisierung der Produktstruktur. Das Konzept der Bedienung sollte möglichst keine "Vorschriften" für neue Anzeigevarianten machen. Nur so können flexibel und kreativ neue Visualisierungen problemlos in PSA eingebaut werden.</li> <li>• <b>Mächtig:</b> PSA soll ein Modellierungswerkzeug sein. Die Manipulation der Daten ist wichtig. Insbesondere wenn man durch eine gute Darstellung des Modells Verbesserungsmöglichkeiten sieht, möchte man diese auch möglichst einfach eingeben können.</li> </ul>

### 7.2 Variante: Verschiedene Tools für Visualisierung und Datenmanipulation

Beschreibung	<p>PSA wird in verschiedene Tool unterteilt:</p> <ul style="list-style-type: none"> <li>• <b>Editor:</b> Der Editor bietet die Möglichkeit das Modell der Produktstruktur zu manipulieren. Natürlich müssen hierzu auch die bereits vorhandenen Daten angezeigt werden. Die Anzeigen sind aber eher einfach und primär darauf ausgelegt die Daten zu ergänzen und zu ändern. Auch Bedienkonzepte wie z.B. Filtern der Daten gäbe es im Editor nicht.</li> <li>• <b>Präsentation:</b> Das Präsentationstool bietet die Möglichkeit, das Produktstrukturierungsmodell anzuzeigen. Es dient dem Entwickler das Modell explorativ zu verstehen und wird eingesetzt um mit den Kunden verschiedene Perspektiven des Modells anzuschauen.</li> </ul>
Vorteile	<ul style="list-style-type: none"> <li>• Durch die Trennung der Problembereiche kann die Komplexität des Userinterfaces verkleinert werden.</li> <li>• Die Flexibilität für neue Visualisierungen ist maximal.</li> <li>• Es können für verschiedene Devices optimierte Präsentationstools entwickelt werden. So wäre z.B. ein iPad Client vorstellbar. (Während ein Editor auf dem iPad wohl eher unpraktisch wäre.)</li> <li>• Dem Kunden kann man das Tool für die Präsentation zur Verfügung stellen. Somit kann er auch nach Abschluss des Projekts die Dokumentation nutzen. Trotzdem muss man ihm nicht das ganze PSA geben.</li> </ul>

- Nachteile
- Wenn auf einer Darstellung Mängel des Modells festgestellt werden, muss man in einem anderen Programm die Daten editieren. Hierzu muss zuerst an die entsprechende Stelle navigiert werden.
  - Bei Änderung des Modells hat man nicht gleich eine mächtige Anzeigemöglichkeit, um die neuen Daten zu sehen.

### 7.3 Variante: Total verschmolzen

Beschreibung PSA besteht aus einem Tool. Die Visualisierungen zeigen nicht nur die Daten an, sondern ermöglichen auch die Navigation und das Editieren des Modells. Z.B. könnte man in der Übersicht durch einen Klick auf den Datenstrom auf die Gegenüberstellung der Attribute gelangen.

- Vorteile
- Wenn einem gerade etwas auffällt ist es nicht nötig die Ansicht zu wechseln und man muss sich somit auch nicht neu orientieren.

- Nachteile
- Ein Bedienkonzept ist das Filtern der Daten. Es ist aber wichtig, dass die Dateneingabe trotzdem konsistent bleibt: Es sollte z.B. nicht dasselbe Attribut mehrfach erfasst werden, weil es z.B. durch den Filter gerade ausgeblendet wurde.
  - Es sind auch Ansichten möglich, über welche man die Daten nicht manipulieren kann. Dies würde aber dem Wunsch nach einem einheitlichen Bedienkonzept widersprechen.

### 7.4 Variante: Verschiedene Bereiche innerhalb des Fensters

Beschreibung PSA besteht aus einem Tool. Dieses hat klar getrennte Bereiche für das Editieren und die Visualisierung des Modells. Bei der Navigation sind diese jedoch gekoppelt: Im Editorbereich können jeweils diejenigen Daten manipuliert werden, welche im Anzeigebereich visualisiert werden.

- Vorteile
- Die Einschränkungen auf die Visualisierungen sind klein.
  - Der User hat ein Tool, in dem er alle Arbeiten erledigen kann.

- Nachteile
- Natürlich braucht auch der Editierbereich eine Darstellung der Daten. Es werden wohl ein grosser oder gar mehrere Bildschirme nötig werden.
  - Es sollten jeweils alle Daten, welche in die angezeigte Visualisierung einfließen, auch editiert werden können; diese Daten können jedoch sehr viele und sehr vielseitig werden. Es wird nicht einfach, diese einheitlich und übersichtlich zu organisieren.

## 7.5 Filtern von Daten

Ausgangslage	In verschiedenen Situationen muss das Modell aus verschiedenen Perspektiven betrachtet werden. Dies bedingt jeweils auch eine andere Auswahl der anzuzeigenden Daten. Es ist z.B. wichtig, dass wenn man mit dem Webdesigner spricht diesem nur diejenigen Elemente des Modells gezeigt werden, welche für ihn relevant sind; ansonsten muss man ständig erklären, was ihn nicht zu kümmern braucht.
Datenbasis für die Filter	Für die Daten des eigentlichen Modells wird jeweils auch gespeichert, welche Personen sich für welche Details interessieren. Anhand dieser Informationen könnten dann die entsprechenden Anzeigen automatisch generiert werden.
Änderung der Darstellung	<p>Die Frage ist, wie sich solche Filter auf die Darstellungen auswirken: Werden z.B. in einem UML-Diagramm die irrelevanten Attribute einfach nicht angezeigt oder wird das ganze UML-Diagramm neu aufgebaut. Der Vorteil der ersten Variante ist, dass die Klassen immer noch an derselben Stelle sind; somit muss man sich nach dem Filtern nicht von neuem orientieren. Der Nachteil sind dafür grössere Lücken: Grosse Vierecke für Klassen wo nur wenige Attribute drin stehen oder wenn ganze Klassen fehlen können stören.</p> <p>Eine weitere Variante wäre, die irrelevanten Daten ausgegraut anzuzeigen. Bei sehr vielen Daten wäre man aber vielleicht froh, wenn man nicht in einem sehr grossen Diagramm die relevanten Daten suchen müsste.</p>

## 7.6 Automatisch generierte vs. manuell aufbereitete Visualisierungen

Ausgangslage	Das Modell soll in verschiedenen Darstellungen betrachtet werden können. Soweit möglich sollten die Visualisierungen automatisch generiert werden, damit keine Zeit aufgewendet werden muss um dies manuell zu erledigen. Somit könnte die ganze Energie in das Modellieren des eigentlichen Modells gehen.
Problem	Eine optimale Aussagekraft einer Darstellung benötigt häufig manuelle Optimierung: So kann man z.B. durch geschicktes Anordnen von Klassen in einem Klassendiagramm auch semantische Zusammenhänge verdeutlichen.
Vorteile der automatischen Generierung	<ul style="list-style-type: none"> <li>• Man muss keinen Aufwand betreiben um die diversen Darstellungen zu erstellen.</li> <li>• Nach Eingabe der Daten sind die Visualisierungen sofort verfügbar.</li> <li>• Nach Änderung der Daten müssen nicht auch noch die Visualisierungen angepasst werden.</li> </ul>

- Die Visualisierungen sind durchwegs konsistent aufgebaut.
- Wenn die Daten gefiltert werden, d.h. nur ein Ausschnitt der Daten angezeigt wird, steht jeweils sofort eine Visualisierung bereit.

Vorteile der manuellen Generierung

Durch menschliche Intelligenz können auch semantische Zusammenhänge in die Darstellung einfließen.

Manuelle Anpassung automatisch generierter Ansichten

Eine Variante wäre, die Darstellungen in erster Linie automatisch zu generieren, jedoch die Möglichkeit zu bieten, diese bei Bedarf noch anzupassen. Somit müsste nicht zwingend bei jeder Darstellung Hand angelegt werden, bei ungeschickten Darstellungen könnte aber die computergenerierte Lösung verbessert werden.

Wie sich diese Darstellungen nach einer Veränderung der zugrundeliegenden Daten verhalten, bleibt so aber ungeklärt.

## 7.7 Perspektiven bezogene Präsentationen

Ausgangslage

Das Grundproblem ist dasselbe wie beim Filtern der Daten: In verschiedenen Situationen muss das Modell aus verschiedenen Perspektiven betrachtet werden.

Lösungsansatz

Eine Idee wäre, wenn man Präsentationen erstellen könnte. Eine solche besteht aus mehreren Darstellungen, welche extra für eine spezifische Perspektive aufbereitet werden. Dies beinhaltet sowohl die Auswahl der anzuzeigenden Daten als auch eine Aufbereitung der Darstellungen welche auf die Daten optimiert ist.

Probleme

- Wie verhalten sich die Darstellungen nach Änderungen des Modells?
- Der Aufwand um für alle Anspruchsgruppen eigene Darstellungen von Hand zu erstellen ist unter Umständen gross.

## 7.8 Templates

Idee

Die Idee ist, dass Templates für gewisse Bereiche einer Produktstrukturierung zur Verfügung gestellt werden. Dadurch können zum Einen repetitive Arbeiten verhindert werden und zum Anderen Know-How innerhalb der Firma verbreitet werden.

UML-Modelle

Schön wäre natürlich ein Standard Datenmodell, welches immer wieder verwendet werden kann und nur für den konkreten Fall angepasst werden muss. Bis ein solches entwickelt ist, könnte man auch für Teilbereiche des Datenmodells Templates zur Verfügung stellen.

Standard Tabellen mit standard Werten	Gewisse Bereiche des Datenmodells haben nicht nur immer wieder dieselbe Struktur, sondern auch dieselben Daten. Z.B. Farben und Marketingfarben könnten als Template hinterlegt werden.
Listen	Verschiedene Werte kommen immer wieder vor. So könnte man Listen zur Verfügung stellen, wo diese Werte einfach ausgewählt werden können. Z.B. gibt es häufig auftretende Umsysteme wie ERP, CRM, etc. Oder auch bei den Channels gibt es häufig auftretende Vertreter wie z.B. einen Webshop. Auch für Attribute des Datenmodells könnte man eine Liste von Standardwerten erstellen.

## 7.9 UML Editor

Anzeige	PSA soll verschiedene Konzepte als UML-Modelle visualisieren. Es genügt nicht, wenn UML in anderen Tools erstellt wird und in PSA als JPG angezeigt wird. Wir möchten weitere Inhalte (z.B. Metattribute) anzeigen, oder auch Inhalte filtern. Daher ist es sinnvoll, die UML Diagramme dynamisch zu erzeugen.
Editor	Nicht ganz so einfach ist die Frage, ob PSA einen eigenen UML Editor zur Verfügung stellen soll.
Integrierter Editor	<p>Vorteile:</p> <ul style="list-style-type: none"> <li>• Gute Integration, alles aus einem Guss, man kann alles in einem Tool modellieren</li> <li>• Grosse Flexibilität um neue Features einzubauen (z.B. verschiedene Anordnungen der Klassen wenn nur ein Teil der Attribute und Klassen dargestellt wird)</li> </ul> <p>Nachteile:</p> <ul style="list-style-type: none"> <li>• Aufwand</li> </ul>
UML Diagramme importieren	<p>Vorteile:</p> <ul style="list-style-type: none"> <li>• Es gibt viele gute UML Tools. Man könnte den Entwicklern ermöglichen mit ihrem favorisierten Tool zu arbeiten.</li> </ul> <p>Nachteile:</p> <ul style="list-style-type: none"> <li>• UML ist sehr umfangreich. Sinnvollerweise beschränken wir uns auf ein Subset von UML. Beim Import stellt sich dann die Frage, was mit Elementen geschehen soll, welche von PSA nicht unterstützt werden.</li> <li>• Änderungen am Datenmodell müssten erneut in PSA importiert werden. Es bräuchte einen Automatismus um diese Änderungen zu erkennen und die restlichen Daten an das neue Modell anzupassen.</li> </ul>

## 8 Technologien

Mit welcher Technologie dieses Projekt umgesetzt wird ist noch nicht entschieden. foryouandyourcustomers wünscht sich eine Lösung in Java oder C#.

