

Studienarbeit

Vortragender: Alexander Haag



**Konzeption und Umsetzung einer
Erweiterung des Common
Warehouse
Metamodel (CWM) zur
Beschreibung von imperfekten
Daten**

Übersicht

- Analyse
 - CWM
 - Imperfekte Daten
- Konzeption
 - 1. Ansatz
 - 2. Ansatz
- Umsetzung
- Evaluation
- Ergebnis und Ausblick

Analyse: CWM

- 5 Schichten
- 21 Packages
- Verwendung nur der Packages, die gebraucht werden
- Bsp: Relational, Core & Relationships zur Darstellung von Relationalen Schemata

Management	Warehouse Process			Warehouse Operation		
Analysis	Transformation	OLAP	Data Mining	Information Visualization	Business Nomenclature	
Resource	Object	Relational	Record	Multidimensional	XML	
Foundation	Business Information	Data Types	Expressions	Keys and Indexes	Software Deployment	Type Mapping
Object Model	Core	Behavioral		Relationships	Instance	

Analyse: CWM & Data Warehousing

- Warum CWM?
- Mit dem CWM kann der ganze Data-Warehouse-Prozess modelliert werden
- Ein Data-Warehouse bietet die Möglichkeit große Datenmengen zu analysieren
- Die Daten aus verschiedenen Quellen werden integriert, bereinigt und aggregiert OLAP oder anderen Analysewerkzeuge bereitgestellt

Analyse: imperfekte Daten

- 3 unterschiedliche Typen imperfekter Daten:
 - Fuzzy
 - Unscharfes Wissen („Erik ist groß“, „stockender Verkehr“)
 - Imprecise
 - Ungenaues Wissen („zwischen Bruchsal und Heidelberg“)
 - Uncertain
 - Unsicheres Wissen („in Darmstadt regnet es gerade!“)
Aussage ohne zusätzliches Wissen nicht sicher entscheidbar ob wahr oder falsch!

Übersicht

- Analyse
 - CWM
 - Imperfekte Daten
- Konzeption
 - 1. Ansatz
 - 2. Ansatz
- Umsetzung
- Evaluation
- Ergebnis und Ausblick

Konzeption: 1. Ansatz

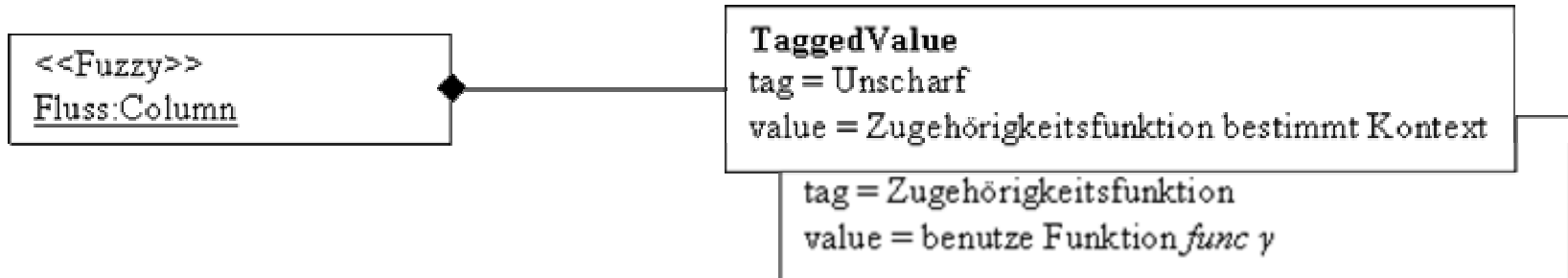
- Erweiterung des CWM mit *Stereotypes* und *TaggedValues*
- Anlehnend an das Kontextmodell von Schindler (auch verwendet in Studienarbeit von Andreas Merkel)
- Methode aus UML bekannt und einfach realisierbar, da vom CWM als Erweiterung bereitgestellt

