

Entwicklung eines Matching- und Mappingverfahrens zur Verbesserung der XML-Schemaevolution

Jan Deffke

Lehrstuhl für Datenbank- und Informationssysteme

26. September 2013

Erstgutachter: PD Dr. -Ing. habil. Meike Klettke

Zweitgutachter: Prof. Dr. Kurt Sandkuhl

Betreuer: Dipl. -Inf. Thomas Nösinger

Gliederung

- 1 Einstieg in die Thematik
 - XML-Schemaevolution
 - Verwendete Techniken/Umgebung
 - Ziele der Masterarbeit

Gliederung

- 1 Einstieg in die Thematik
 - XML-Schemaevolution
 - Verwendete Techniken/Umgebung
 - Ziele der Masterarbeit
- 2 Schema Matching
 - Klassifikation
 - Umsetzung in der Praxis
 - Eignung für eigenen Ansatz

Gliederung

- 1 Einstieg in die Thematik
 - XML-Schemaevolution
 - Verwendete Techniken/Umgebung
 - Ziele der Masterarbeit
- 2 Schema Matching
 - Klassifikation
 - Umsetzung in der Praxis
 - Eignung für eigenen Ansatz
- 3 Integration in CodeX
 - Allgemeine Architektur
 - Implementierung

Gliederung

- 1 Einstieg in die Thematik
 - XML-Schemaevolution
 - Verwendete Techniken/Umgebung
 - Ziele der Masterarbeit
- 2 Schema Matching
 - Klassifikation
 - Umsetzung in der Praxis
 - Eignung für eigenen Ansatz
- 3 Integration in CodeX
 - Allgemeine Architektur
 - Implementierung
- 4 Evaluierung
 - Evolutionsszenario
 - CodeXMapper vs. Evolutionsszenario
 - CodeXMapper vs. CodeX

Gliederung

- 1 **Einstieg in die Thematik**
 - XML-Schemaevolution
 - Verwendete Techniken/Umgebung
 - Ziele der Masterarbeit
- 2 **Schema Matching**
 - Klassifikation
 - Umsetzung in der Praxis
 - Eignung für eigenen Ansatz
- 3 **Integration in CodeX**
 - Allgemeine Architektur
 - Implementierung
- 4 **Evaluierung**
 - Evolutionsszenario
 - CodeXMapper vs. Evolutionsszenario
 - CodeXMapper vs. CodeX
- 5 **Zusammenfassung**
 - Umsetzung der Ziele
 - Ausblick

Entwicklung eines Matching- und Mappingverfahrens zur Verbesserung der XML-Schemaevolution