Mit Java haben wir bis jetzt nur wenige Erfahrungen im Bereich GUI sammeln können. Swing haben wir erst einmal eingesetzt und fühlen uns nicht so wohl.

C# kennen wir nur von dem Modul Microsoft-Technologien. Das MVVM-Pattern hat einen positiven Eindruck hinterlassen.

## 9 Bachelorarbeit

Stand heute

Unser Wissen zum heutigen Zeitpunkt haben wir in diesem Dokument zusammengetragen. Wir haben viel über das Thema Produktstrukturierung gelernt und wir haben einige Ideen entwickelt, wie eine (interaktive) Dokumentation einer Produktstrukturierung aussehen könnte. Dies wird uns in die Lage versetzen, in einer Folgearbeit PSA zu entwickeln. Es wird aber noch viel Input nötig sein, um die Details auch wirklich so zu gestalten, wie sie in einem realen Fall von Nutzen sind.

Welche Probleme sind zu erwarten

Wir sehen heute 3 grössere Probleme:

- Erfahrung in UI-Entwicklung: Weder Michael Steiner noch Christoph Rosenberger haben Erfahrung in der Entwicklung eines UIs. Beide haben nie etwas komplexeres als ein simples Eingabeformular gemacht.

Natürlich kann man Erfahrung durch nichts ersetzen. Trotzdem erhoffen wir uns durch den Besuch des Faches User Interfaces zwei weiteren nützlichen Input. Wir werden bei der Arbeitsplanung darauf achten, dass Entscheide zur Benutzerführung so weit möglich eher erst zur Mitte des Semesters getroffen werden müssen.

- Aussagekräftige, gut skalierende Visualisierungen: Gute Visualisierungen zu finden, welche Aussagen gut und schnell herüberbringen, ist nicht immer einfach und braucht manchmal das Glück einer guten Idee. Auch unsere bisherigen Ansätze müssen noch mit realen Daten und Datenmengen getestet werden.

Hier sehen wir keinen besseren Lösungsansatz als trial and error: Wir werden unsere Ideen umsetzen und anhand des Feedbacks von foryouandyourcustomers sehen, ob sich unsere Visualisierungen eignen. Andererseits haben wir bei foryouandyourcustomers äusserst kreative Köpfe kennen gelernt, so dass wir diesem Problem eher gelassen entgegen sehen.

- Konkreter Input: Wir haben zwar viel über Produktstrukturierung gelernt; trotzdem wissen wir nicht, welche Probleme in einem realen Fall zu lösen sind, welche Details modelliert werden müssen, etc. Hier sind wir auf den Input von foryouandyourcustomers angewiesen. Leider dürfen wir aber nicht davon ausgehen, dass immer eine geeignete Person für uns zur Verfügung steht um mit uns die nötigen Details zu diskutieren.

Um auf dieses Problem zu reagieren, haben wir drei verschiedene Strategien. Alle basieren auf einem agilen, iterativen Prozessmodell:

- Bei der Planung für eine Iteration werden wir das Thema jeweils den verfügbaren Personen anpassen: Je nachdem, welches die Interessen

- und Stärken der Person sind, werden wir PSA in diese Richtung weiter entwickeln.
- Wenn möglich sollten wir Themenbereiche angehen, welche ein begeisterndes Resultat versprechen. Hierdurch erhoffen wir uns eine gesteigerte Motivation bei den Mitarbeitern von foryouandyourcustomers für unser Projekt.
  - Wir sehen Iterationen vor, für welche wir kein oder nur wenig Input seitens foryouandyourcustomers brauchen. Dies könnten z.B. Code Refactorings oder UI Überarbeitungsiterationen sein.
- Sollten alle diese Ansätze nicht funktionieren, müssten wir auf die äusserst ineffiziente trial and error Strategie ausweichen: Wir programmieren das was wir uns in unsere Fantasie als nützlich vorstellen, und passen es dann entsprechend dem Feedback seitens foryouandyourcustomers an.

#### Ziele der Arbeit

Unser Ziel für die Bachelorarbeit ist ein Tool, welches bei der Entwicklung und Dokumentation einer Produktstrukturierung hilft. Welche Themengebiete wir genau bearbeiten, lässt sich kaum planen; zu sehen ist aber, dass sicherlich nicht die ganze Welt von den Umsystemen über das MCM mit seiner Datenstruktur, Klassifizierung und Kategorisierung bis hin zu den Touchpoints und Customer Journeys sauber und detailliert abgebildet werden können.

Unser Ziel ist, dass die behandelten Themenbereiche gut abgebildet werden und die Software Produktionsqualität hat. Dies bezieht sich auch auf die Codequalität: Nicht nur halten wir Abstürze eines "selbst entwickelten" Tools bei Kunden für inakzeptabel, wir möchten auch ermöglichen, dass PSA von anderen Entwicklern weiterentwickelt werden kann.

#### Grundsätzliches Vorgehen

Wir denken es macht für dieses Projekt in grossem Masse Sinn, nach den agilen Grundsätzen vorzugehen. Features wie Multiuser- oder Netzwerkfähigkeit sollten nicht schon am Anfang eingebaut werden, sondern erst dann und nur dann, wenn dies von foryouandyourcustomers auch wirklich so gewünscht wird.

Auch beim Thema UML Import vs. eigenen UML Editor programmieren können wir problemlos zuerst einen einfachen Import vorsehen. Wird später festgestellt, dass dies nicht genügt, kann immer noch problemlos ein eigener Editor programmiert werden.

Die Navigation und das Editieren der Daten möchten wir zu Beginn so weit wie möglich trennen. Dies hält uns eine möglichst grosse Flexibilität bei der Entwicklung der Visualisierungen offen. Sollte sich dies als nicht praktikabel erweisen, können wir im mittleren Drittel Iterationen benützen, um eine konsistente, benutzerfreundliche Bedienung des Tools zu entwickeln.

#### Elaboration

Um mit dem Projekt zu beginnen eignet sich insbesondere das Datenmodell des MCM und einige dazugehörige Metadaten. Hierzu brauchen wir keine weiteren Inputs. Wir müssen nicht viel

programmieren (das UML Klassenmodell können wir importieren). Trotzdem haben wir eine erste lauffähige Programmversion welche durch sämtliche Daten Layer und Technischen Layer durchgeht.

Ausserdem ist das Datenmodell unserem Verständnis nach der Dreh- und Angelpunkt der ganzen Produktstrukturierung. Von hier aus können wir in alle Richtungen weiter entwickeln: Seien dies inhaltliche Ergänzungen, z.B. mit den Umsystemen, weiteren Bedienkonzepten wie z.B. Templates für die Datenmodelle, etc.

## 10 Prototyp

Ziele	<p>Mit Hilfe des Prototyps möchten wir den Mitarbeitern der Firma foryouandyourcustomers möglichst konkret unsere Überlegungen und Ideen vorstellen. Mit unseren Bildern konnten wir schon zu guten Diskussionen anregen. Jedoch war es nie möglich einen Filter zu setzen oder etwas hinzuzufügen. So musste man sich immer überlegen was noch möglich wäre und was nicht. Man war also immer noch in einer Gedankenwelt. Dank des Prototyps haben wir es schwarz auf weiss und die Mitarbeiter von foryouandyourcustomers sehen was wir effektiv haben und was nicht.</p>
Technologien	<p>Aus zeitlichen Gründen entschieden wir uns den Prototyp mit PHP und MySQL zu entwickeln. Als Template-Engine verwenden wir Smarty. Einfache Konzepte werden mit HTML, CSS und Javascript dargestellt. Komplexere Darstellungen werden (aus pragmatischen Gründen) auf dem Server als Image gerendert.</p> <p>Natürlich ist für eine produktive Version nicht PHP vorgesehen sondern Java oder C#.</p>

