

Anpassung eines Metamodells zur Beschreibung von imperfekten Daten in einem Data-Warehouse



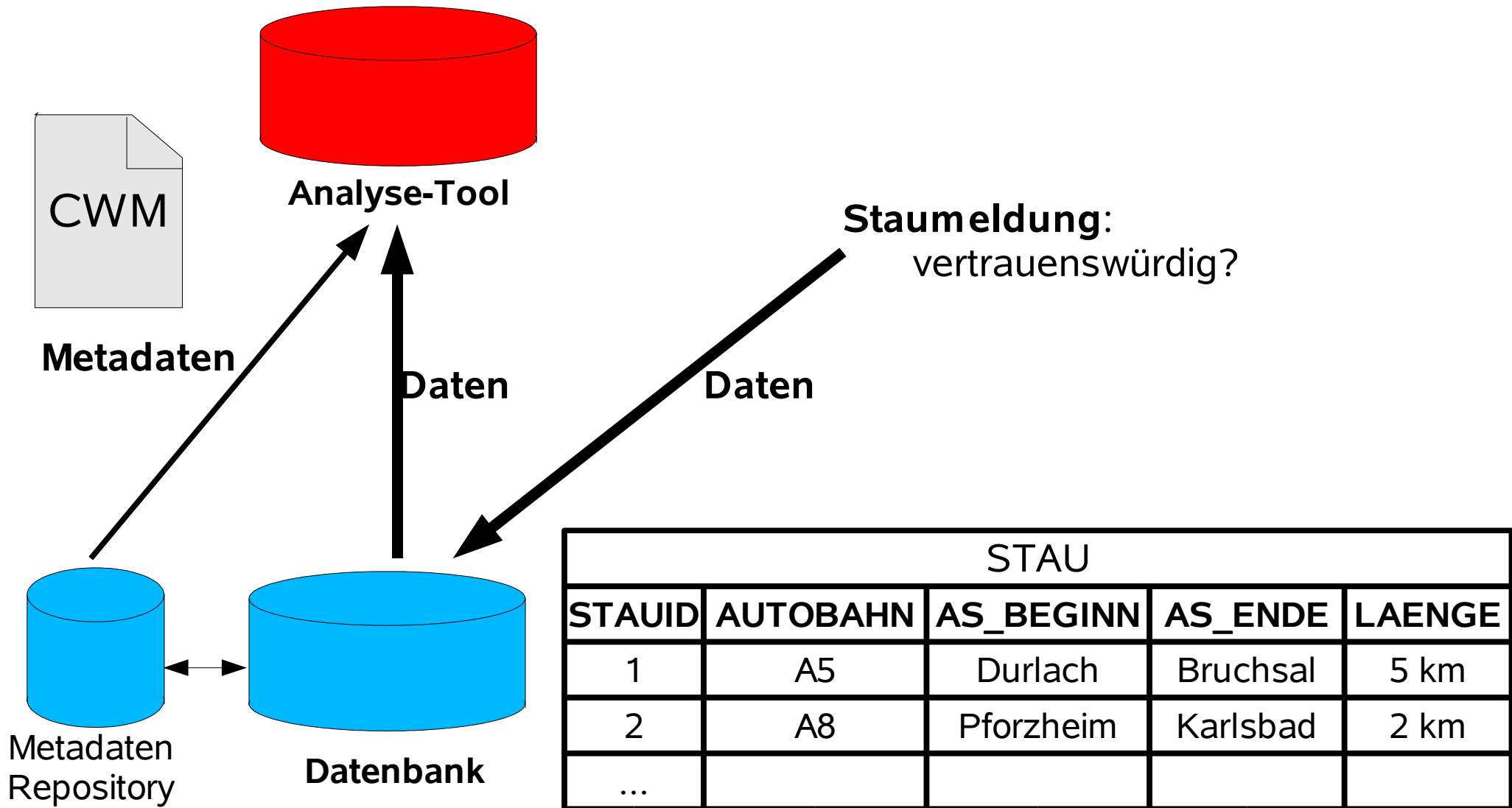
Studienarbeit

Nils Hilt

April 2005



Motivation





Gliederung

1. Motivation
2. Einführung
 - a.) Metadatenarchitektur
 - b.) Common Warehouse Metamodel
3. CWM Erweiterungstechniken
4. Imperfektion in Datenbanken
5. Erweiterung zur Beschreibung imperfekter Informationen
6. Ausblick



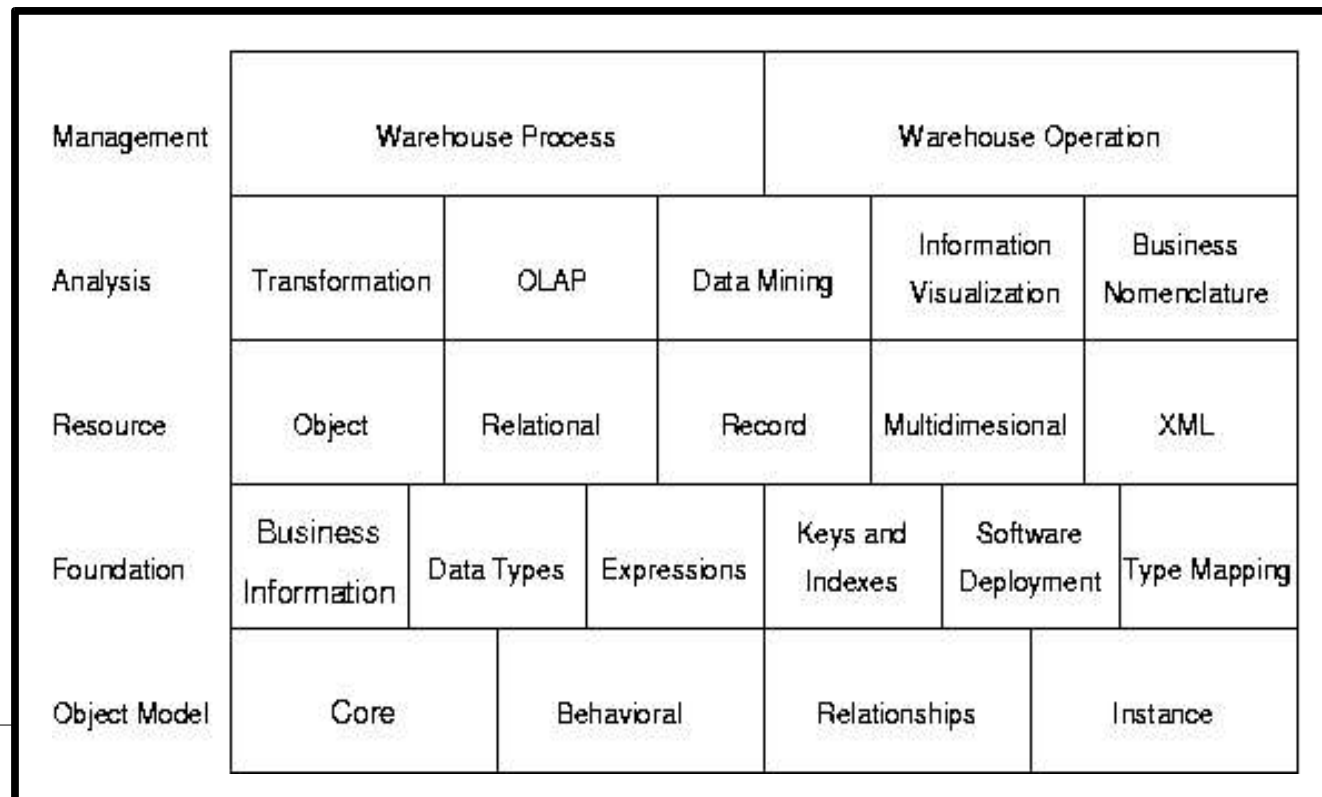
OMG Metadaten Architektur

Metaebene	MOF Bezeichnungen	Beispiele
M3	Meta-Metamodell	MOF Modell
M2	Metamodell Meta-meta Daten	UML Metamodell CWM Metamodell
M1	Modell Metadaten	UML Modell CWM Modell
M0	Objekt Daten	Modellierte Systeme Warehouse-Daten