Jan Deffke

Einstieg in die Thematik

XML-Schemaevolution

Verwendete Techniken/Umgebung
Ziele der Masterarbeit

Schema Matching

Klassifikation
Umsetzung in der Praxis
Eignung für eigenen Ansatz

Integration in CodeX

Allgemeine Architektur
Implementierung

Evaluierung

Evolutionsszenario
CodeXMapper vs. Evolutionsszenario
CodeXMapper vs. CodeX

Zusammenfassung

Umsetzung der Ziele
Ausblick

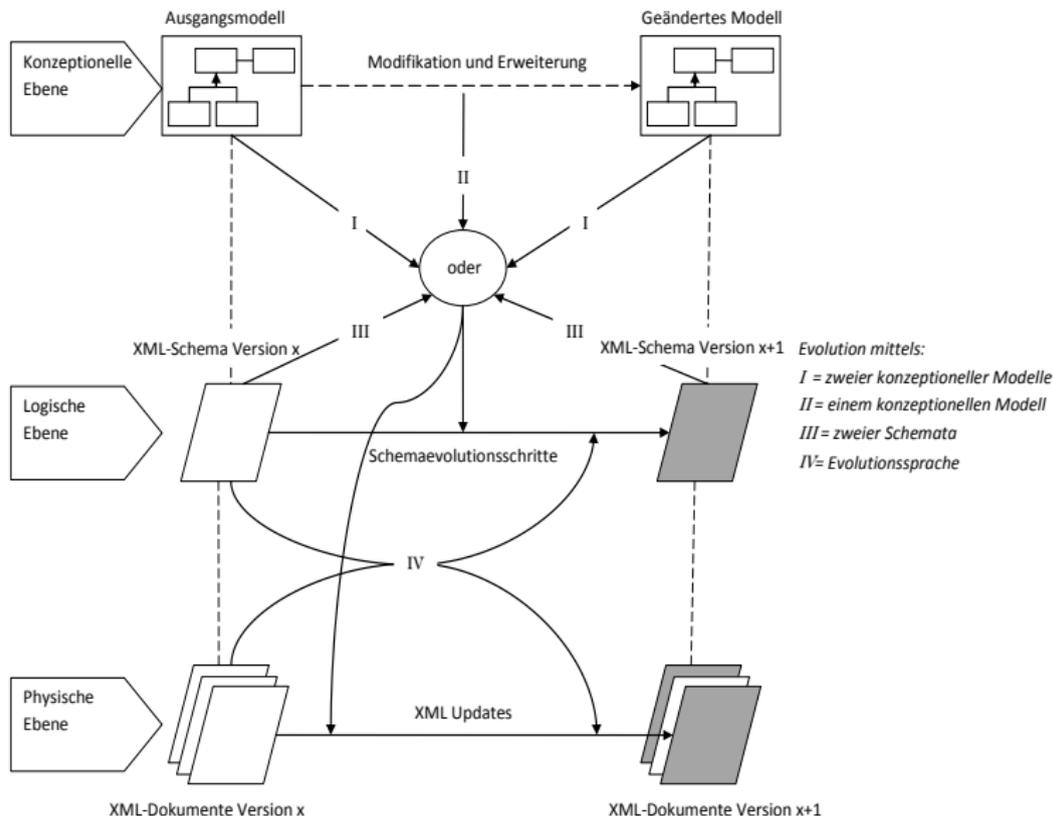


Figure: XML-Schemaevolution

Entwicklung eines Matching- und Mappingverfahrens zur Verbesserung der XML-Schemaevolution