## 10.1 Klassendiagramm

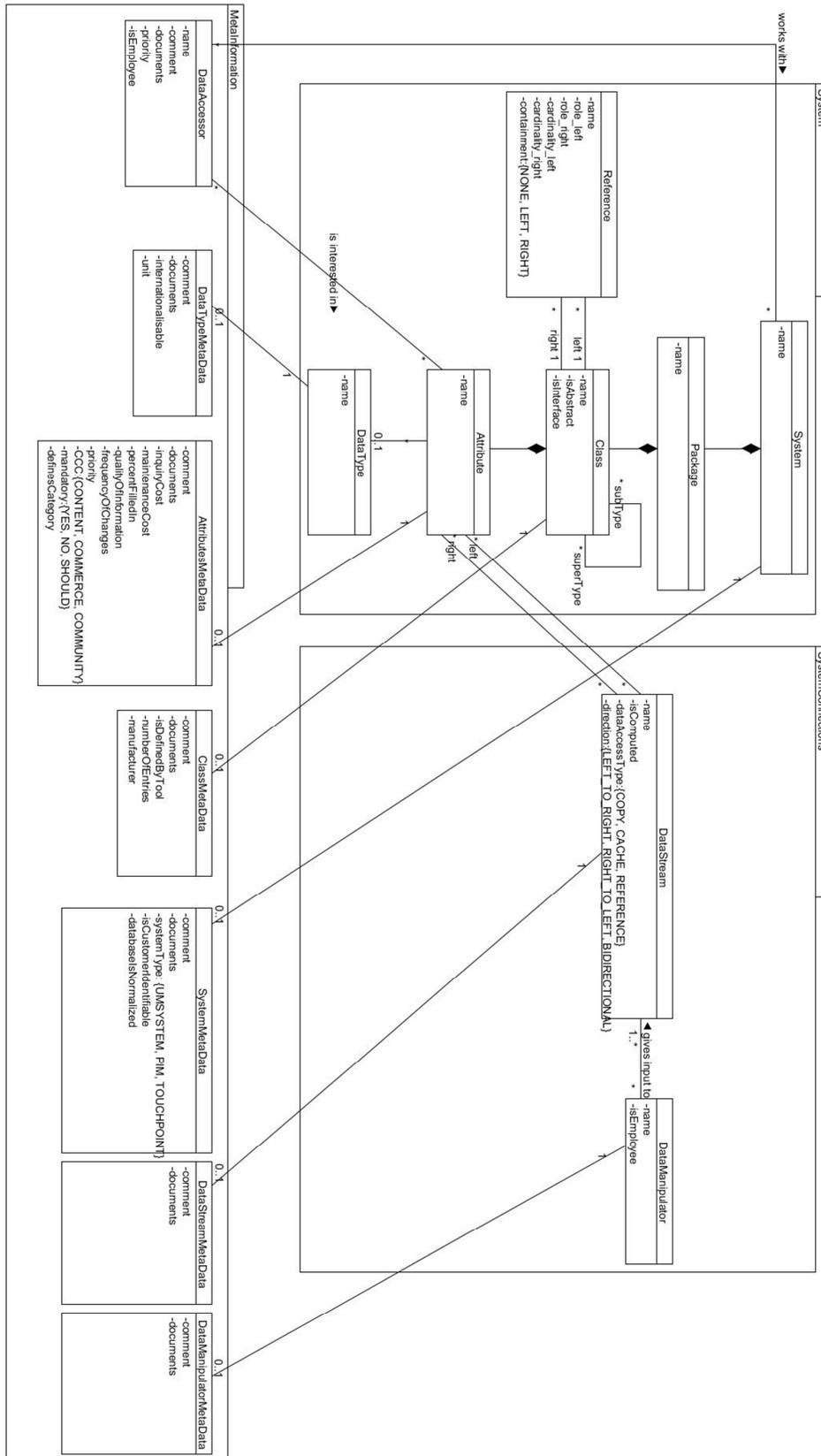


Abbildung 11 Prototyp Klassendiagramm

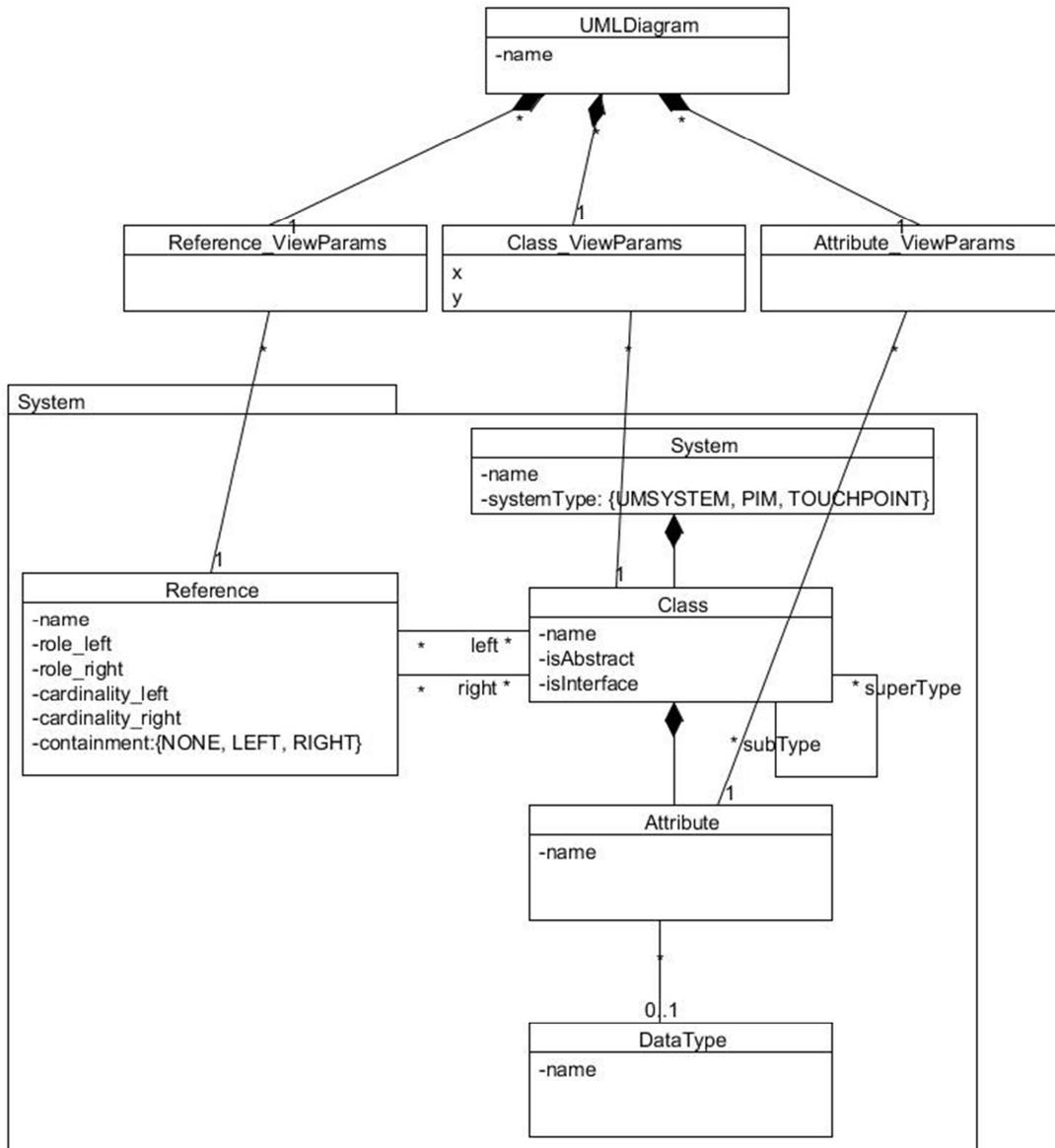


Abbildung 12 Prototyp UML Klassendiagramm

## 10.2 Darstellungen

### Übersicht

Die Übersicht zeigt alle Umsysteme, das PIM im Zentrum und die Kanäle. Des Weiteren sieht man auf einen Blick, wie diese zusammenhängen und man hat auch schon einen kleinen Einblick in den Datenfluss zwischen den einzelnen Systemen.

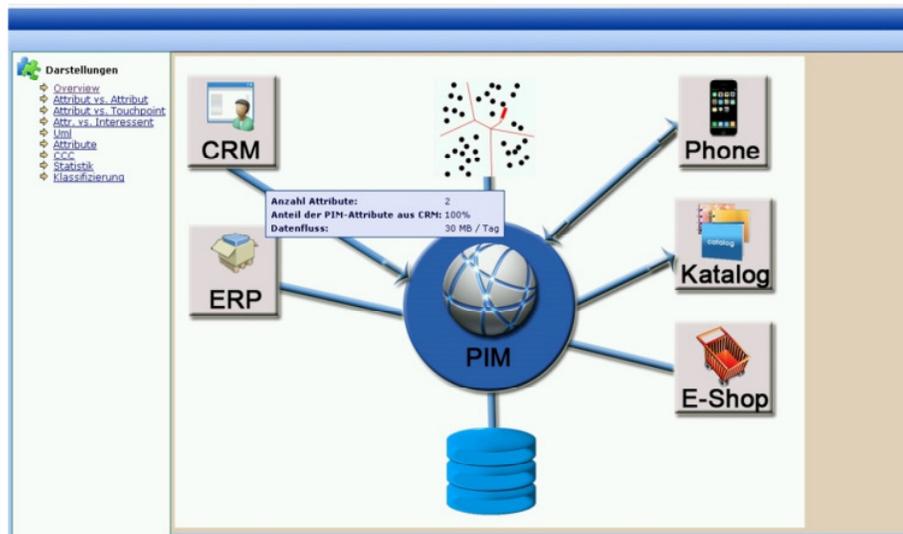


Abbildung 13 Prototyp Übersicht

### UML

Unser Prototyp ist fähig, UML-Dateien, welche mit Umllet (<http://www.umlet.com>) erstellt wurden, zu importieren. Dies ermöglicht es, Systeme relativ schnell abzubilden.

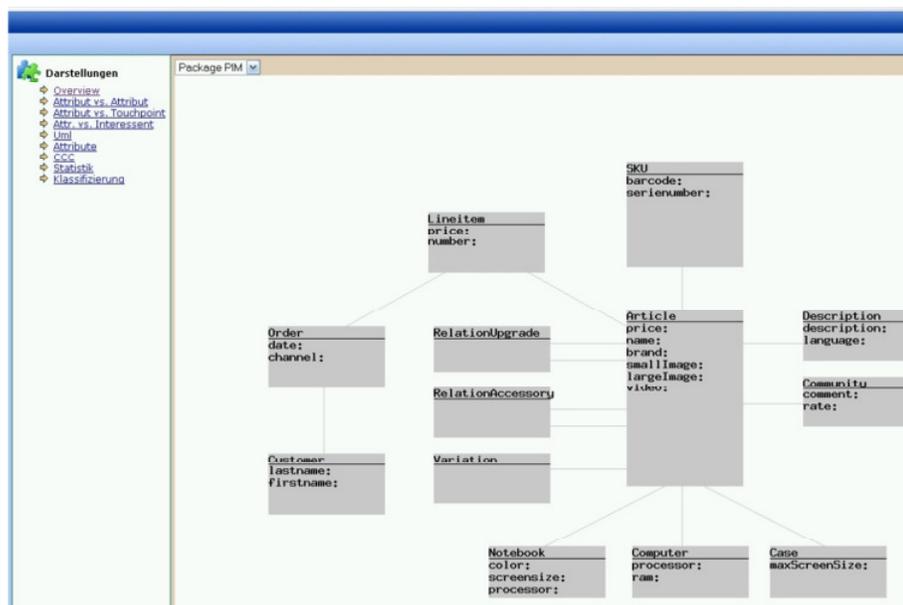


Abbildung 14 Prototyp UML

## Attribute

Wurden die UML-Beschreibungen importiert, können jetzt Attribute von verschiedenen Systemen miteinander verknüpft werden. Zusätzlich können Metaattribute definiert werden, welche die Verknüpfung der Attribute beschreiben (z.B. ob es sich um eine bidirektionale Verbindung handelt).

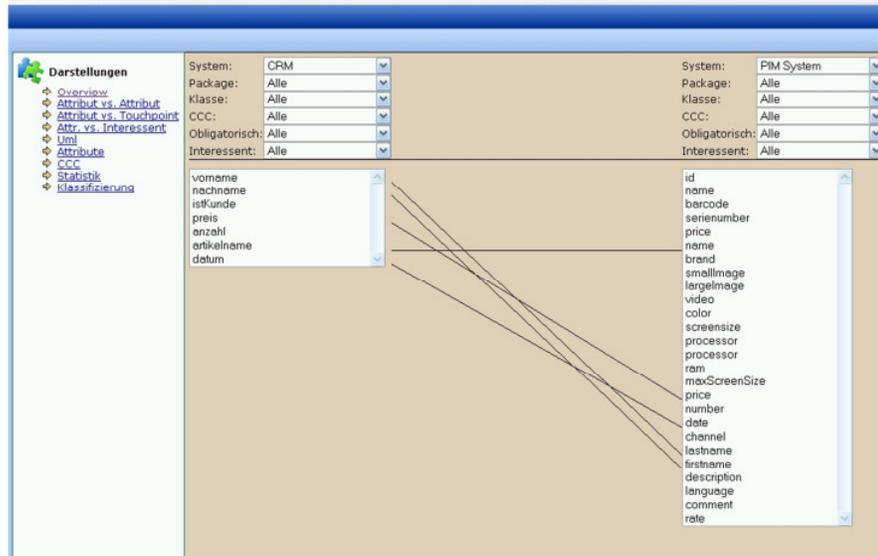


Abbildung 15 Prototyp Attribut Gegenüberstellung

## Kanal

Auch eine visuelle Darstellung eines Kanals (z.B. ein Webshop) lässt sich in der „Links-Rechts-Darstellung“ anzeigen. Dies ermöglicht eine gute Übersicht über die Attribute, welche in einem Kanal überhaupt benötigt werden. Es lässt sich auch schnell ein Überblick über fehlende Attribute wie zum Beispiel Darstellungsinformationen, welche ein Website-Designer benötigt, gewinnen.