Common Warehouse Metamodel

- Unterschiedliche Pakete zur Modellierung des gesamten Data-Warehouse-Prozesses
 - z.B.: Beschreibung relationaler Datenbasisschemata
 - Schichtenarchitektur
- Darstellung durch UML Klassendiagramme
- Erweiterbar





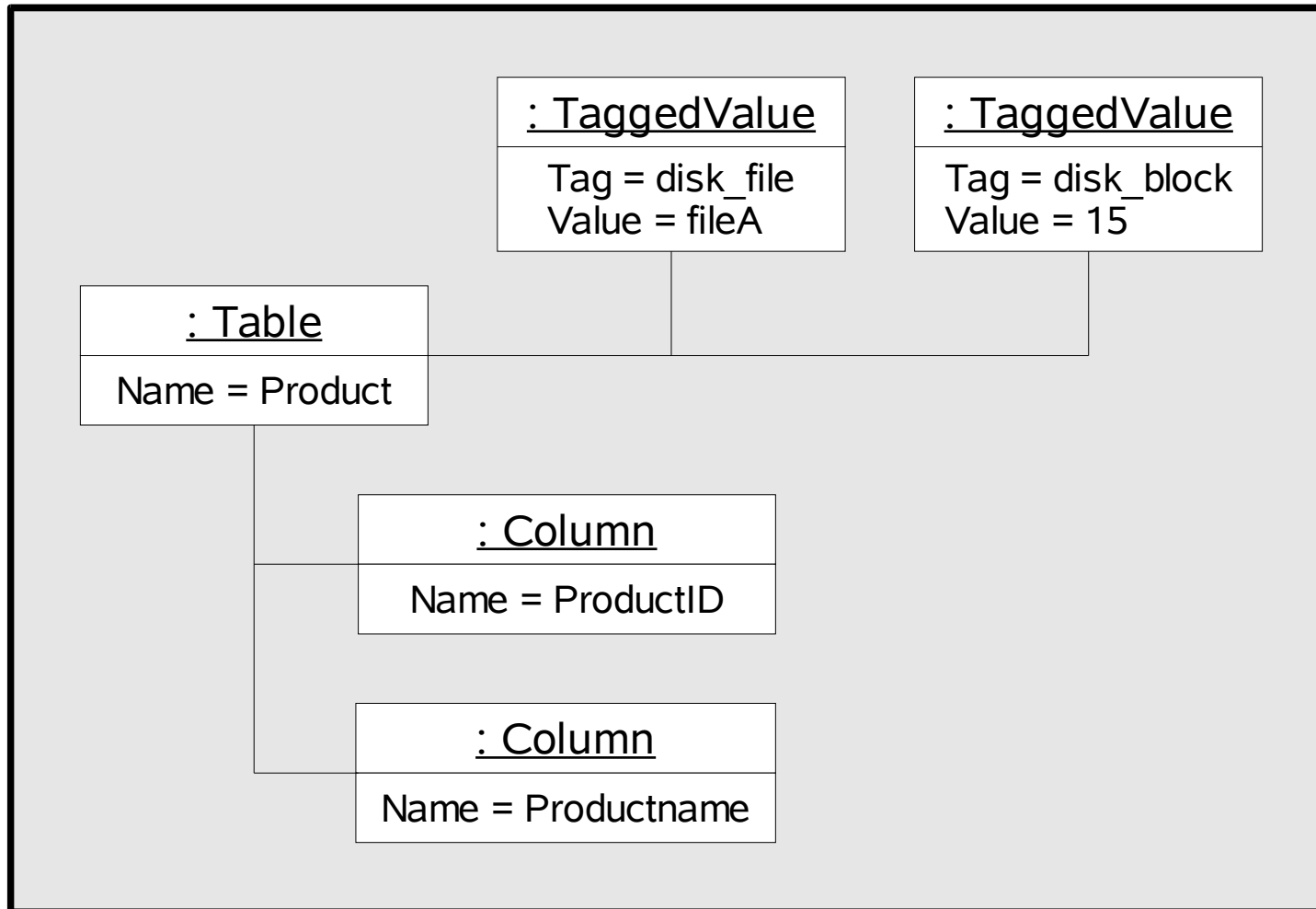
CWM Erweiterungstechniken

- 2 unterschiedliche Techniken
- Einfache Erweiterung
 - Standardkonform
 - Erweiterungsmechanismen
 - Stereotypes
 - TaggedValues
- Modellierete Erweiterungen
 - Zusätze zum Metamodell
 - Nicht mehr Standardkonform



Beispiel: TaggedValues

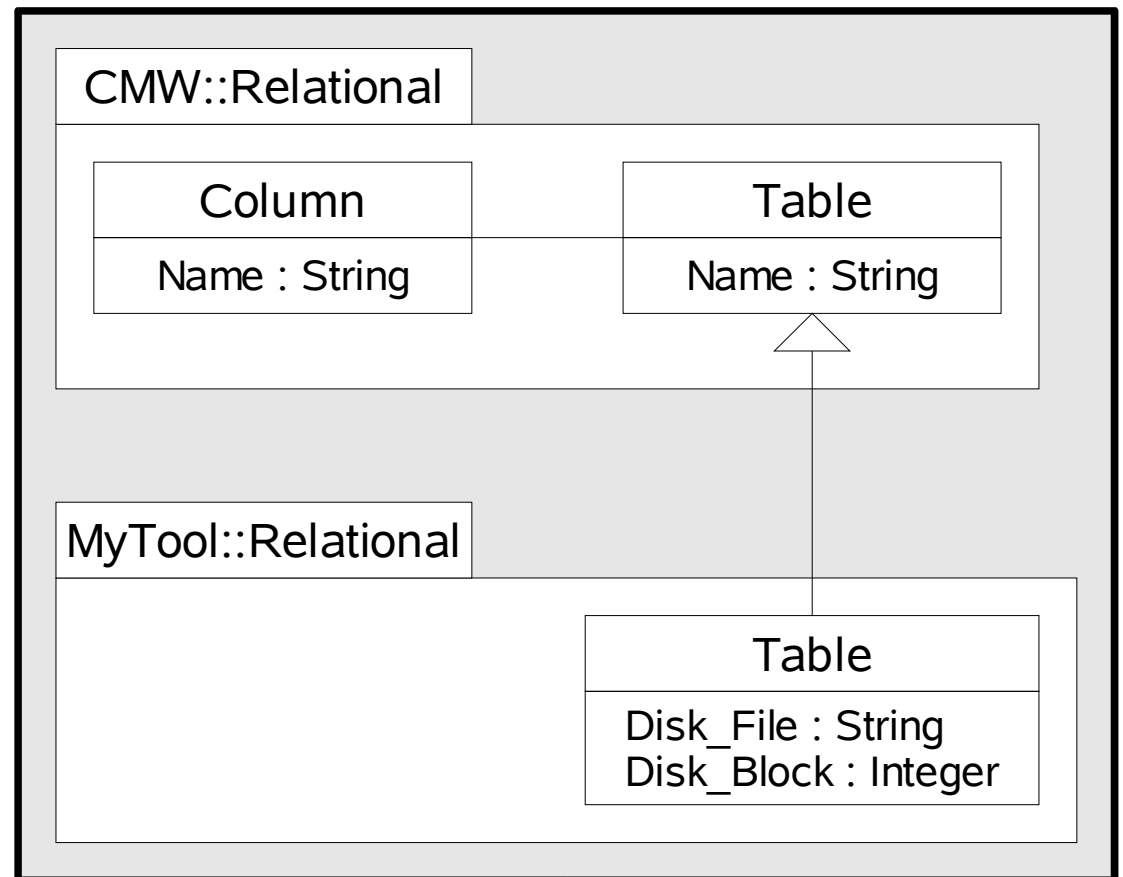
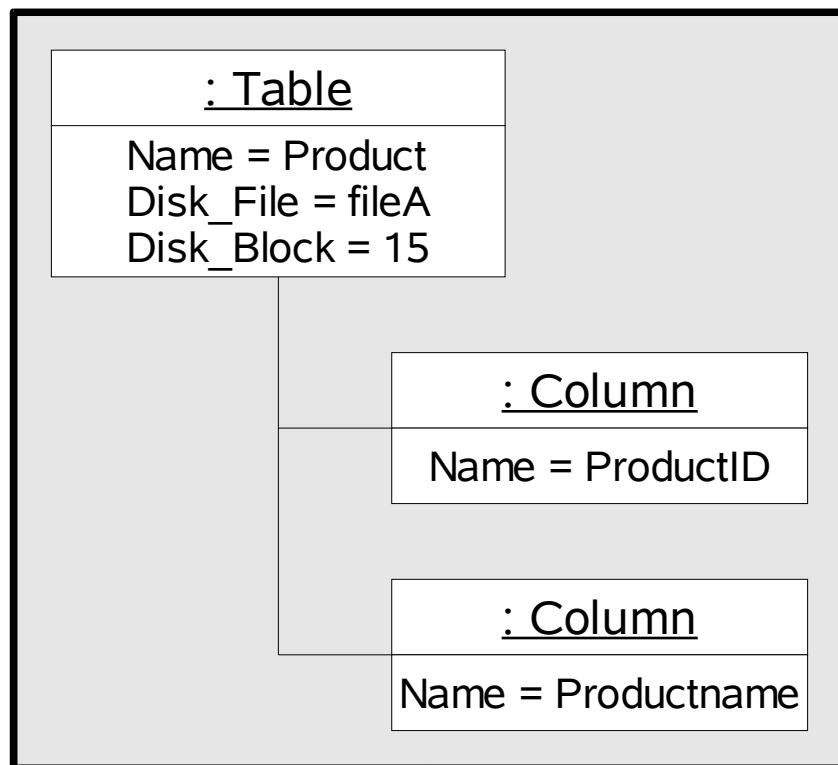
- Einfacher Erweiterungsmechanismus