Konzeption: 1. Ansatz

ID	Startkilometer	Endkilometer	Fluss	Störungstyp
1	A5-112	A5-122	Stau	Unfall
2	A81-40	A81-44	lebhaft	Baustelle
3	A8-288	A8-301	Stau	Baustelle
4	A61-82	A61-85	stockend	Baustelle
5	A7-71	A7-75	Stau	Unfall

1. Ansatz nicht ausreichend, da mit Funktionen auch gerechnet werden soll !!

	<<Imprecise>>	<<Imprecise>>	<<Fuzzy>>	<<Uncertain>>
ID	Startkilometer	Endkilometer	Fluss	Störungstyp
1	A5-112	A5-122	Stau	Unfall
2	A81-40	A81-44	lebhaft	Baustelle
3	A8-288	A8-301	Stau	Baustelle
4	A61-82	A61-85	stockend	Baustelle
5	A7-71	A7-75	Stau	Unfall

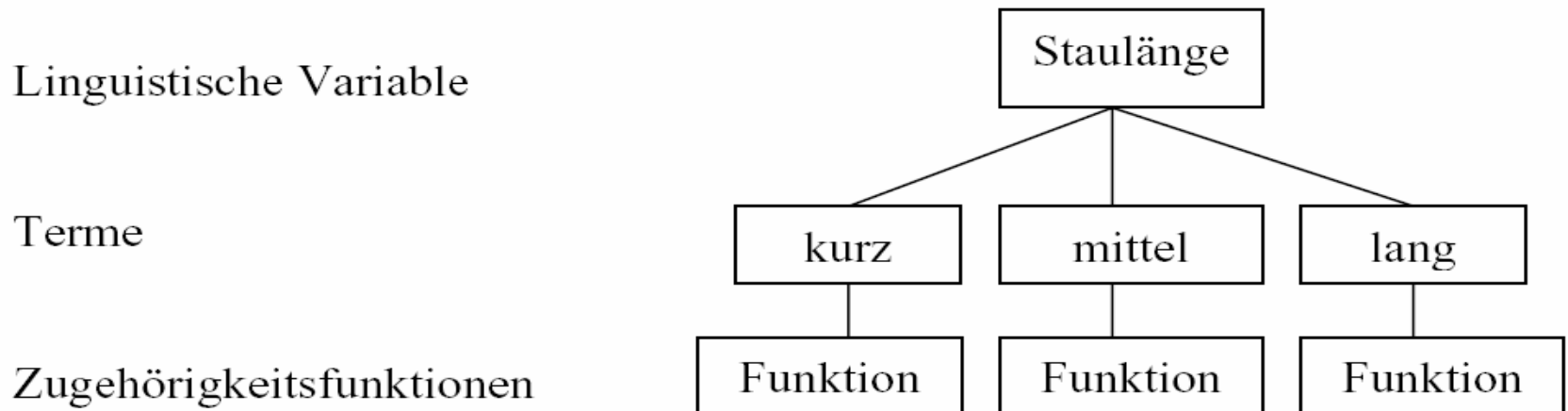


Konzeption: 2. Ansatz

- Erweiterung durch Vererbung
- Ergänzen eines eigenen Packages, wie bei der Erweiterung um ER-Modelle
- Im Package enthalten sind:
 - Funktionen (Wahrscheinlichkeit und Zugehörigkeit)
 - Imperfekte Erweiterung der Klasse ‚Column‘ aus dem Relational Package (bsp. Fuzzy-Column)

Konzeption: 2. Ansatz

- Eine Fuzzy-Column besitzt ein oder mehrere Terme, wie eine Linguistische Variable
- Jeder Term hat eine Zugehörigkeitsfunktion zu dieser linguistischen Variablen