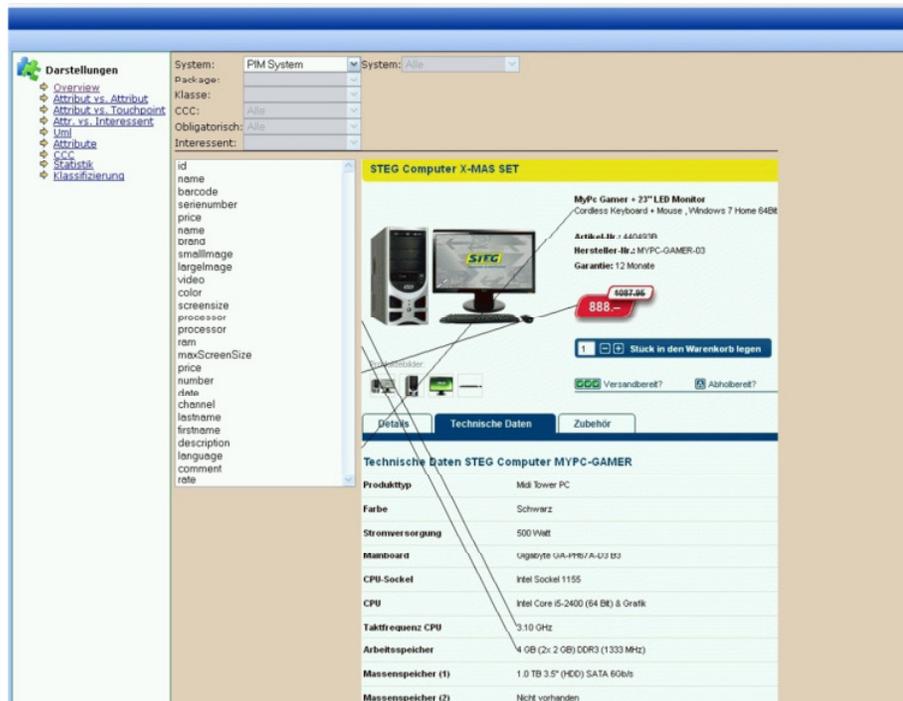


Abbildung 16 Prototyp Gegenüberstellung Attribute und Kanal

Klassifizierung

Hier wird die Klassifizierung dargestellt. Mit geeigneten Filtern ist es auch hier möglich, nur die Bereiche, welche aktuell interessieren, anzuzeigen.

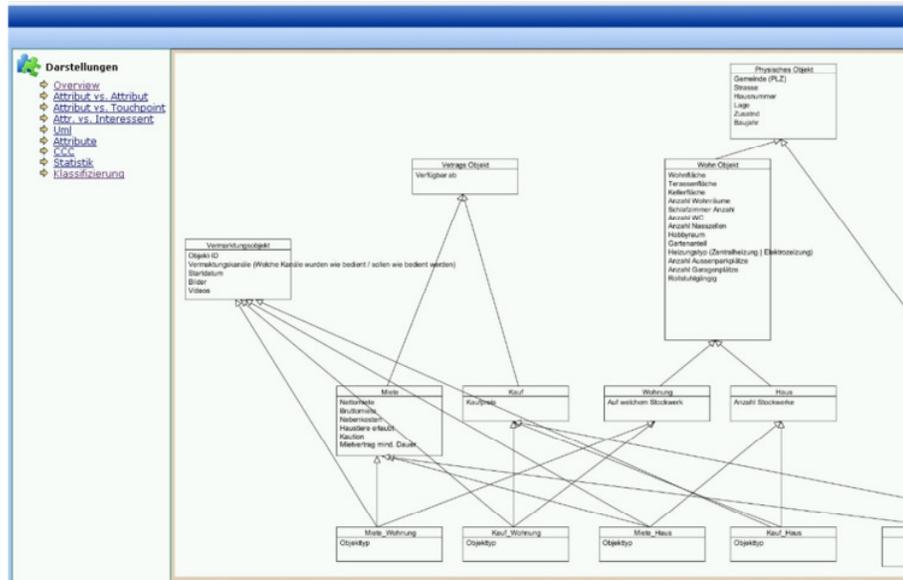


Abbildung 17 Prototyp Klassifizierung

CCC

Hier werden alle Attribute nach den drei „C’s“ sortiert.

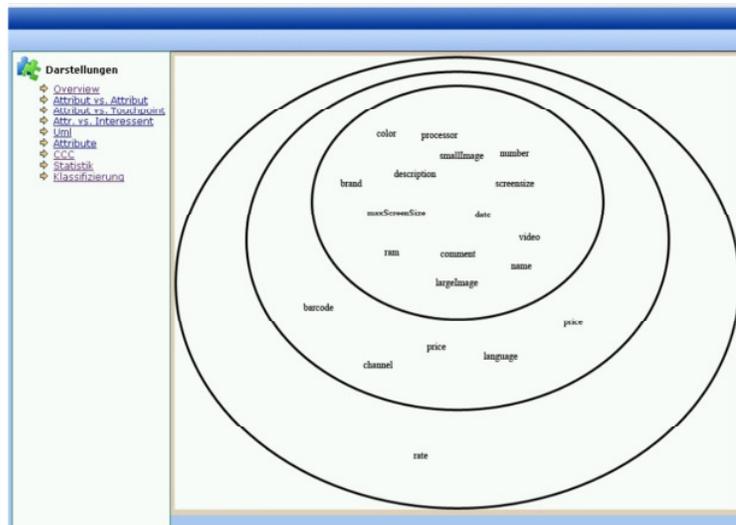


Abbildung 18 Prototyp CCC

Attribute-Detail

Hier werden alle Details (Metaattribute) zu den einzelnen Attributen aufgelistet. Fragen wie „wie wichtig ist ein Attribut“, „was kostet ein Attribut“ und „woher kommen die Daten für dieses Attribut“ werden hier beantwortet.

Darstellungen		Klasse: Bestellposition				
<b>preis</b>	Wichtigkeit: 0 Fusion: Datentyp:	Laufende Kosten: 0 Synchronisation: Obligatorisch: YES	Einmalige Kosten: 0 C: Berechnet	CONTENT	Änderungshäufigkeit: 0 Qualität: 0 Ausgefüllt: 0%	
<b>anzahl</b>	Wichtigkeit: 0 Fusion: Datentyp:	Laufende Kosten: 0 Synchronisation: Obligatorisch: YES	Einmalige Kosten: 0 C: Berechnet		Änderungshäufigkeit: 0 Qualität: 0 Ausgefüllt: 0%	
<b>artikelname</b>	Wichtigkeit: 0 Fusion: Datentyp:	Laufende Kosten: 0 Synchronisation: Obligatorisch: YES	Einmalige Kosten: 0 C: Berechnet	CONTENT	Änderungshäufigkeit: 0 Qualität: 0 Ausgefüllt: 0%	
<b>datum</b>	Wichtigkeit: 0 Fusion: Datentyp:	Laufende Kosten: 0 Synchronisation: Obligatorisch: YES	Einmalige Kosten: 0 C: Berechnet		Änderungshäufigkeit: 0 Qualität: 0 Ausgefüllt: 0%	

Abbildung 19 Prototyp Metaattribute

Interessenten

In dieser Darstellung wird gezeigt, wer sich aus welchem Grund für welche Attribute interessiert (dies kann z.B. ein Endkunde (Persona), ein Webdesigner oder jemand, der die Attribute befüllt, sein).

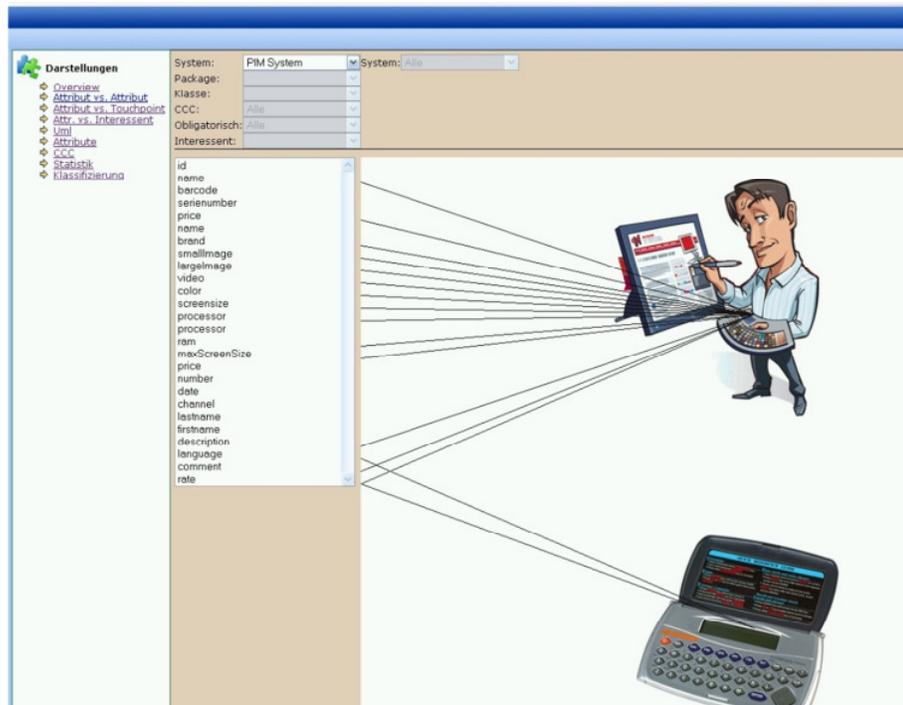


Abbildung 20 Prototyp Interessenten

## Statistik

Dieser Bereich erlaubt es Statistiken über das ganze System oder über Teilbereiche darzustellen. Fragen wie „wie teuer alle Attribute im Unterhalt sind“ werden hier beantwortet.



Abbildung 21 Prototyp Statistik

## 11 Alternative Lösungsansätze

### 11.13D

Der erste Teil unserer Aufgabenstellung bezog sich auf die 3D-Visualisierung von komplexen Produktstrukturen. Wir haben dazu einige Entwürfe gemacht, mussten aber feststellen, dass diese Art von Visualisierung keinen Gewinn bringt.

**Übersicht** Ob die Übersicht in 3D oder 2D dargestellt wird macht eigentlich keinen grossen Unterschied. Die Informationen bleiben die gleichen und der 3D-Effekt bringt mehr Verwirrung als Nutzen.