Jan Deffke

Einstieg in die Thematik

XML-Schemaevolution

Verwendete Techniken/Umgebung
Ziele der Masterarbeit

Schema Matching

Klassifikation
Umsetzung in der Praxis
Eignung für eigenen Ansatz

Integration in CodeX

Allgemeine Architektur
Implementierung

Evaluierung

Evolutionsszenario
CodeXMapper vs. Evolutionsszenario
CodeXMapper vs. CodeX

Zusammenfassung

Umsetzung der Ziele
Ausblick

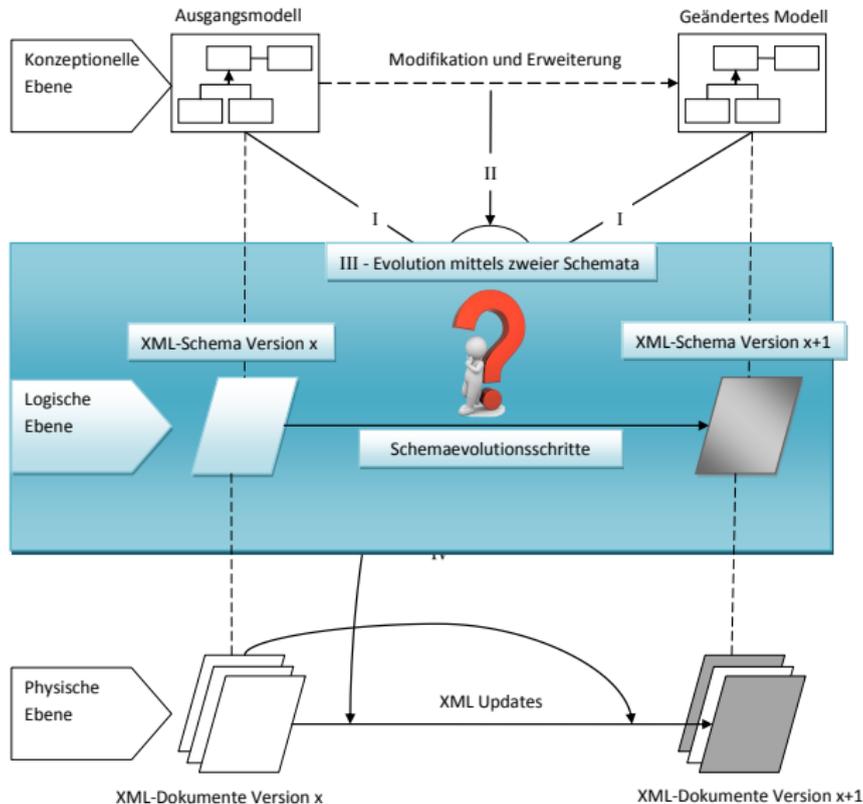


Figure: Evolution mittels zweier Schemata

Entwicklung eines Matching- und Mappingverfahrens zur Verbesserung der XML-Schemaevolution

Jan Deffke

Einstieg in die Thematik

XML-Schemaevolution

Verwendete Techniken/Umgebung
Ziele der Masterarbeit

Schema Matching

Klassifikation
Umsetzung in der Praxis
Eignung für eigenen Ansatz

Integration in CodeX

Allgemeine Architektur
Implementierung

Evaluierung

Evolutionsszenario
CodeXMapper vs. Evolutionsszenario
CodeXMapper vs. CodeX

Zusammenfassung

Umsetzung der Ziele
Ausblick

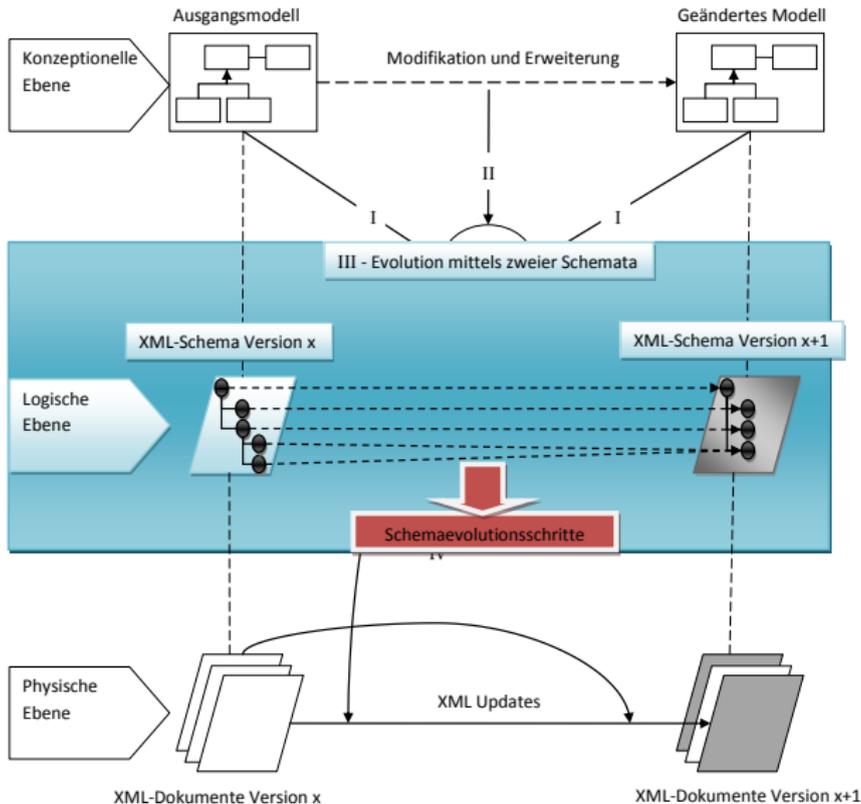


Figure: Schema Matching und Mapping

- CodeX(Conceptual Design and Evolution for XML-Schema)
 - Grundlage: konzeptionelles Modell: EMX (Entity Model for XML-Schema)