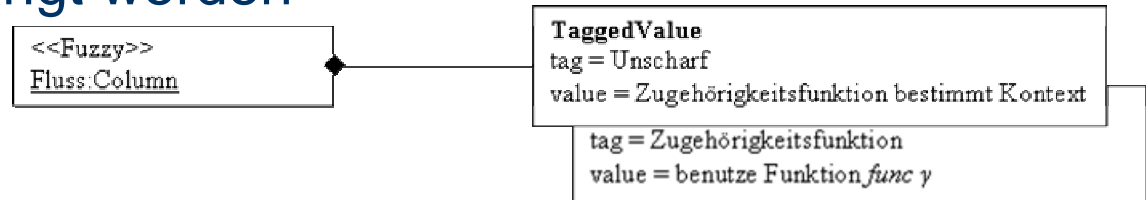


Übersicht

- Analyse
 - CWM
 - Imperfekte Daten
- Konzeption
 - 1. Ansatz
 - 2. Ansatz
- **Umsetzung**
- Evaluation
- Ergebnis und Ausblick

Umsetzung: 1. Ansatz

- Fuzzy-, Imprecise-, Uncertain- und ImperfectTable-Stereotypes wurden in Java implementiert
- Jeder *Stereotype* hat seine eigene *TaggedValues* oder *Constraints*, die bei der Instanziierung der *Stereotypes* mit erzeugt werden
- Anbindung an das CWM im Core-Package und Speicherung im Package **de.uka.ipd.cwm.imperfect**
- Funktionen können damit in Textform beschrieben und an Columns angehängt werden