Beispiel: Modellierte Erweiterung

- Erweiterung des CWM-Metamodells mit allen bekannten objektorientierten Techniken
 - z.B.: Vererbung





Vergleich Erweiterungstechniken

- Stereotypes und TaggedValues
 - Einfach
 - Nicht mächtig genug
 - Keine Typisierung
 - Geringe Ausdrucksmöglichkeiten
- Objektorientierte Techniken
 - Sehr flexibel
 - Erweiterung nicht mehr standardkonform
 - Eingeschränkte Austauschmöglichkeiten

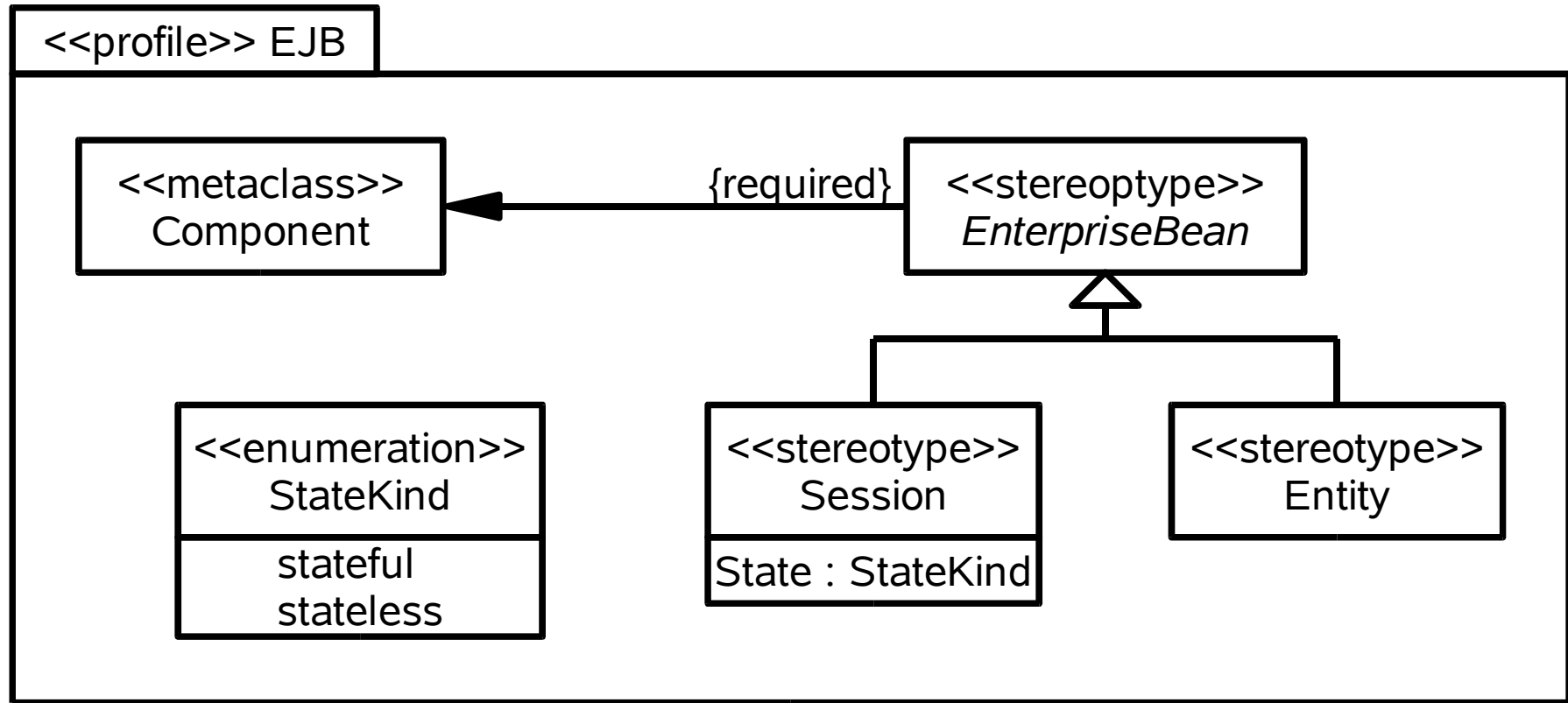


UML Version 2.0

- CWM Stereotypes und TaggedValues entsprechen UML Version 1.3 Erweiterungsmechanismus
- UML Version 2.0 veröffentlicht
 - UML 2.0 Infrastructure
 - Basiselemente
 - Neuer Erweiterungsmechanismus: **Profile**
 - UML 2.0 Superstructure
 - Kompletter Sprachraum