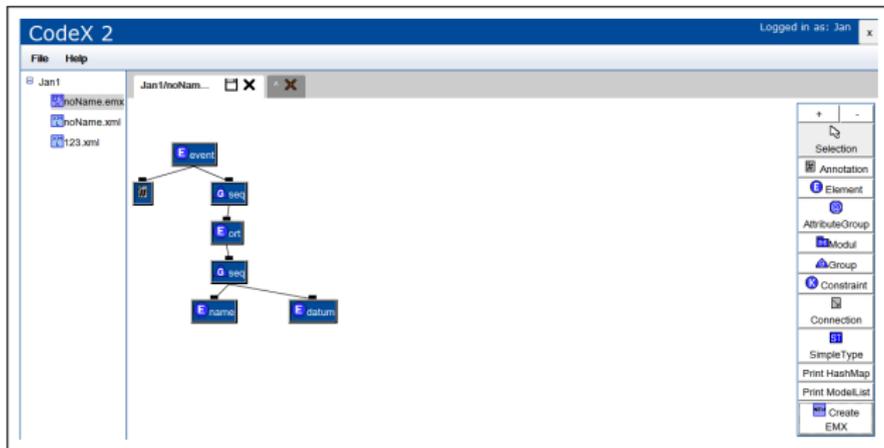


Figure: Screenshot aus dem CodeX-Editor

- ELaX (Evolution Language for XML-Schema)
 - Ziel: (formale) Anpassungen von XML-Schemata und Loggen von Evolutionsschritten
 - `elax ::= ((< add > | < delete > | < update >)" ;")+;`
 - komplette Spezifikation: www.ls-dbis.de/elax

Entwicklung eines Matching- und Mappingverfahrens zur Verbesserung der XML-Schemaevolution