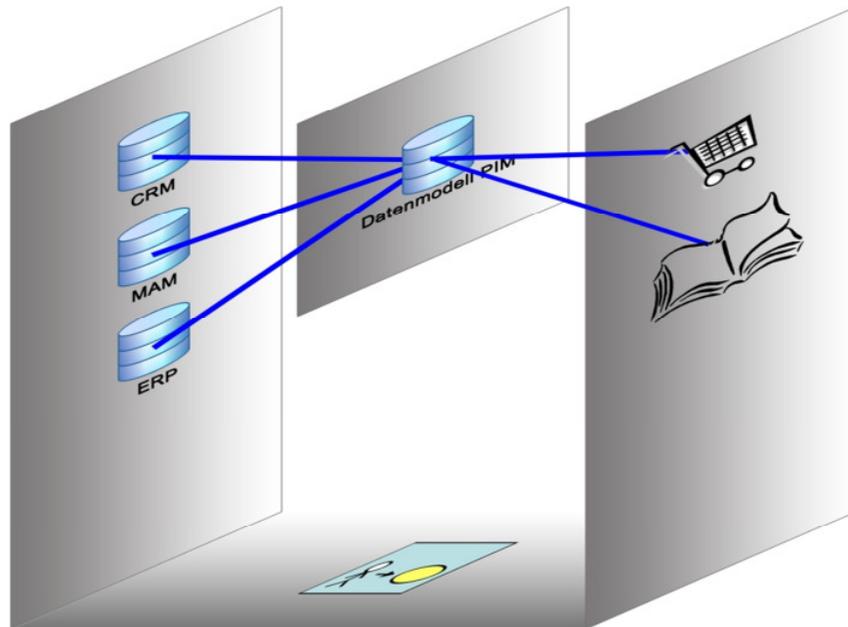


Abbildung 22 3D Übersicht

**Umsysteme & Aktoren**

In dieser Ansicht wollten wir die Umsysteme mit den Aktoren und deren Aktionen verknüpfen. Eine tabellarische Darstellung wäre sinnvoller.

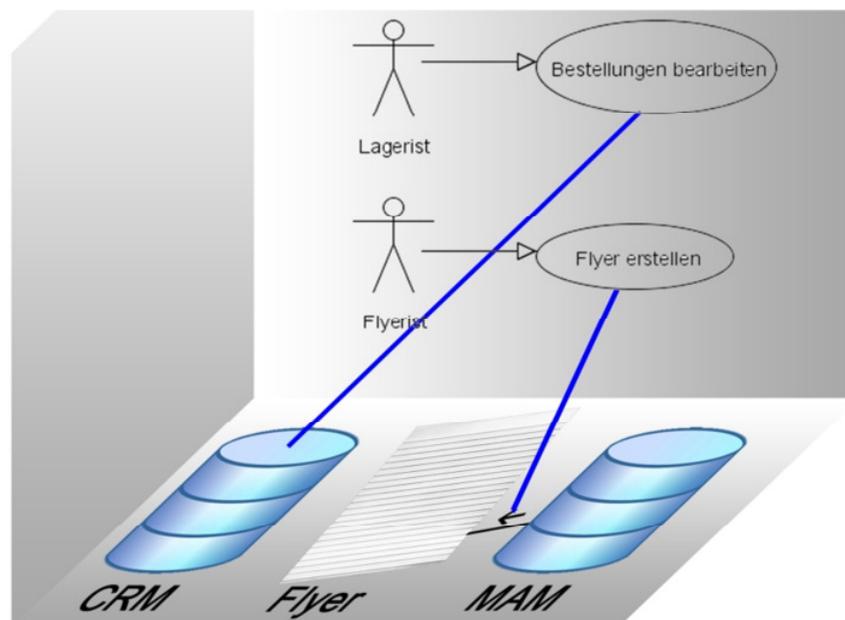


Abbildung 23 3D Umsysteme und Aktoren

Klassifizierung & Attribute

Diese Darstellung zeigt wie man die konkreten Attribute auf einer Klassifizierung abbilden könnte.

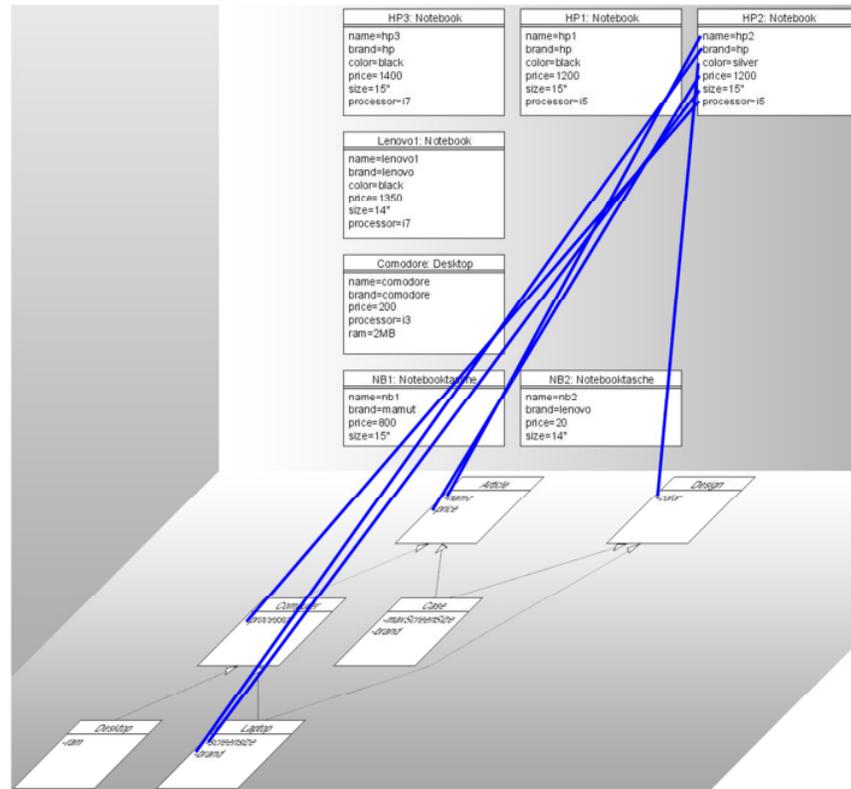


Abbildung 24 3D Klassifizierung und Attribute

Klassifizierung & Datentypen

Diese Darstellung zeigt wie man Datentypen auf einer Klassifizierung abbilden könnte.

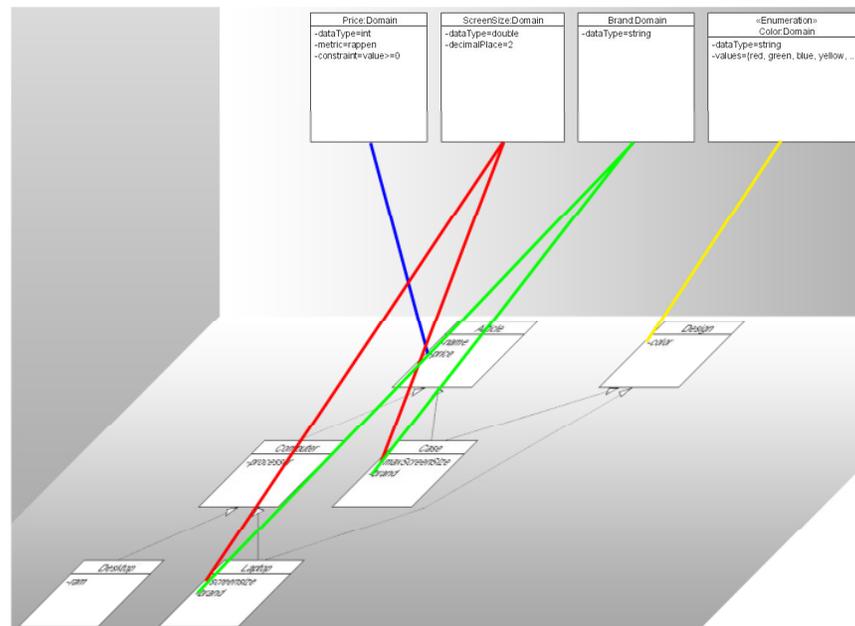


Abbildung 25 3D Klassifizierung und Datentypen

CCC

Hier werden die drei „C“ dargestellt. Man könnte diese Attribute mit den konkreten Attributen verbinden um zu zeigen welche üblichen Attribute noch fehlen. Auch diese Darstellung eignet sich nicht wirklich in dieser Form.

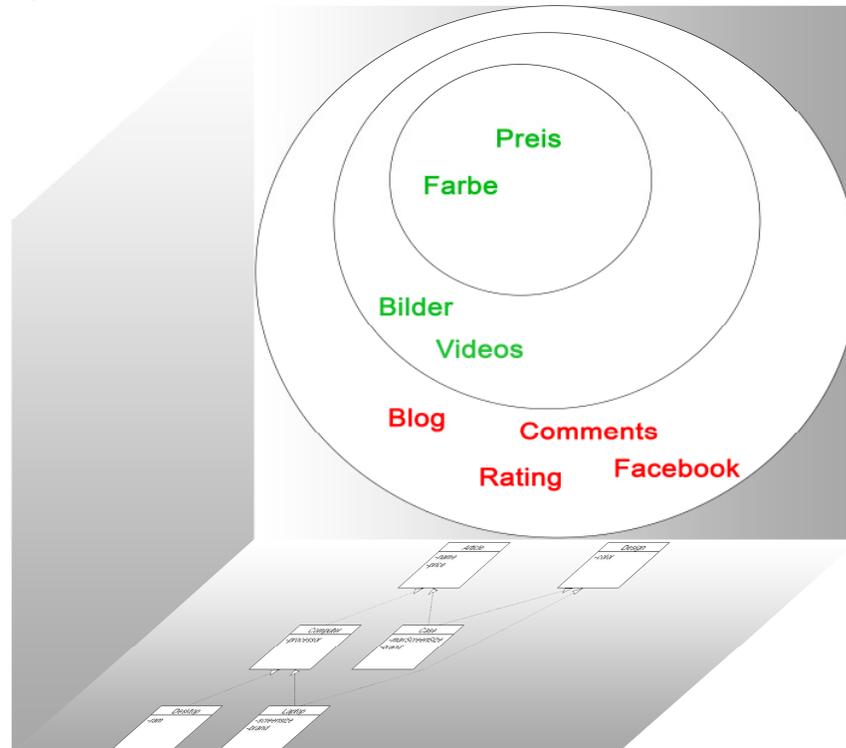


Abbildung 26 3D Drei C

Umsysteme,  
PIM & Aktoren

Links sieht man die Umsysteme und wie sie mit dem PIM (unten) verbunden sind. Hinten sind die Aktoren und ihre Aktionen definiert. Auch diese Grafik würde sich in 2D abbilden lassen, was vermutlich sinnvoller wäre.