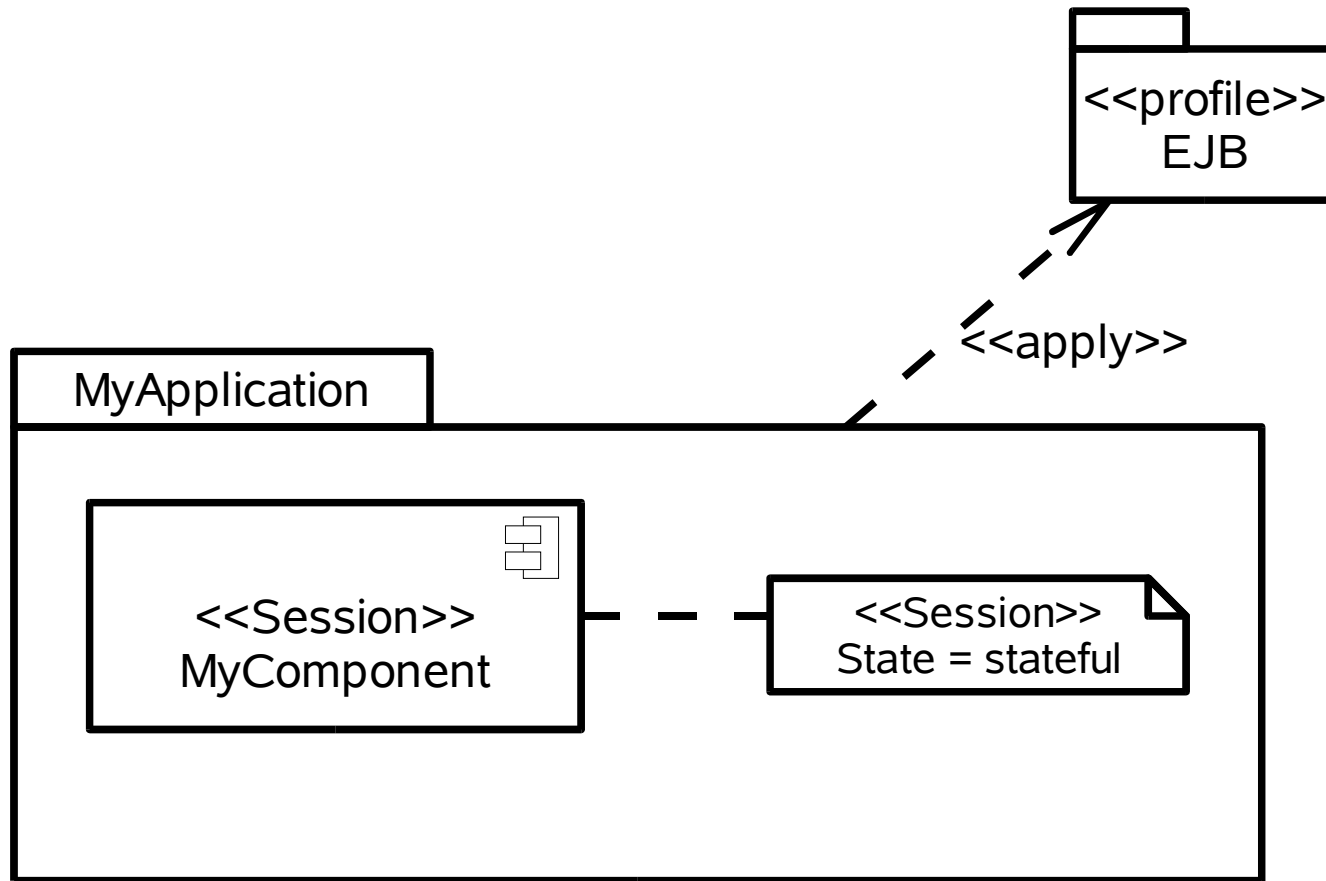


UML2 Profil: Beispiel





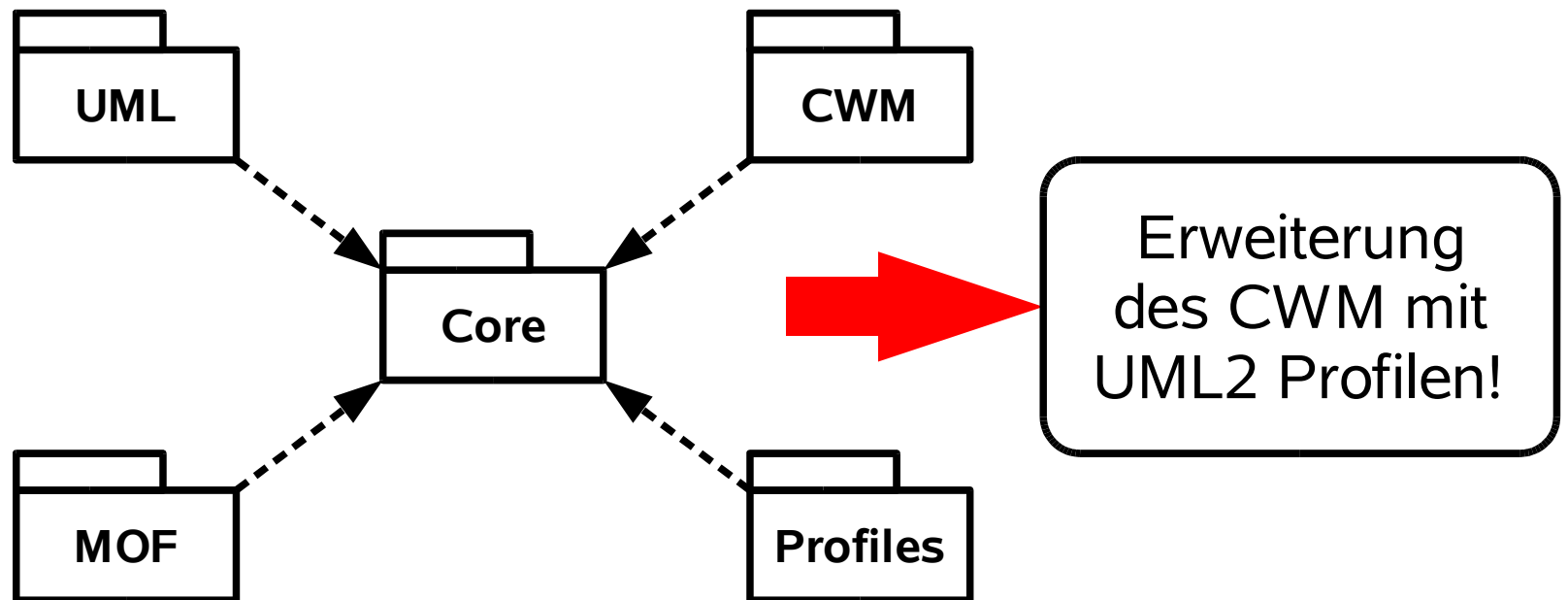
UML2 Profil: Anwendung





Zusammenhang UML - CWM

- Heute
 - CWM basiert noch auf UML Version 1.3
 - UML Version 2.0 veröffentlicht
- Zukunft





Imperfektion in Datenbanken

- Imperfekte Informationen im Verkehrsumfeld
 - **Unsichere** Informationen
 - Beispiel: Staumeldung
 - Wahrscheinlichkeitswerte
 - **Unschärfe** Informationen
 - Beispiel: Beschreibung des Verkehrsflusses
 - Klasseneinteilung: *frei, zäh, gestaut*
 - Übergang fließend und nicht exakt festgelegt
 - Linguistische Variablen
 - **Ungenaue** Informationen
 - Beispiel: physikalische Messungen