Jan Deffke

Einstieg in die Thematik

XML-Schemaevolution
Verwendete Techniken/Umgebung
Ziele der Masterarbeit

Schema Matching

Klassifikation
Umsetzung in der Praxis
Eignung für eigenen Ansatz

Integration in CodeX

Allgemeine Architektur
Implementierung

Evaluierung

Evolutionsszenario
CodeXMapper vs. Evolutionsszenario
CodeXMapper vs. CodeX

Zusammenfassung

Umsetzung der Ziele
Ausblick

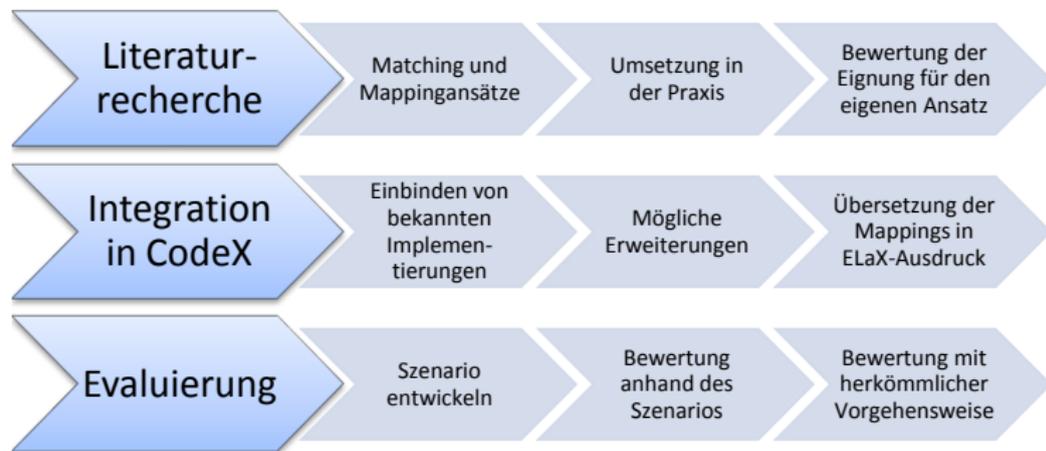


Figure: Ziele der Masterarbeit

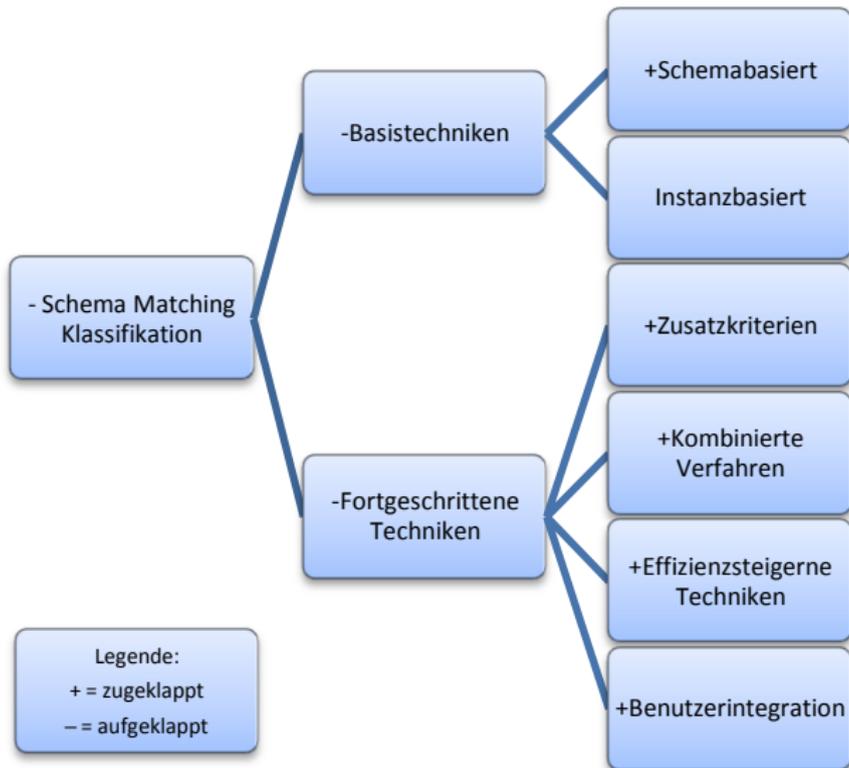