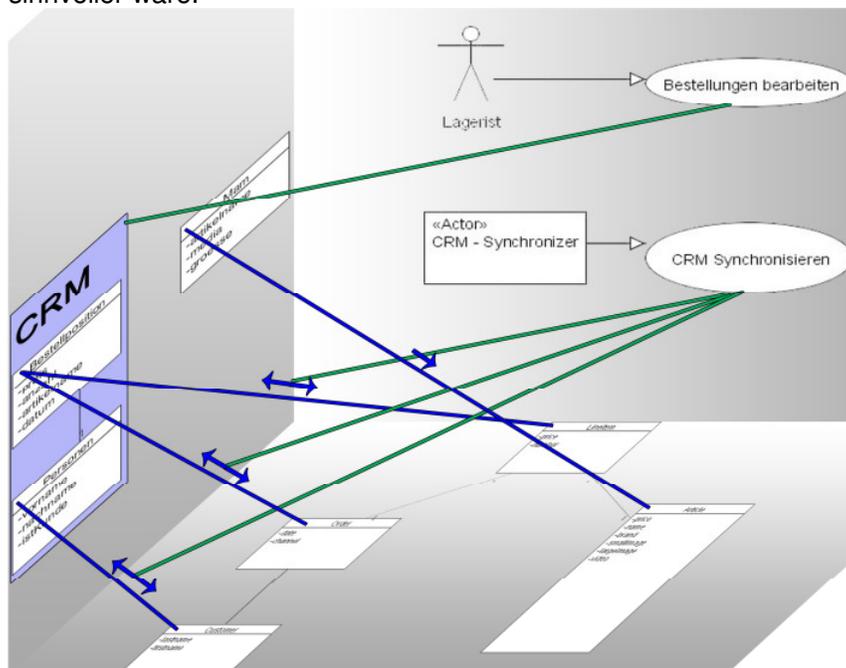


Abbildung 27 3D Umsysteme, PIM, Aktoren

## CAD

Wir haben mal geprüft wie das 3D Verfahren in einem etwas grösseren Projekt aussehen würde. Links an der Wand befinden sich wiederum die Umsysteme. Unten liegt das PIM und rechts die verschiedenen Kanäle. Es sieht zwar spektakulär aus und man kann es sogar noch relativ gut interpretieren. Der Nutzen ist aber auch hier leider nicht zu erkennen.

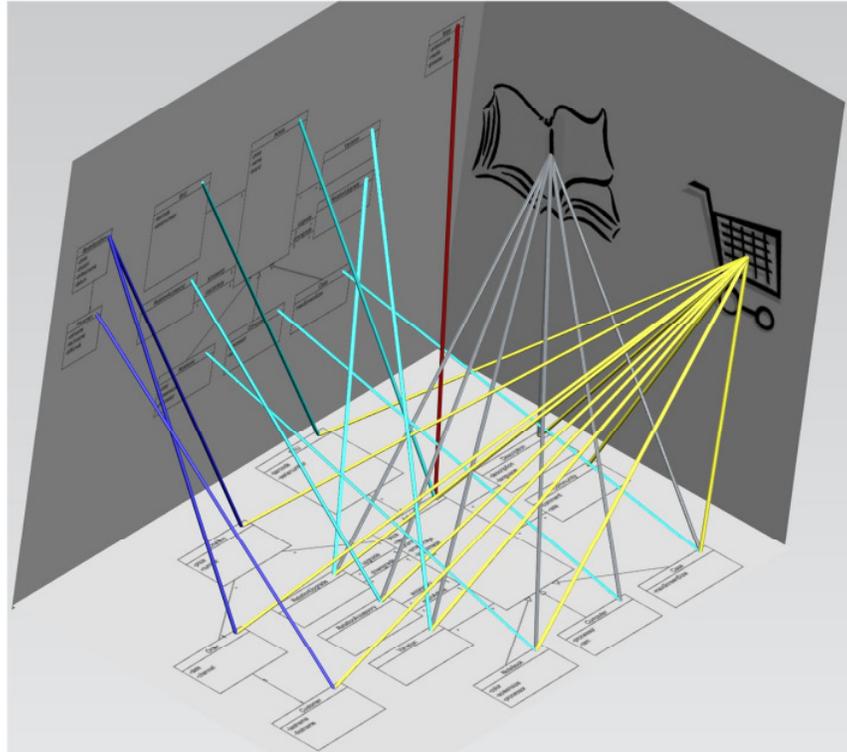


Abbildung 28 3D CAD

### Anaglyphendarstellung

Wir erzeugten aus unseren 3D Darstellungen ein Stereogramm welches mit einer Rot-Grün Brille betrachtet werden kann. Diese Methode erlaubt es sehr viele Informationen übersichtlich auf einen Blick darzustellen. Da aber die 3D Darstellungen allgemein keinen grossen Vorteil bieten und man bei dieser Variante sogar noch eine 3D-Brille benötigt, haben wir diese Variante nicht weiter verfolgt.

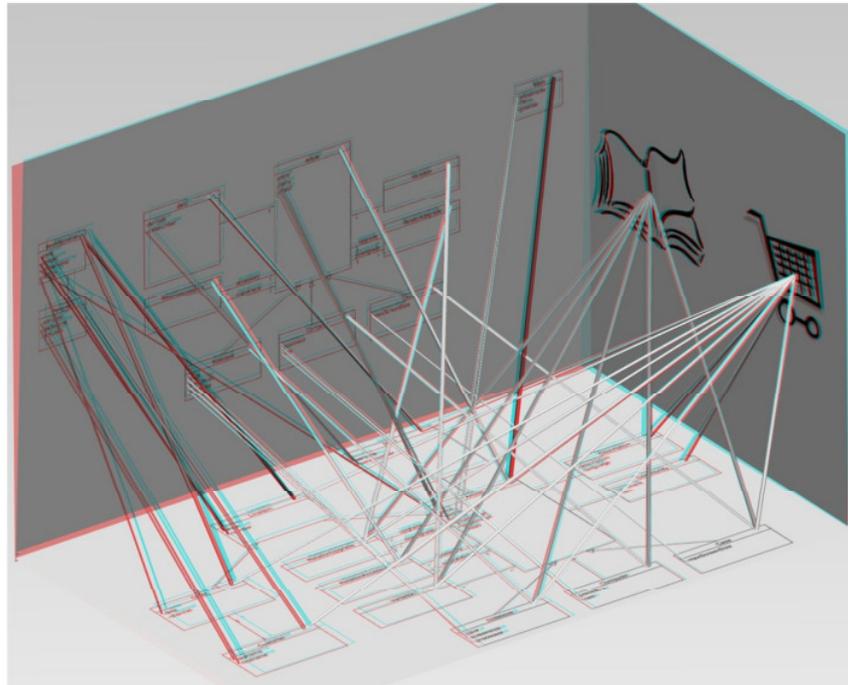


Abbildung 29 3D Stereogramm

## 11.2 EMF

„Eclipse Modeling Framework (EMF) is an Eclipse-based modeling framework and code generation facility for building tools and other applications based on a structured data model. From a model specification described in XMI, EMF provides tools and runtime support to produce a set of Java classes for the model, a set of adapter classes that enable viewing and command-based editing of the model, and a basic editor. Models can be specified using annotated Java, UML, XML documents, or modeling tools, then imported into EMF. Most important of all, EMF provides the foundation for interoperability with other EMF-based tools and applications.” [1]

### Ablehnender Entscheid

Wir hätten uns EMF sowohl als Basistechnologie für PSA vorstellen können, als auch das Datenmodell des MCM im Ecore Modell zu modellieren. Beide Ideen haben wir verworfen. EMF ist eine sehr weitreichende und komplexe Plattform. Wie man in den unteren Abschnitten sieht hätte man einiges an Hilfe bekommen um z.B. Editoren zu erzeugen. Die durch EMF mitgebrachte Komplexität und der nötige Einarbeitungsaufwand um korrekt damit umzugehen schienen uns allerdings höher als der mögliche Gewinn. Auch die Modellierung des Datenmodells des MCM im Ecore Modell wäre nur dann sinnvoll gewesen, wenn dies von foryouandyourcustomers gewünscht worden wäre, um von den weitreichenden EMF Möglichkeiten gebrauch zu machen.

### 11.3 GMF, GEF, GEF3D

- GMF** “The Graphical Modeling Framework (GMF) is a framework within the Eclipse platform. It provides a generative component and runtime infrastructure for developing graphical editors based on the Eclipse Modeling Framework (EMF) and Graphical Editing Framework (GEF). The project aims to provide these components, in addition to exemplary tools for select domain models which illustrate its capabilities.” [2]
- GEF** “Graphical Editing Framework (GEF) is a framework that was developed for the Eclipse platform. It is used to create graphical editors for various diagrams like electrical networks or tree diagrams. Such diagrams offer easy editing capabilities for data in specific domains and are well suited as a graphical representation of that data. GEF editors can be used inside an Eclipse RCP application, the application framework offered by Eclipse. Usually, GEF is used as a part of the Graphical Modeling Framework (GMF) which combines the Eclipse Modeling Framework (EMF) and GEF to create the code for both the data model and the diagram editor. Editors generated with GEF consist of the following components:
- The diagram editor including tool palette
  - Figures which graphically represent the underlying data model elements
  - EditParts which match figures and their respective model elements
  - Request objects for user input
  - EditPolicy objects which evaluate the requests and create appropriate command objects
  - Command objects that edit the model and provide undo-redo
- ” [3]
- GEF3D** “GEF3D is an Eclipse GEF extension bringing 3D to diagram editing. That is with GEF3D you can create 3D diagrams, 2D diagrams and combine 3D with 2D diagrams. GEF3D extends GEF by providing 3D enabled draw and controller classes. Instead of drawing 2D figures, you can now draw 3D figures. Existing GEF-based 2D editors can be embedded into 3D editors with minimal effort. This is achieved by projecting the output of the embedded editors onto 3D planes and bridging the gap between 2D and 3D content: 3D objects can transparently access 2D content and vice versa. Additionally, multi-diagram editors can be created by combining several single diagram editors. For example multiple 2D diagrams can be displayed and edited simultaneously on different planes and 3D elements can be used to display inter-model connections. Fully 3D graph based editors can be implemented with little or no knowledge of 3D programming since the framework is used exactly the same way as the original 2D framework.” [4]

## 12 Glossar

PCM	Product Content Management: System in welchem Produktstruktur abgelegt ist. (Die wichtigsten: Heiler, Hybris, Stibo, Riversand)
MCM	Multi Channel Management
PIM	Product Information Management
Channel	Ein Kanal bietet dem User z.B. dem Kunden auf spezifische Art und Weise Zugang zu einer Dienstleistung eines System.
Touchpoint	Ein Touchpoint ist eine logische Einheit. Er beschreibt den Zugang zu einem konkreten Service über einen konkreten Channel.
Umsystem	Primär Systeme zum Verwalten von Daten welche mit dem MCM verbunden werden.