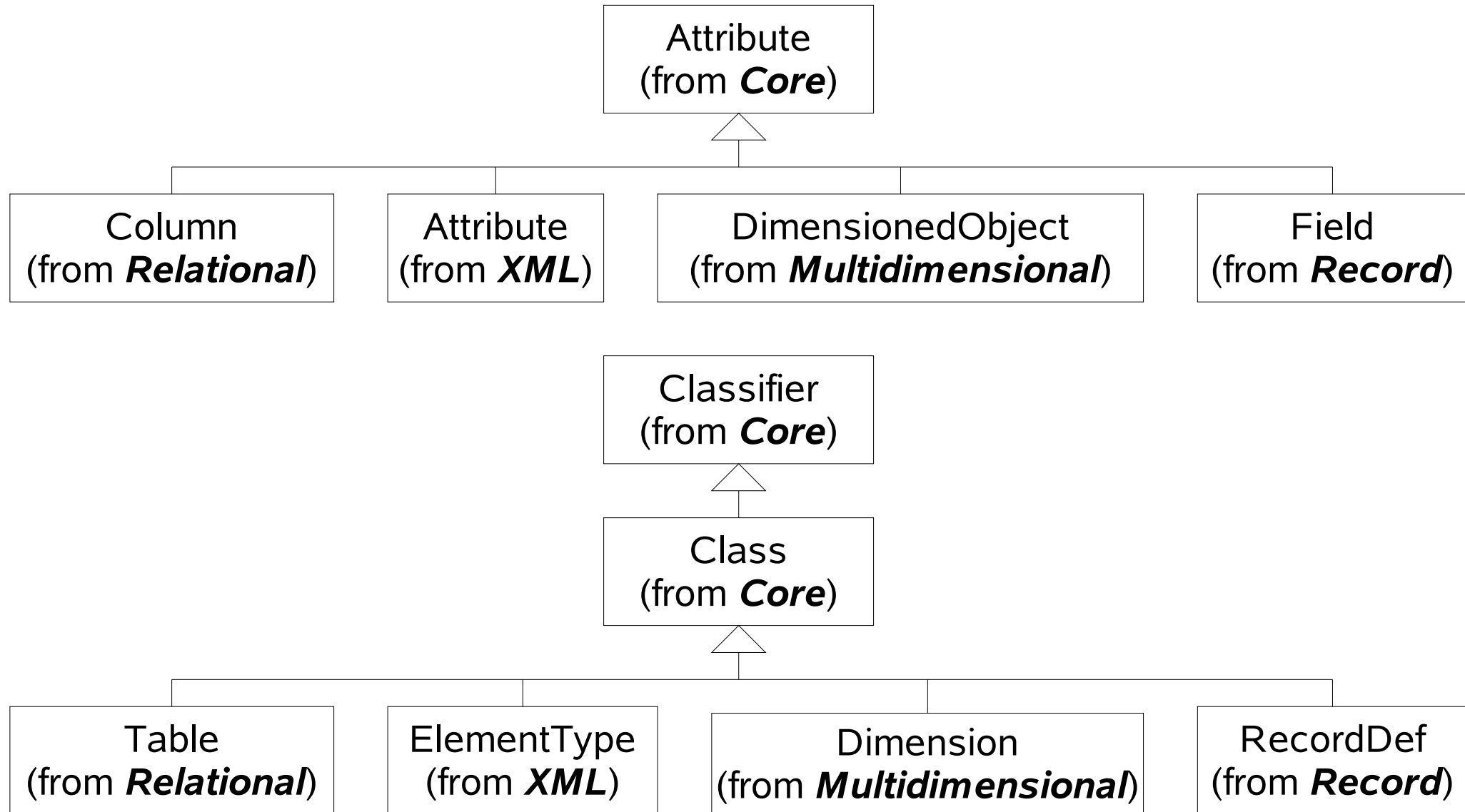


Ansatzpunkte einer Erweiterung

- Gesucht: generischer Ansatzpunkt
 - Bisherige Arbeiten
 - Erweiterung des Relational-Package um Imperfektion
 - Gewünscht: Unabhängigkeit vom Datenmodell
- Ansatzpunkte liegen nur im Core-Package
 - Keine Klasse aus der Ressourcen-Schicht geeignet
 - Gemeinsame Oberklasse in Core-Package



„Classifier equivalence“



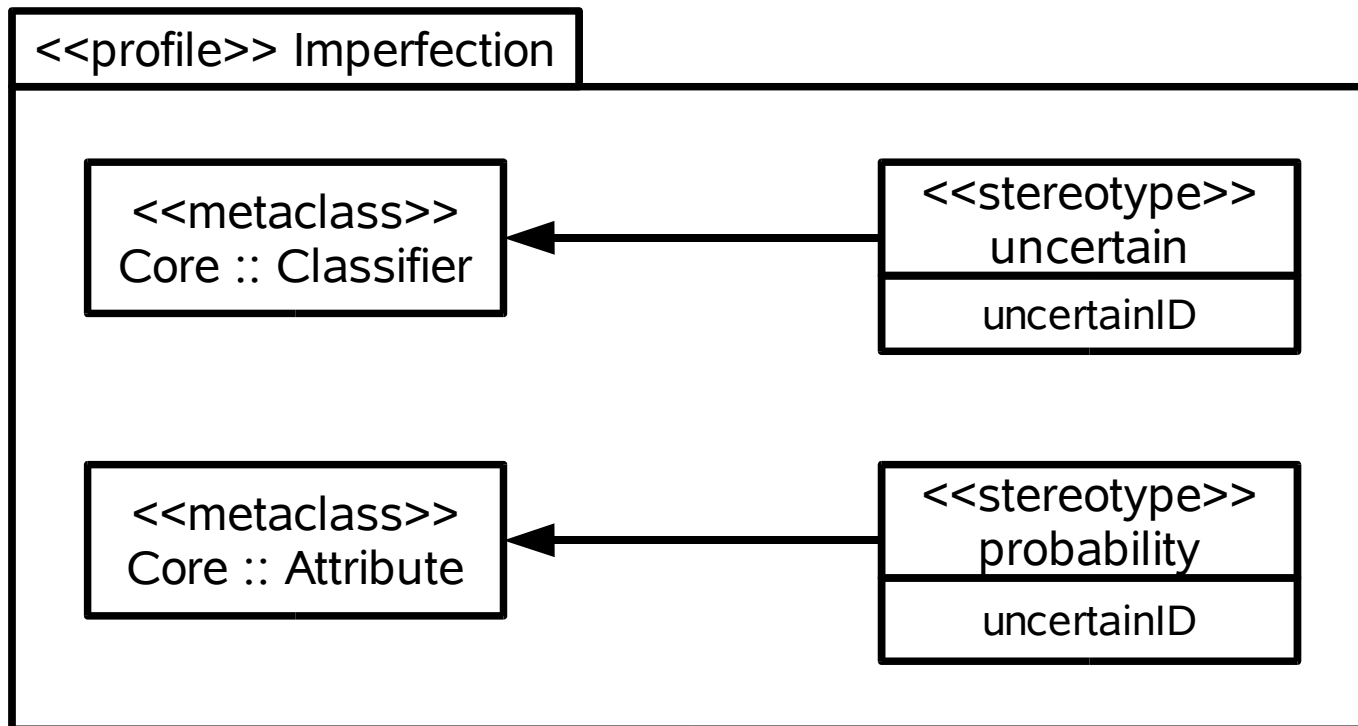


Profil <<Imperfection>>

- Verwendung des UML2 Profilmechanismus zur Entwicklung eines CWM Profils
 - Stereotypen zur Beschreibung von
 - Unsicheren Informationen
 - Unscharfen Informationen
 - Nicht untersucht
 - Ungenaue Informationen
 - Jetzt vorgestellt
 - Nur Stereotypen für unsichere Informationen