Figure: Schema Matching Klassifikation

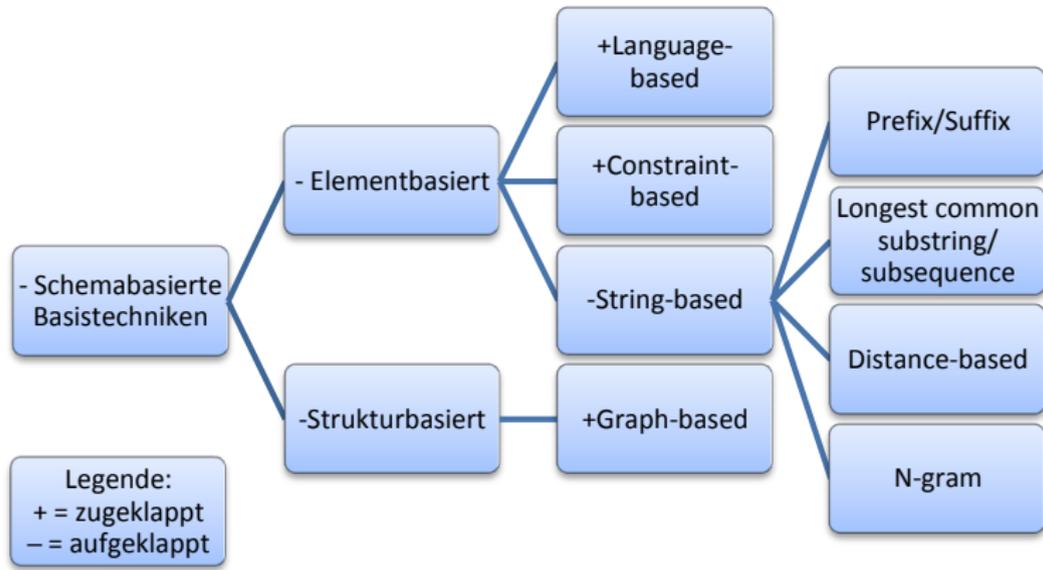


Figure: Schemabasierte Basistechniken

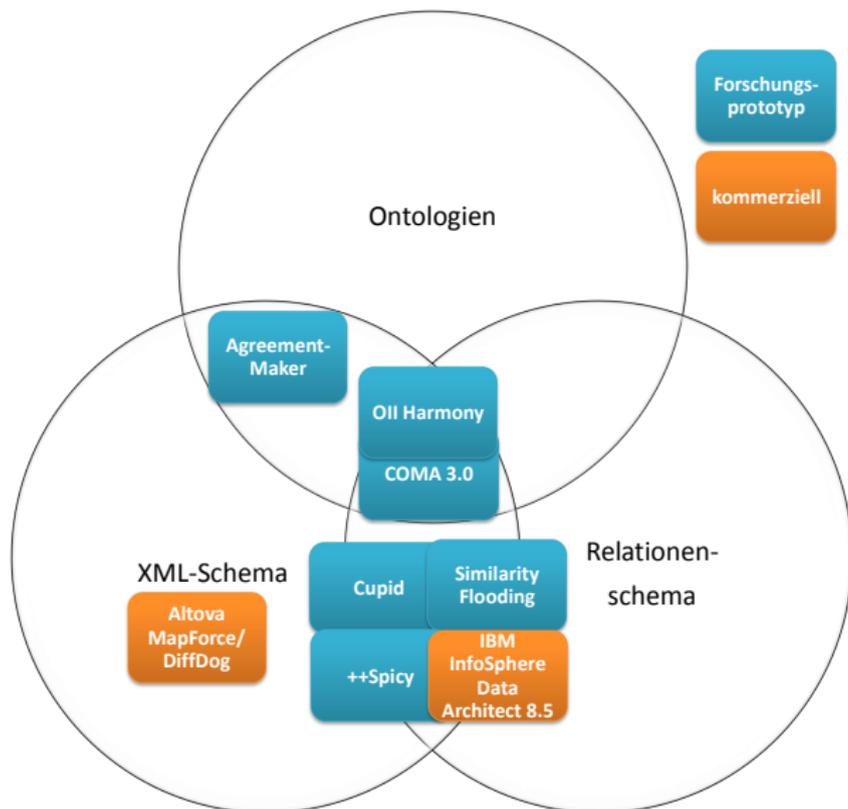


Figure: Umsetzung in der Praxis

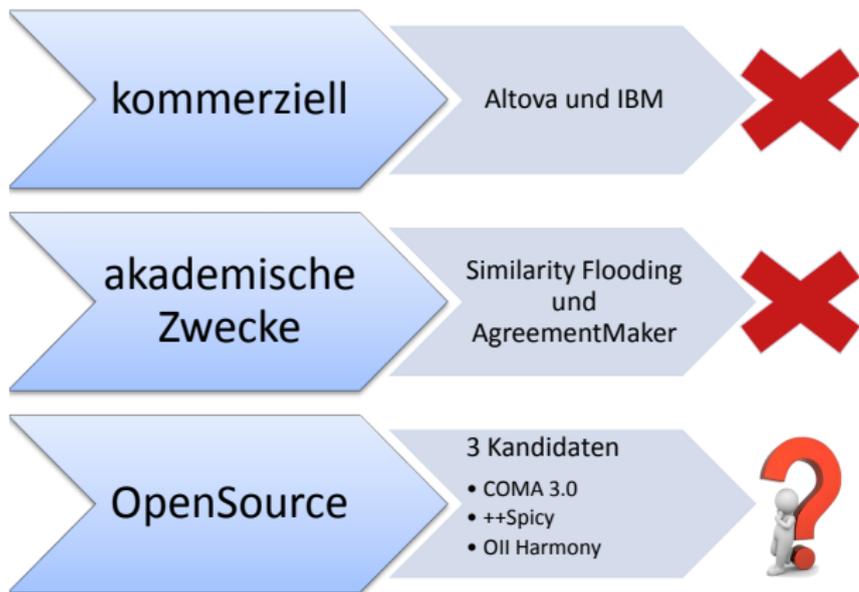


Figure: Eignung für eigenen Ansatz 1/4

Kriterien	COMA 3.0	++Spicy	Oll Harmony
XML- Unterstützung	ja	ja	ja
Basistechniken (schemabasiert)	String-, Language, Constraint- und Graph-based	nur Graph- based	String-, Language, Constraint- und Graph-based