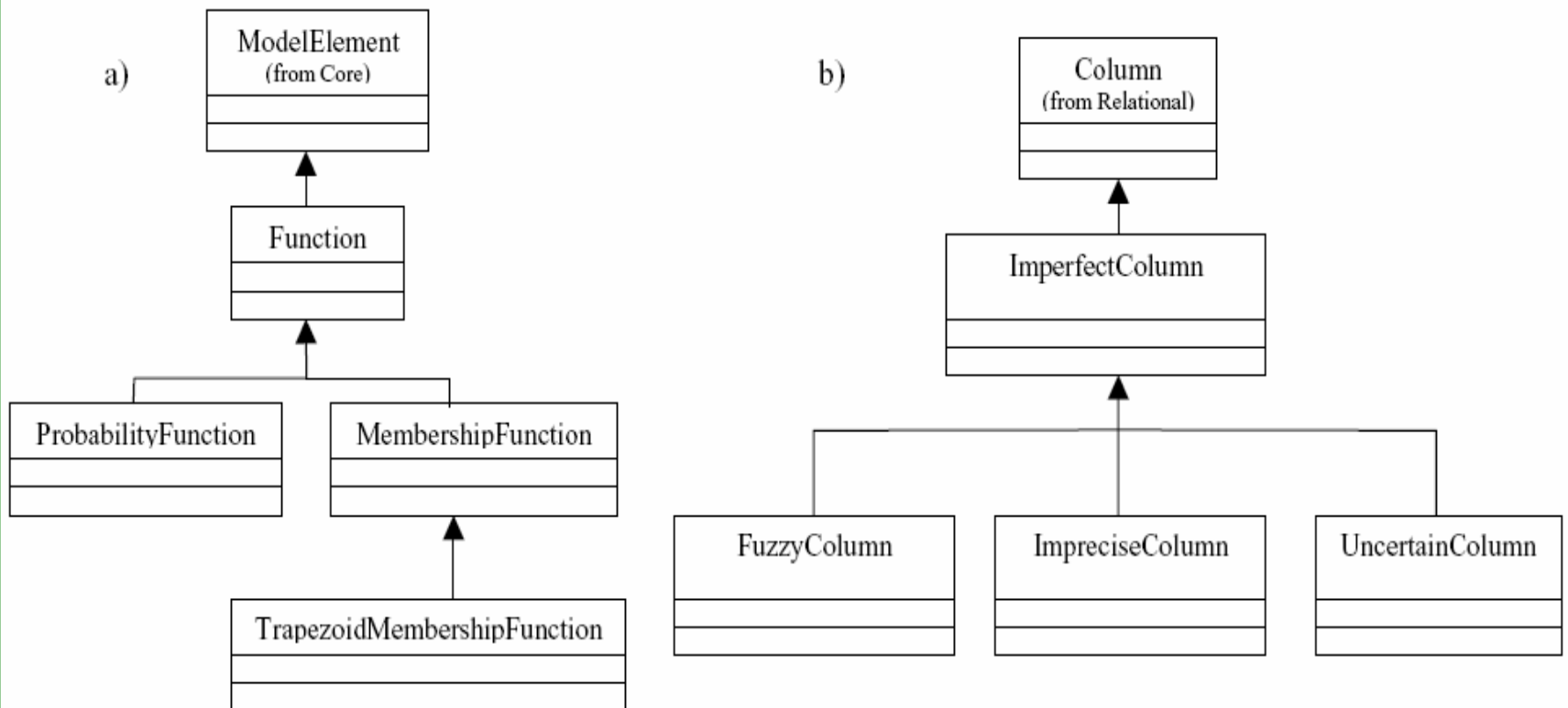


Umsetzung: 2. Ansatz

- Klassen für Funktionen inklusive Beispielfunktion (TrapezoidMembershipFunction) und Klassen für imperfekte Columns wurden implementiert
 - Zusammenfassen aller neuen Klassen im Package **de.uka.ipd.cwm.imperfect**
-
- Erweiterung der vorhandene XMI-DTD zur Darstellung der Erweiterung

Umsetzung: 2. Ansatz

Vererbungsstruktur der neuen Klassen



Übersicht

- Analyse
 - CWM
 - Imperfekte Daten
- Konzeption
 - 1. Ansatz
 - 2. Ansatz
- Umsetzung
- Evaluation
- Ergebnis und Ausblick

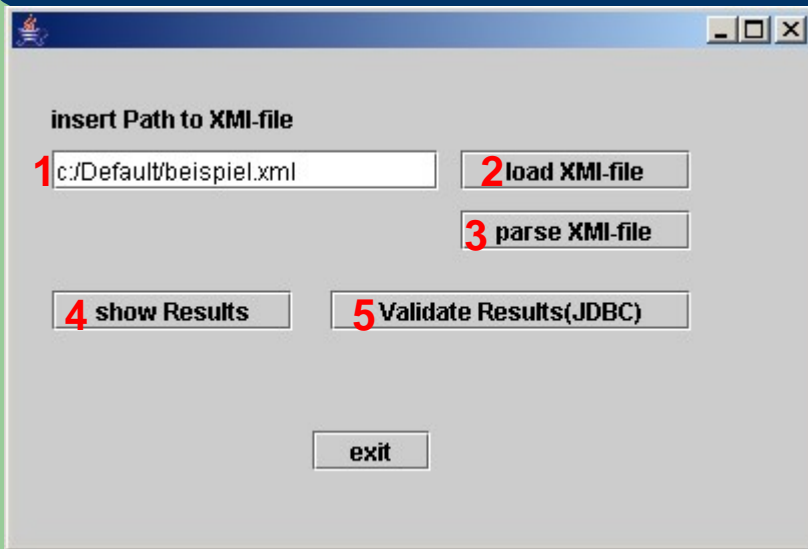
Evaluierungswerkzeug

- Werkzeug übernimmt XML-Datei im XMI-Format (XML Metadata Interchange)
- Übergibt die Datei dem XERCES-XML-Parser
- Liest das relationale Schema aus dem DOM-Tree des Parsers
- Aufbau eines SQL-Skripts zum Erzeugen eines SQL-Schemas in einer Datenbank
- Beispiel:

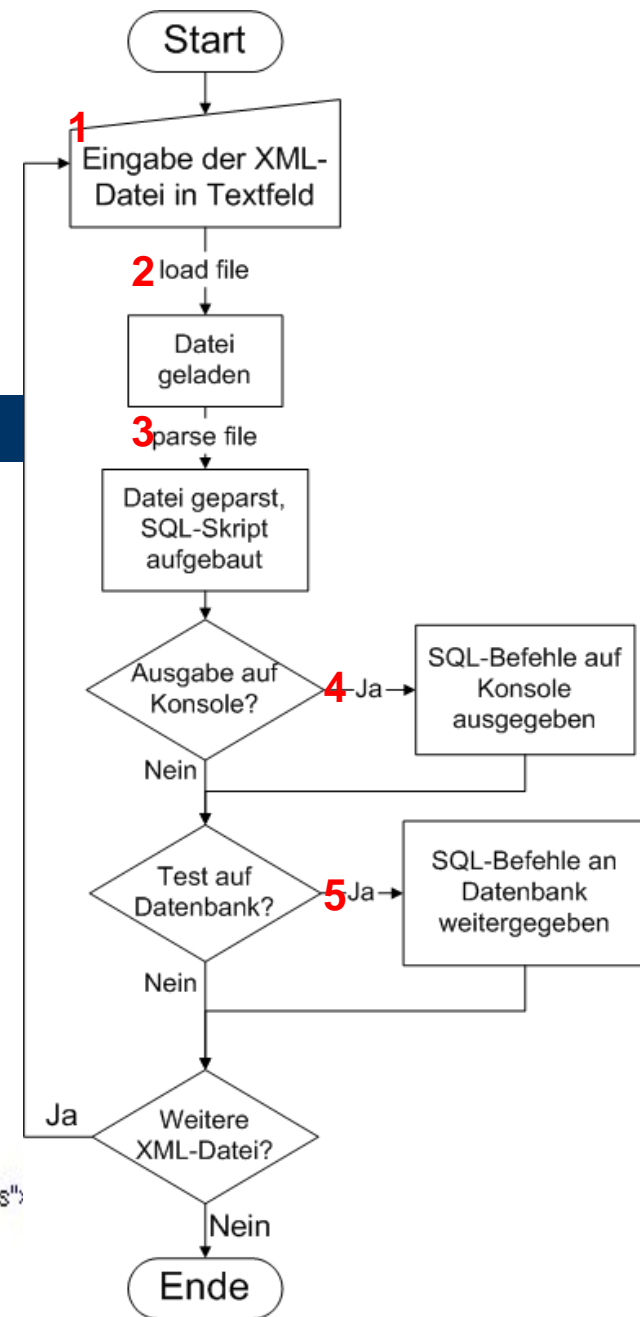
MeldungsID	Eingangszeit	Störungstyp	Straße	Geschwindigkeit	Verkehrsfluss
1	13:11,24/07/2004	Unfall	A5	30	stockend
2
3
...
...

Fuzzy

Evaluierungswerkzeug



```
<CWM:Classifier.feature>  
  <CWMRDB:Column xmi.id="spalte_5" name="Geschwindigkeit" owner="tabelle_1" type="NUMBER"/>  
</CWM:Classifier.feature>  
<CWM:Classifier.feature>  
  <CWMEXT:FuzzyColumn xmi.id="fuzzyspalte_1" associatedColumn="spalte_3" owner="tabelle_1" name="Verkehrsfluss":  
    <CWM:Classifier.feature>  
      <CWMEXT:TrapezoidMembershipFunction xmi.id="function_1" ll="80" ul="90" ur="350" lr="350" term="frei"/>  
    </CWM:Classifier.feature>  
  </CWMEXT:FuzzyColumn>  
</CWM:Classifier.feature>
```



Übersicht

- Analyse
 - CWM
 - Imperfekte Daten
- Konzeption
 - 1. Ansatz
 - 2. Ansatz
- Umsetzung
- Evaluation
- Ergebnis und Ausblick

Ergebnis

- Erste Erweiterung zu wenig, da mit Funktionen auch gerechnet werden soll
- Zweite Erweiterung ausreichend um Funktionen zu beschreiben und damit zu arbeiten
- Werkzeug versteht die Erweiterung durch Ergänzungen in der XMI-DTD und kann diese umsetzen

Ausblick

- Ergänzen der anderen Imperfektionsarten (Imprecise & Uncertain)
- Implementierung weiterer Zugehörigkeits- oder Wahrscheinlichkeitsfunktionstypen
- Werkzeug soll auch XML-Datei umändern können und nicht nur das Schema auslesen
- Erweiterung soll irgendwann auch mit kommerziellen Data-Warehouse-Werkzeugen bearbeitbar sein

Vielen Dank für Ihre Aufmerksamkeit



??? Noch Fragen ???

oder

!!! Anregungen !!!