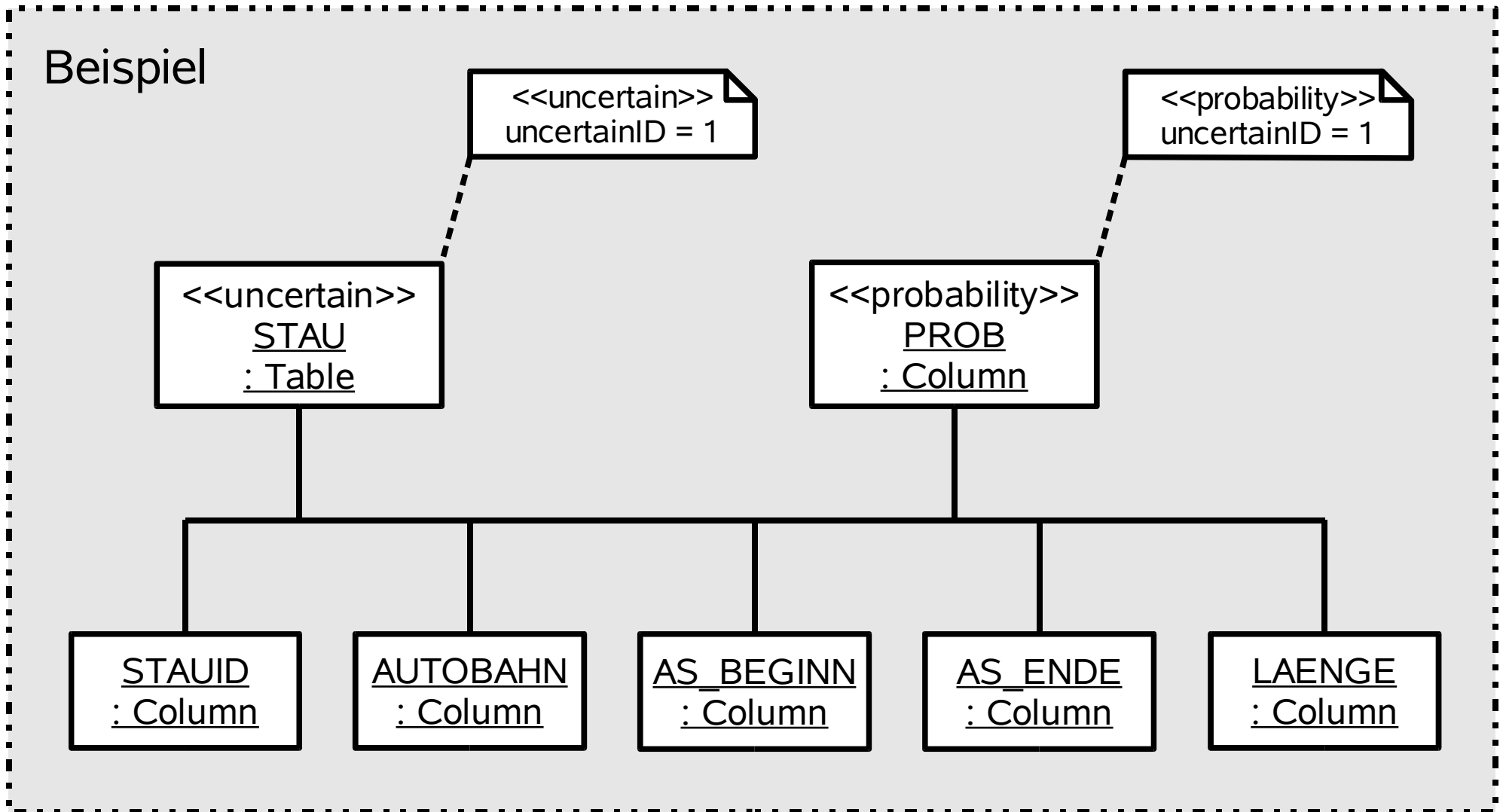
Stereotyp <<uncertain>>





Stereotyp <<uncertain>>: Beispiel

Beispiel



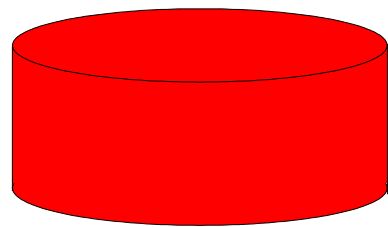


Visualisierungstool

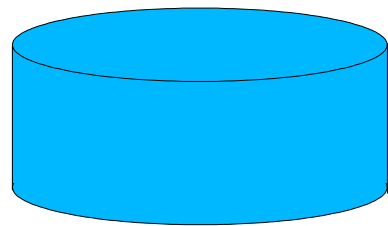
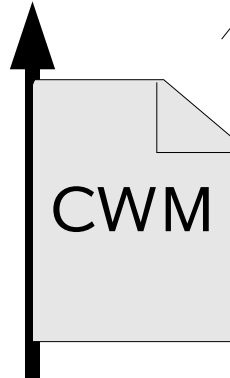
- Graphisches Analyse-Werkzeug:
Oliver Forsters Studienarbeit
 - Schnittstelle zur Beschreibung imperfekter Informationen
 - Bisher: Daten aus Textdateien
 - Neu: Import der Daten aus Datenbank, die CWM-konform beschrieben sind



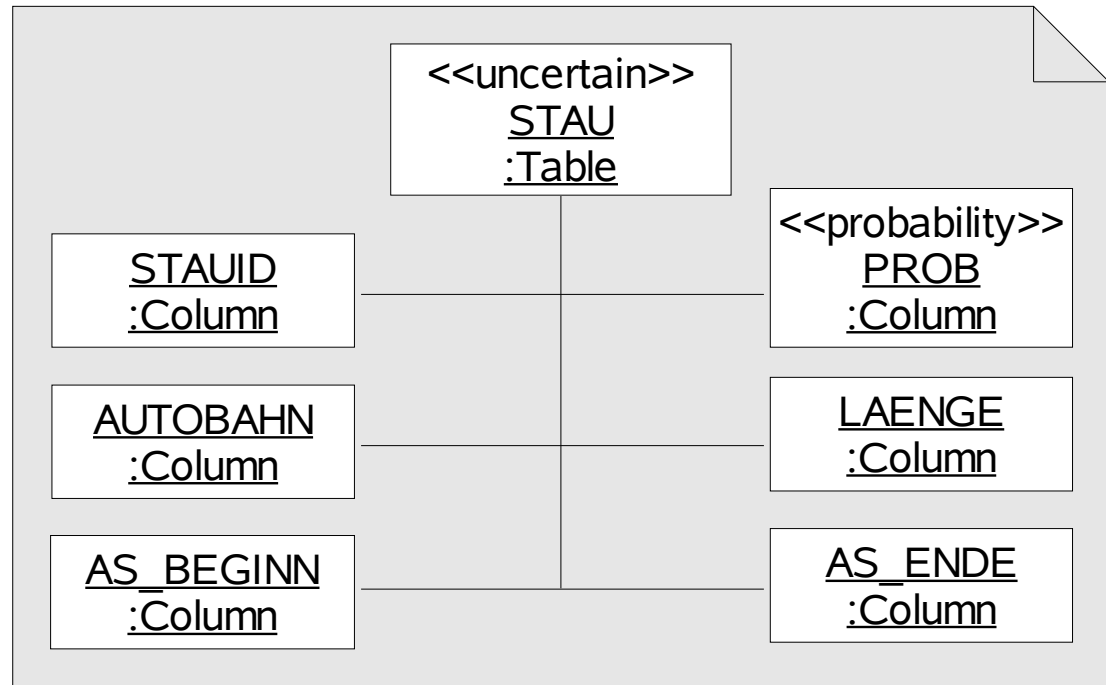
Zusammenfassung



Graphisches
Analyse-Tool



Stau-DB



STAU					
STAUID	AUTOBAHN	AS_BEGINN	AS_ENDE	LAENGE	PROB
1	A5	Durlach	Bruchsal	5 km	100 %
2	A8	Pforzheim	Karlsbad	2 km	40 %
...					



Ausblick

- CWM Version 2.0
- Weiterentwicklung CWM-Import
 - Anpassung an zukünftiges CWM 2.0
 - Unterstützung von mehreren Datenmodellen
 - Profil <<Imperfection>> ist generisch