## 13 Quellen

### 13.1 Bilder

Für unsere Darstellungen haben wir folgende Icons aus dem Internet verwendet:

- <http://www.root-service.de/Images/Design/Webshop.jpg>
- <http://www.intercms.nl/images/stories/webshop.jpg>
- <http://www.agsupport.co.nz/images/database.gif>
- <http://www.shakehands.com/media/product/12/sh-bilanz-48.png>
- <http://www.proffix.net/Portals/0/Content/Icons/lager.png>
- [http://www.mricons.com/store/png/121836\\_40666\\_64\\_ecommerce\\_empty\\_shoppingcart\\_webshop\\_icon.png](http://www.mricons.com/store/png/121836_40666_64_ecommerce_empty_shoppingcart_webshop_icon.png)
- <http://www.mpotechnology.ch/res/iphone.gif>
- [http://images04.clasies.com/images/advertisements/2010/11/30/pagina-web-dominio-host-logo-230-paginas-sencillas\\_05d08f6d\\_4.jpg](http://images04.clasies.com/images/advertisements/2010/11/30/pagina-web-dominio-host-logo-230-paginas-sencillas_05d08f6d_4.jpg)
- [http://images.maplin.co.uk/300/a47ay\\_trans.jpg](http://images.maplin.co.uk/300/a47ay_trans.jpg)
- [http://www.beycoo.com/image/image\\_gallery?uuid=02ad6c53-1112-48a7-a534-72535f6fe271&groupId=11436&t=1295888532057](http://www.beycoo.com/image/image_gallery?uuid=02ad6c53-1112-48a7-a534-72535f6fe271&groupId=11436&t=1295888532057)
- <http://www.aprov-it.com/img/oferta/crm.png>

Stand: 22.12.2011

### 13.2 Literaturverzeichnis

- [1] Wikipedia, Eclipse Modelling Framework  
[http://en.wikipedia.org/wiki/Eclipse\\_Modeling\\_Framework](http://en.wikipedia.org/wiki/Eclipse_Modeling_Framework)  
Letzter Zugriff am 23.11.2011
- [2] Wikipedia, Graphical Modeling Framework  
[http://en.wikipedia.org/wiki/Graphical\\_Modeling\\_Framework](http://en.wikipedia.org/wiki/Graphical_Modeling_Framework)  
Letzter Zugriff am 23.11.2011
- [3] Wikipedia, Graphical Editing Framework  
[http://en.wikipedia.org/wiki/Graphical\\_Editing\\_Framework](http://en.wikipedia.org/wiki/Graphical_Editing_Framework)  
Letzter Zugriff am 23.11.2011
- [4] Eclipse.org, GEF3D  
<http://www.eclipse.org/gef3d/>  
Letzter Zugriff am 23.11.2011

## 14 Abbildungsverzeichnis

Abbildung 1 Layer der Datenbasis.....	10
Abbildung 2 Einfache Modellierung des System Layer .....	12
Abbildung 3 UML Klassendiagramm Meta Model .....	14
Abbildung 4 Mögliche User des Systems .....	16
Abbildung 5 Beispiel von Metadaten.....	17
Abbildung 6 PIM Übersicht .....	19
Abbildung 7 Gegenüberstellung Attribute und Kanäle .....	19
Abbildung 8 Gegenüberstellung Attribute und Kanal .....	20
Abbildung 9 Auflistung Metaattribute .....	20
Abbildung 10 Statistik.....	21
Abbildung 11 Prototyp Klassendiagramm .....	33
Abbildung 12 Prototyp UML Klassendiagramm .....	33
Abbildung 13 Prototyp Übersicht .....	34
Abbildung 14 Prototyp UML.....	34
Abbildung 15 Prototyp Attribut Gegenüberstellung .....	35
Abbildung 16 Prototyp Gegenüberstellung Attribute und Kanal .....	35
Abbildung 17 Prototyp Klassifizierung .....	36
Abbildung 18 Prototyp CCC .....	36
Abbildung 19 Prototyp Metaattribute.....	37
Abbildung 20 Prototyp Interessenten.....	37
Abbildung 21 Prototyp Statistik.....	38
Abbildung 22 3d Übersicht .....	39
Abbildung 23 3d Umsysteme und Aktoren.....	39
Abbildung 24 3d Klassifizierung und Attribute.....	40
Abbildung 25 3d Klassifizierung und Datentypen.....	40
Abbildung 26 3d drei C.....	41
Abbildung 27 3d Umsysteme, PIM, Aktoren .....	41
Abbildung 28 3d CAD.....	42
Abbildung 29 3d Stereogramm.....	43

## 15 Anhang

### 15.1 A: Fragenkatalog der Sitzung vom 06.12.2011

#### 15.1.1 Situation Heute

- Was ist das Produkt einer Produktstrukturierung?
  - Ein Datenmodell für ein PIM?
  - Eine Multichannel Lösung?
  - Welche Probleme werden gelöst?
  - Welche Dokumente erstellt?
  - Welche Programme geschrieben?
  - Wer arbeitet nach eurer Beratung weiter mit euren Resultaten?
  - Wie / Was wird weiter gearbeitet?
- Wieso fyayc?
  - Was ist euer Know How?
  - Grosse Firmen haben selbst eine IT-Abteilung und Entwickler. Wieso braucht es fyayc?
  - Was leistet ihr?
  - Was zeichnet euch aus?
  - Was ist innovativ?
- Was macht ihr?
  - Was / Wie viel wird bei einem solchen Projekt selbst entwickelt?
  - Was wird programmiert?
  - Was wird modelliert?
  - Was wird von 3rd Party Tools geleistet?
  - Was wird konfiguriert?
- Schwierigkeiten
  - Welche Probleme / Fragen sind während der Produktstrukturierung zu lösen?
  - Welche Fehler können passieren?
    - Welche Fehler passieren häufiger?

- Was sind die Folgen dieser Fehler?
  - Was muss beachtet werden, damit diese Fehler nicht passieren?
- Klassifizierung
  - Was bringt das?
  - Welches Problem wird damit gelöst?
  - Was ist daran schwierig?
  - Welche (technischen) Folgen hat dies für den Rest des Projekts?
  - Wie sieht das Resultat aus?
- Kategorisierung
  - Was bringt das?
  - Welches Problem wird damit gelöst?
  - Was ist daran schwierig?
  - Welche (technischen) Folgen hat dies für den Rest des Projekts?
  - Wie sieht das Resultat aus?
- Daten
  - Werden nur alte Daten anders verfügbar gemacht?
  - Oder werden viele neue Daten erfasst?
- Wer ist alles an Prozess beteiligt?

## 15.1.2 Datenbasis

### 15.1.2.1 Konzepte / Entitäten

- Welche Entitäten werden modelliert?
- Wie hängen diese Entitäten zusammen?
- Was ist die innere Struktur der Entitäten?
- Welche Daten werden zu einer Entität erfasst?
- Welche Metadaten interessieren?
- Wie werden diese Entitäten modelliert?
- Wie viele Instanzen eines Konzepts gibt es?

#### 15.1.2.1.1 PIM

- Innere Struktur des PIM
  - Sind da 1 oder mehrere Systeme?
  - Falls mehrere Systeme:
    - Wie spielen diese zusammen (technisch und logisch)
    - Wie präsentieren sich diese nach aussen?
    - Wie werden sie modelliert?
  - Welche Anforderungen stellt dies an PSA?
- Genügt ein UML-Diagramm als Modellierung?
- Welche Meta Attribute braucht es?
  - Für die Klassen
  - Für die Attribute

#### 15.1.2.1.2 Umsystem

- Ist UML eine sinnvolle Art, die Umsysteme zu modellieren?

#### 15.1.2.1.3 Touchpoint

- Wie sieht der Datenlayer der Touchpoints aus? Macht es Sinn, die Touchpoints in UML zu modellieren?

#### 15.1.2.1.4 Datenströme

- Müssen allfällige Berechnungen der Attribute weiter modelliert werden, oder genügt ein Kommentarfeld?

#### 15.1.2.1.5 User (Mitarbeiter und Kundensegmente)

- Wer arbeitet mit welchen Systemen und ist an welchen Attributen interessiert?

- Wie spielen die Systeme für diesen User zusammen (Z.B. Im Katalog hat es einen Code um das Produkt im Webshop zu finden)

#### 15.1.2.1.6 Weitere Möglichkeiten

- Datentypen
- Infrastruktur (Server, Netzwerke)

#### 15.1.2.2 Zustände

- Muss nur der geplante Zustand modelliert werden?
- Oder muss auch der Alte Zustand angezeigt werden können?

### 15.1.3 Einsatz des Tools

#### 15.1.3.1 Prozess

- Wie soll sich PSA in die Prozesse / Abläufe von fyayc integrieren?
- Bei welchen Phasen soll es eingesetzt werden?
- Welche Probleme soll es lösen?
- Welche Use Cases gibt es?

#### 15.1.3.2 User

- Wer benutzt das System?
  - Welche fyayc Mitarbeiter?
  - Welche Mitarbeiter der Kunden?
- Welches Know How können wir erwarten?
  - UML
  - kennt Begriffe aus der Produktstrukturierung
- Wie gross ist das Technikverständnis und die Abstraktionsfähigkeit der User?
- Soll PSA von mehreren Usern gleichzeitig eingesetzt werden?
  - Nur für Anzeige
  - Auch für Modellierung

- Welche Sprachen sollen unterstützt werden?
  - Genügt eine oder müssen es mehrere sein?
  - Sonstige Anforderungen an Internationalisierung? (Rechts-Links Schriften etc.)

### 15.1.3.3 Daten Anzeige

- Welche Einsatzmöglichkeiten gibt es?
  - Verschiedene Darstellungen und Diagramme helfen das System zu verstehen.
  - Dem fyayc-Entwickler ermöglichen, die gesamte bisherige Modellierung zu analysieren.
  - Für die Kunden-Entwickler werden alle nötigen Details dokumentiert, so dass diese das geplante System umsetzen können.
  - Mit den Angestellten des Kunden kann die geplante Lösung besprochen werden: Man kann dem Angestellten zeigen, was das neue System aus seiner Sicht bedeutet. Somit kann man feststellen, ob alle seine Bedürfnisse abgedeckt werden.
  - Man kann die neue Lösung den Angestellten des Kunden präsentieren, so dass die Akzeptanz für das Projekt steigt.
  - Man kann dem Manager des Kunden aufzeigen, dass man seine Firma verstanden hat (durch eine Beschreibung der neuen Systeme).
  - Man kann dem Manager des Kunden aufzeigen, wie die neuen Touchpoints und das PIM zusammen arbeiten und sollte ihn so überzeugen können, dass diese Lösung für die Zukunft vorbereitet ist.
  - Der Manager des Kunden sieht die zentrale Rolle des PIM und ist bereit, die nötigen Mittel zu sprechen um dieses sauber zu designen.
- Was muss Angezeigt werden / Welche Aussage muss eine Anzeige machen für einen Use Case?
  - Z.B. Welche Information / Aussage braucht der Katalogmacher?

- Genügt es, wenn der gewünschte Zielzustand modelliert wird, oder muss auch der Ist-Zustand modelliert werden? Und wie man diesen erreicht?
  - Müssen Änderungen an bestehenden Systemen dokumentiert werden?

#### 15.1.3.4 Daten Eingabe

- Sollen Daten von externen Tools (insbesondere UML Tools) importiert werden. In PSA baut man dann darauf auf (Problem: Synchronisierung der Daten, wenn extern etwas geändert wird). Oder muss PSA für alle nötigen Daten einen Editor bereitstellen.
- Welche Hilfestellungen können geboten werden?
  - Welches Know How kann transferiert werden?
    - UML-Templates
    - Liste von Standardwerten
    - Zu befüllende Metaattribute helfen dem Entwickler, die richtigen Fragen zu stellen.
    - Standard Datensätzen (z.B. verschiedene Farbtabelle) reduzieren repetitive Aufgaben.
- Sind immer in etwa dieselben Daten von Interesse, oder sollte es eine Möglichkeit geben die zu erfassenden Meta Daten projektspezifisch zu wählen.

#### 15.1.4 Technologie

##### 15.1.4.1 PIM-Systeme

- Welchen Einfluss haben Systeme wie Hybris, Heiler, etc. auf die Modellierung
- Wie kann dies PSA beeinflussen?
- Welche dieser Tools sollen in diesem Sinne unterstützt werden?

##### 15.1.4.2 Entwickler Tools

- Welche Tools werden (heute) eingesetzt für die Entwicklung (z.B. UML-Tools wie Enterprise Architect)?
- Wie dokumentiert ihr (heute) eure Resultate?