Vielen Dank





Ende

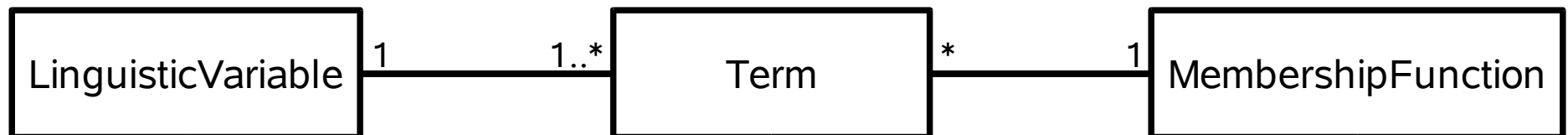
Ende der Bildschirmpräsentation



Gründe für UML2 Profil

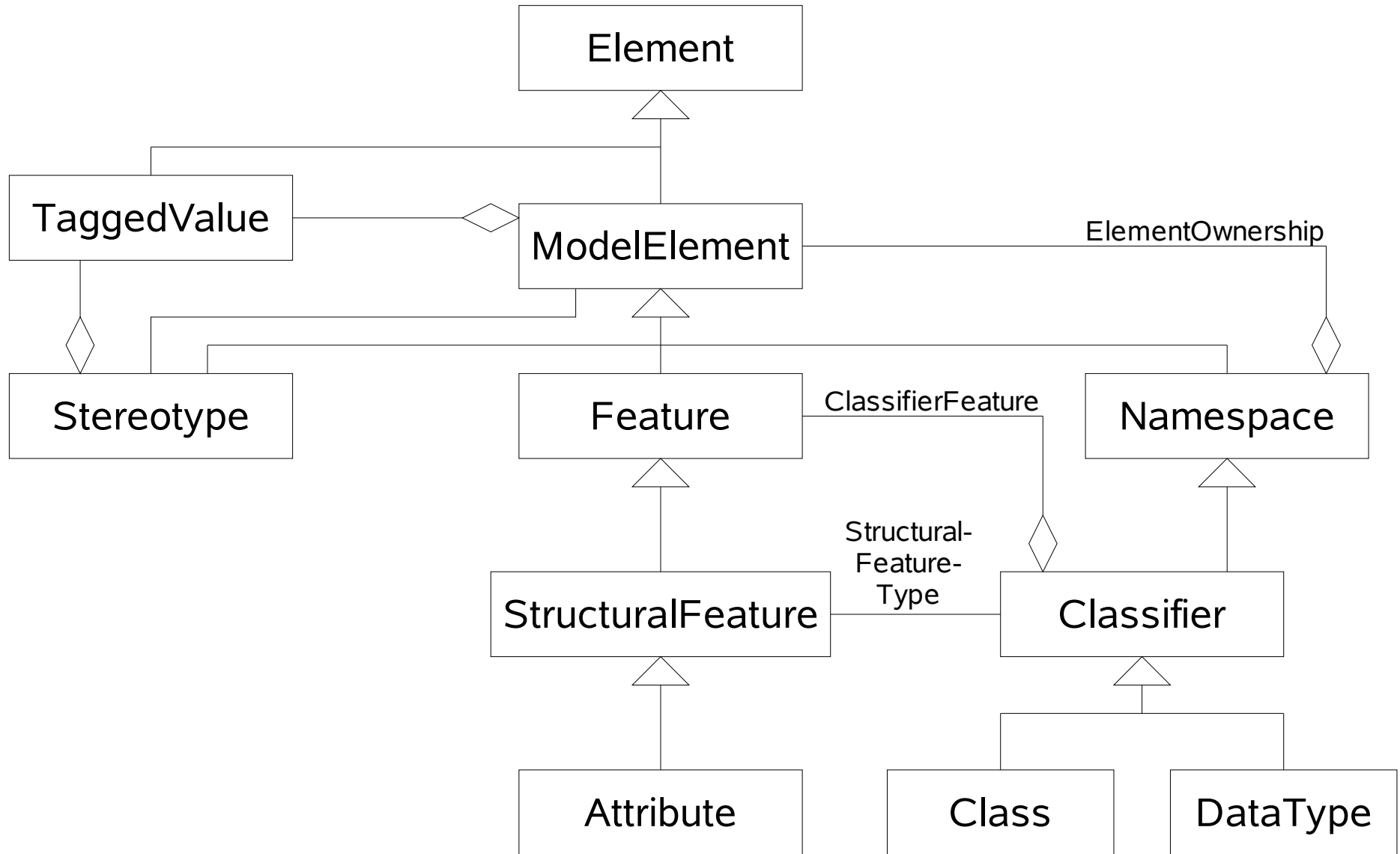
- CWM 2.0 und UML 2.0 basieren auf gleichem Core (UML2 Infrastructure)
 - Profile für alle Metamodelle, die auf Core basieren (Quelle: UML 2.0 Infrastructure Spezifikation)
- Mächtigeres Konzept als CWM Stereotypes und TaggedValues

- Einteilung von Sachverhalten in unscharfe Mengen
 - Charakteristische Funktion nicht auf die Werte 0 und 1 beschränkt
- Linguistische Variable
 - Formalisierung umgangssprachlicher Ausdrücke



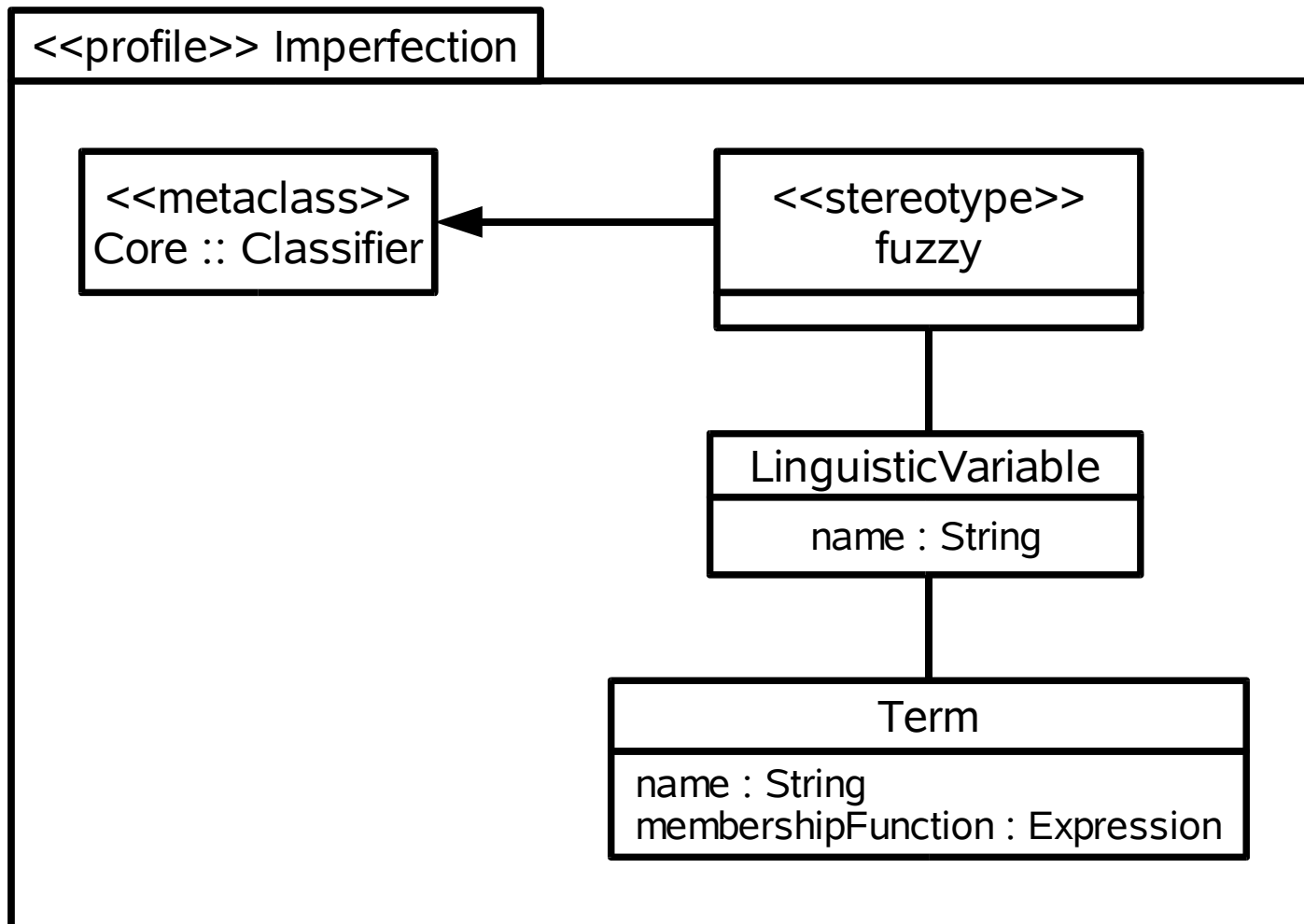


Core Package



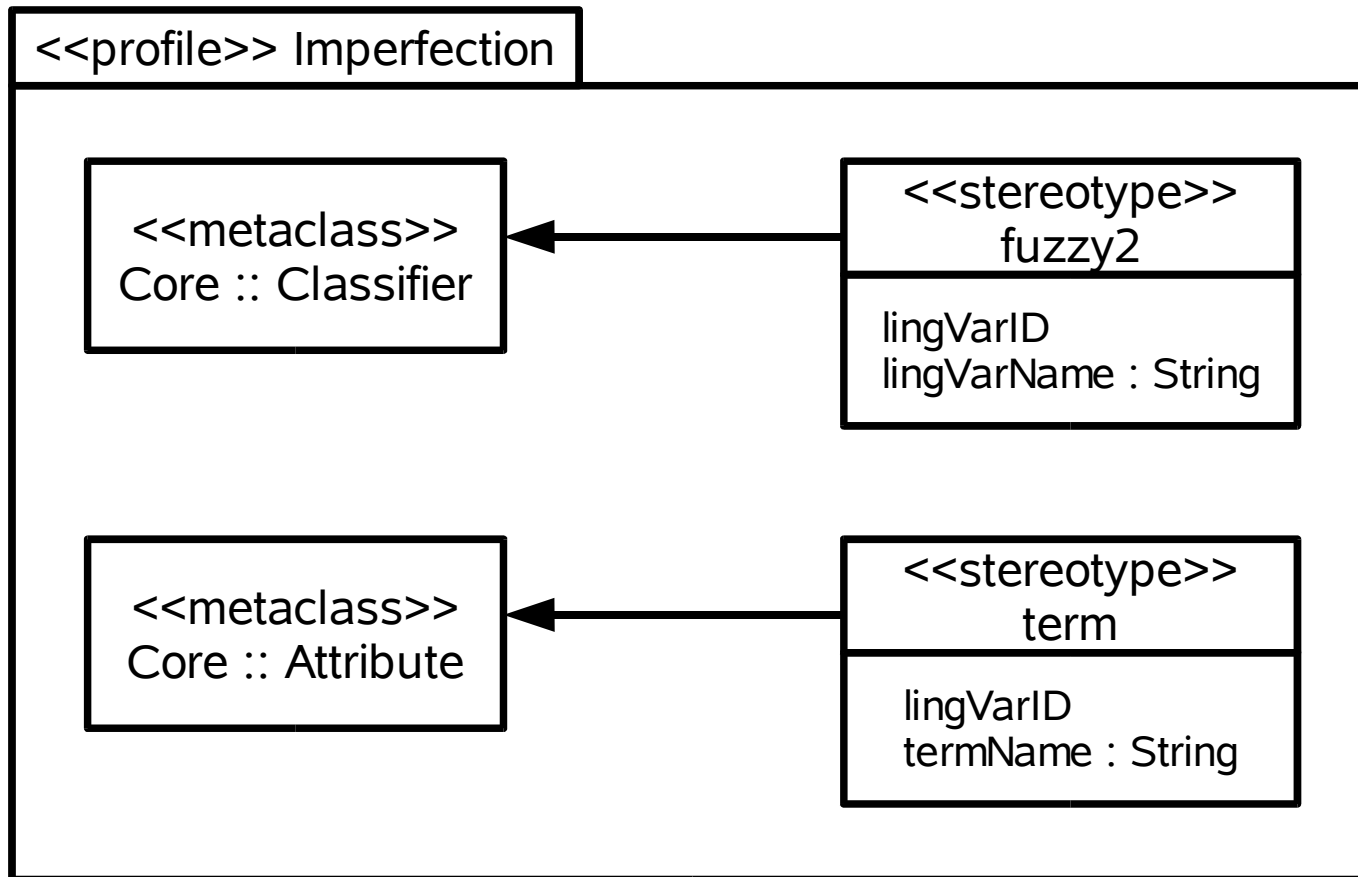


Stereotyp <<fuzzy>>



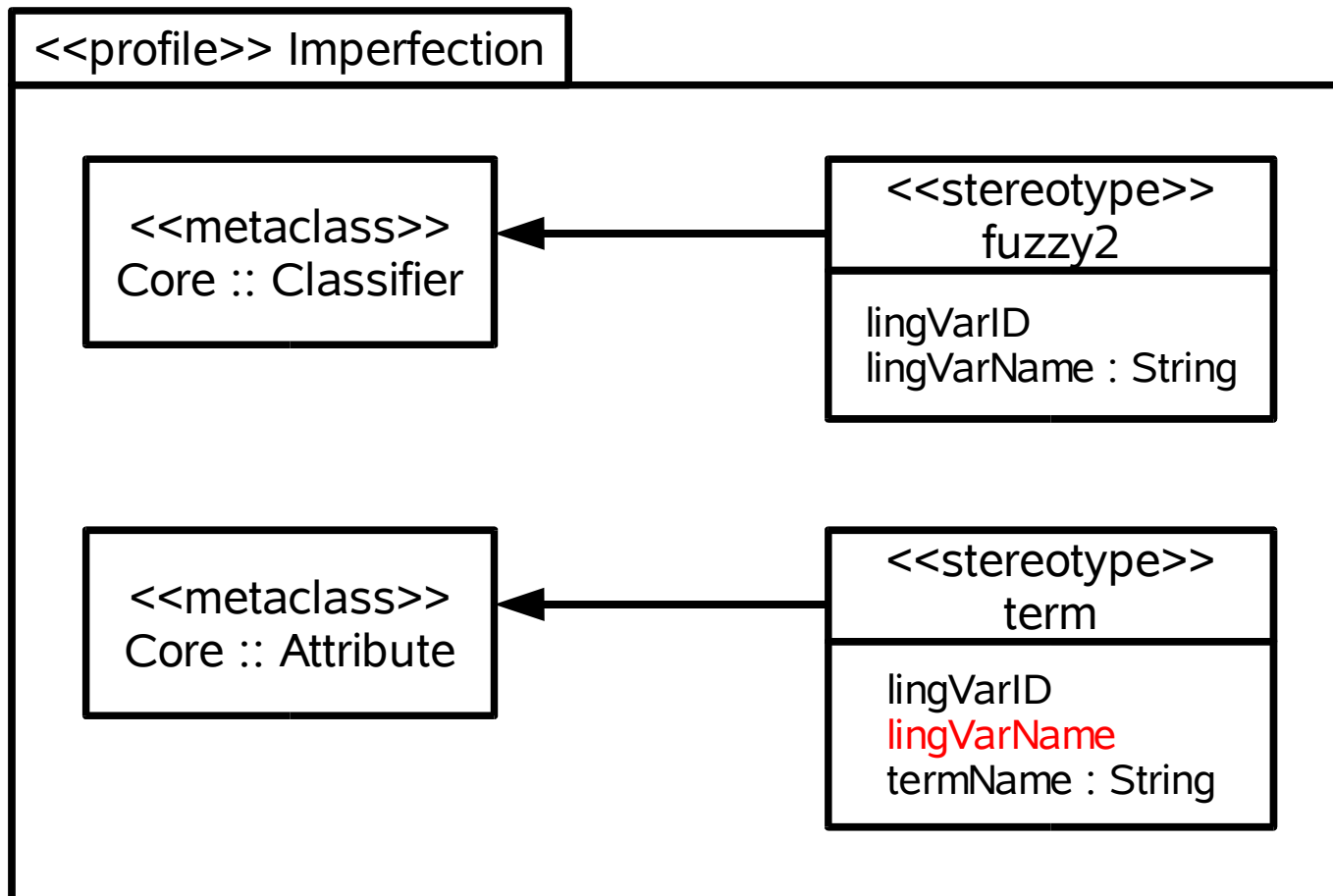


Stereotyp <<fuzzy2>>





Anpassung Stereotyp <<fuzzy2>>





Umsetzung von Profil und Modell

- Problem
 - Modellierung mit graphischen Werkzeug
 - Keine Erweiterung von CWM Metaklassen
 - Vollständige Codierung von Hand zu aufwändig
- Lösung
 - Simulation mit standardisierten CWM Mitteln
 - Stereotypes und TaggedValues
 - Objektorientierte Erweiterung



Visualisierungstool: CWM-Import