- Welche Daten müssen importiert und / oder exportiert werden?
- Welche Datenformate müssen importiert und / oder exportiert werden?
- Braucht es Schnittstellen zu Technologien der Umsysteme und Touchpoints?

#### 15.1.4.3 Mögliche Technologien für die Umsetzung von PSA

- Auf welche Technologien muss PSA aufbauen
  - Welche Programmiersprachen dürfen verwendet werden?
- Welche Entwicklerwerkzeuge sollen eingesetzt werden
  - Welche Entwicklungsumgebung
  - Welches Tool für die Modellierung (z.B. Enterprise Architect)?

#### 15.1.4.4 Laufzeitumgebung

- Welche Anforderungen an die Laufzeitumgebung werden dürfen gestellt werden?
  - Wie gross sind die Bildschirme
  - Ist ein Netzwerk vorhanden?
    - Welche Datenrate?
    - Wie viel Traffic darf verursacht werden?
  - Ist eine Datenbank vorhanden?

#### 15.1.5 Diverses / Abschluss

- Welches Ziel wird mit PSA verfolgt? / Welches Problem soll mit PSA gelöst werden?
- Was ist der Vorteil / Gewinn beim Einsetzen von PSA?
- Wie hilft PSA dem Entwickler?
- Wo liegen die Schwerpunkte?
- Was muss besonders gut umgesetzt sein, welches sind Randthemen?
- Was muss zuerst umgesetzt sein, damit PSA eingesetzt werden kann?
- Wie gross darf Einarbeitungsaufwand sein?
- Wie viel Zeit darf für die Eingabe der Daten aufgewendet werden?
- Auch für Rechercharbeiten

## 15.2 B: Aufgabenstellung

### Visualisierung von komplexen Produktstrukturen

Studiengang	Informatik (I)
Semester	HS 2011/2012 (19.09.2011-19.02.2012)
Durchführung	Studienarbeit
Fachrichtung	Software
Institut	IFS: Institut für Software
Gruppe	Christoph Rosenberger Michael Steiner
Betreuer	Daniel Keller
Industriepartner	foryouandyourcustomers.com
Aufgabenstellung	<p>Im Produkt-Informationssystem (PIM) einer grossen Firma (Bsp. Migros, Hornbach) stecken in erster Linie Daten über die Produkte und ihre Eigenschaften (Preis, Farbe, Gewicht, Lieferant...). Dazu kommen aber noch Klassifizierungen ('Socken', 'Tierfutter', mit Vererbung) und Kategorisierungen ('extra leicht', 'pastellfarbig', 'Sommerkollektion 2011', 'ist Zubehör zu X', 'ist upgrade zu Y', 'ist Ersatz für Z' ...). Das macht die Sache schon recht komplex, vor allem, wenn Zugriffs-Effizienz eine Rolle spielt. Wir möchten ein Tool bauen (UML? auf iPad? 3D Datenwürfel?) welches die Datenstrukturen visualisiert, so dass ein Informatik-Unkundiger Vor- und Nachteile nachvollziehen kann, wenn verschiedene Varianten der (Neu-)Strukturierung diskutiert werden.</p>

Die Studenten sollen dazu verschiedene Aspekte der Problemstellung erläutern und damit die Grundlagen für eine Folgearbeit erarbeiten.

Daniel Keller

## 15.3 C: Erklärung über die eigenständige Arbeit

### 15.3.1 Christoph Rosenberger



#### Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer vereinbart wurde,
- dass ich sämtliche verwendeten Quellen angegeben habe.

Ort, Datum:

Pfäffikon, 23.12.2011

Name, Unterschrift:

*Christoph Rosenberger*

### 15.3.2 Michael Steiner



#### Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer vereinbart wurde,
- dass ich sämtliche verwendeten Quellen angegeben habe.

Ort, Datum:

Pfaffikon, 23.12.2011

Name, Unterschrift:

Michael Steiner, 

## 15.4 D: Persönliche Berichte

### 15.4.1 Christoph Rosenberger

"Visualisierung von komplexen Produktstrukturen" war und ist ein Abtaucher in eine neue Perspektive auf die Welt der Konsumwelt. Was läuft hinter den Kulissen einer Firma mit hunderten und tausenden Produkten, wie zum Beispiel der Migros, damit auf verschiedenen Verkaufskanälen, im Laden, im Internet, in Katalogen die entsprechenden Verkaufsinformationen konsistent vorhanden sind? Und auch über zehn Jahre nach platzen der .com Blase ist die elektronische Revolution des Produktmanagements keines falls abgeschlossen: Vielleicht wurde die experimentelle Kinderstube gerade verlassen. Aber vom einfachen Einsatz einer Standardlösung sind wir noch Jahre entfernt.

Unser Industriepartner foryouandyourcustomers, eine sehr junge Firma, ist auf dem Weg zum Marktleader im Bereich Multichannel Management und Produktstrukturierung. Eines ihrer Probleme ist allerdings eine ausführliche und einheitliche Dokumentation ihrer Erkenntnisse während eines Produktstrukturierungs Projekts.

Genau da setzt auch unsere Arbeit an: Wir sollten etwas machen, das ihnen nützt. Als vage Richtung war die Idee einer 3D Visualisierung gegeben. Nebst dieser "etwas schwammigen" Auftragsstellung wurden wir am Anfang des Projekts an zwei Halbtagen in die Themenwelt der Produktstrukturierung eingeführt.

Somit fingen wir mit unserer Arbeit an: Ein genaues Ziel hatten wir nicht, entsprechend auch keine Agenda. Wir liessen die Arbeit - und so auch die Probleme - einfach auf uns zu kommen. Wir überlegten uns, wie 3D Darstellungen aussehen könnten und wir suchten nach Technologien um dies auch als Software umzusetzen.

In Woche 5 hatten wir ein weiteres Meeting bei foryouandyourcustomers und präsentierten unsere 3D Visualisierungen. Das Feedback war enthusiastisch und positiv, inhaltlich wurde uns aber klar, dass wir durch die dritte Dimension kaum einen Vorteil hatten.

Nach einer kurzen Abwägung gegenüber den grossen Technischen Problemen die aus diesem Ansatz entstanden wären, haben wir uns im weiteren Verlauf auf zwei Dimensionen beschränkt. Dafür wurde uns ein anderes Problem immer deutlicher: Wir wussten gar nicht so recht, was wir da visualisieren sollten. Leider konnten wir keine Dokumentation eines realen Projektes bekommen. Unser Fehler war dann, dass wir das so akzeptierten und nicht nachhaken. Wir hatten auch Hemmungen einfach weitere Termine mit foryouandyourcustomers zu vereinbaren und ihre Zeit in Anspruch zu nehmen. So versuchten wir unsere eigenen Baby Beispiele zu basteln.

In der Mitte des Semesters einigten wir uns dann mit Herrn Keller darauf, dass diese Arbeit eine Vorarbeit für eine Bachelorarbeit sein sollte. Um unsere Ideen konkreter präsentieren zu können, und da es sich auch hierbei um eine Software Engineering Arbeit handeln sollte, begannen wir einen Prototypen zu schreiben. Wir gingen das Risiko ein, auf einer uns nur oberflächlich bekannten Technologie etwas umsetzen zu wollen: EMF. Als wir aber nach zwei Wochen mehr Fragen als Antworten hatten, entschieden wir uns um. Um möglichst schnell vorwärts zu kommen nahmen wir einfache Technologien: PHP und MySQL.

Mit dieser guten Entscheidung kamen wir bei der Umsetzung schnell voran und wir konnten unsere Ideen schnell umsetzen. Dafür wurde ein anderes Problem wieder deutlich. Wir mussten nun ein konkretes Datenmodell für unseren Prototypen haben. Durch pragmatische

Entscheidungen haben wir dieses auch erstellt. Aber wir verstanden immer besser, dass wir die eigentlichen Probleme der Produktstrukturierung nur oberflächlich verstanden!

Um nun eine möglichst gute Vorbereitung auf die Bachelorarbeit zu liefern, vereinbarten wir ein letztes Treffen bei foryouandyourcustomers. Diesmal mit dem expliziten Ziel, unsere Fragen beantwortet zu bekommen. Mit einer fünf seitigen Frageliste konnten wir dann doch noch einiges Licht ins Dunkle bringen. Und auch bei diesem Treffen lernten wir wieder vollkommen neue Aspekte und Probleme der Produktstrukturierung kennen.

Die restlichen Wochen nutzten wir nun, um all unsere Erkenntnisse zu dokumentieren. Leider haben wir immer noch nicht ein so tiefgehendes Verständnis um weitgehend eigenständig ein Tool zu entwickeln. Wir sehen aber auch, dass wir vieles gelernt und entwickelt haben, dass wir wirklich etwas geleistet haben. Und so denken wir, dass wir eine solide Basis erarbeitet haben für eine Folgearbeit: Wir sind optimistisch dass, unter Anwendung der uns beigebrachten agilen Software Engineering Methoden, wir ein nützliches Tool für foryouandyourcustomers, deren Kunden und, natürlich, deren Kunden, entwickeln werden können.

### 15.4.2 Michael Steiner

Da wir beide keine Erfahrungen im Bereich der Produktstrukturierung hatten, mussten wir zuerst eingearbeitet werden. Es stellte sich heraus, dass das Thema relativ komplex ist und wir mehr Zeit in das Verständnis dieses Business investieren mussten, als wir eigentlich geplant hatten. Nachdem wir das Gefühl hatten, dass wir jetzt genug verstanden haben, begannen wir Darstellungen zu entwerfen, welche die Probleme bei solchen Projekten aufzeigen und den Entwickler unterstützen. Dadurch kamen noch mehr Fragen auf und so verstrich die Zeit. Unsere Aufgabe bestand darin, 3D Visualisierungen zu entwerfen. Wir investierten viel Zeit in diese und mussten dann (leider relativ spät) feststellen, dass wir auf dem falschen Weg sind. So starteten wir einen neuen Versuch im 2D Bereich. In der Mitte des Projektes begannen wir dann etwas konkreter zu werden. Wir starteten einige Versuche mit EMF und entschieden uns dann aus zeitlichen Gründen einen Prototyp mit PHP zu entwerfen. Nun war es uns auch möglich konkrete Fragen auszuformulieren. Wir erstellten einen Fragekatalog welcher wir dann mit foryouandyourcustomers ausfüllten. Ich denke, dass dies ein sehr guter und wichtiger Schritt für unsere Arbeit war. Nichtsdestotrotz bin ich der Meinung, dass wir diesen Schritt etwas früher hätten machen sollen. Uns gelang es viel Wissen von foryouandyourcustomers niederzuschreiben, so dass wir in einer Folgearbeit direkt mit der Entwicklung beginnen können.

Ich konnte viel über den Ablauf eines Projektes und vor allem dessen Hürden lernen. Durch die Reflektionen im Team und mit dem Dozenten bin ich sicher, dass ich in meinen späteren Projekten wieder ein grosses Stück effizienter arbeiten kann. Auch was den Umgang mit dem Kunden betrifft, konnte ich einiges lernen. Aber nicht nur im Bereich Projektmanagement konnte ich viel profitieren. Ich weiss jetzt schon einiges über Produktstrukturen und dieses Wissen lässt sich auch auf andere Probleme übertragen.