Figure: Eignung für eigenen Ansatz 2/4

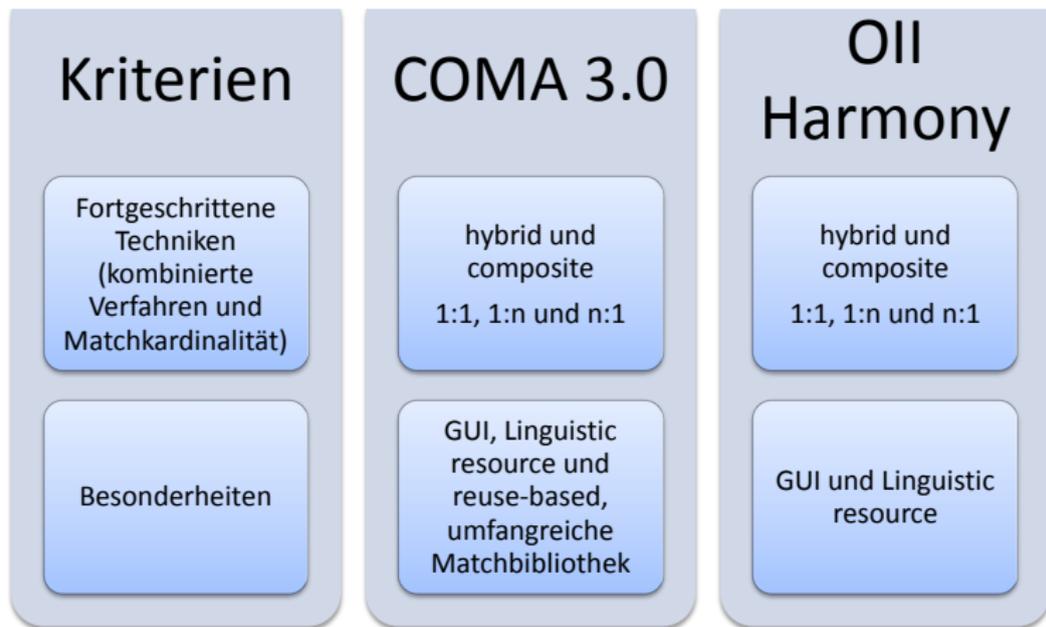


Figure: Eignung für eigenen Ansatz 3/4

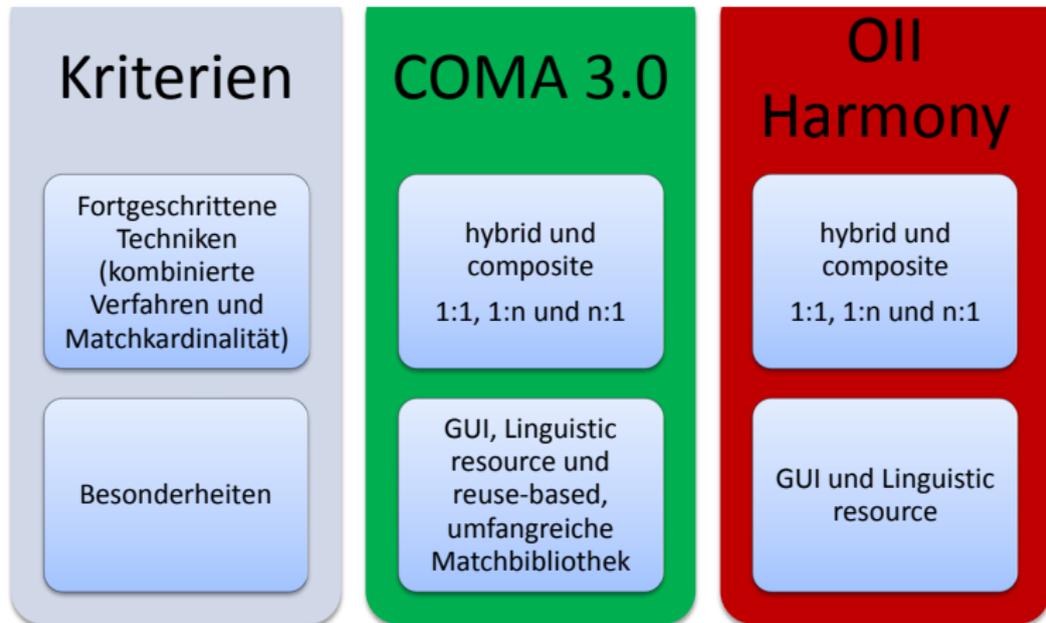


Figure: Eignung für eigenen Ansatz 4/4

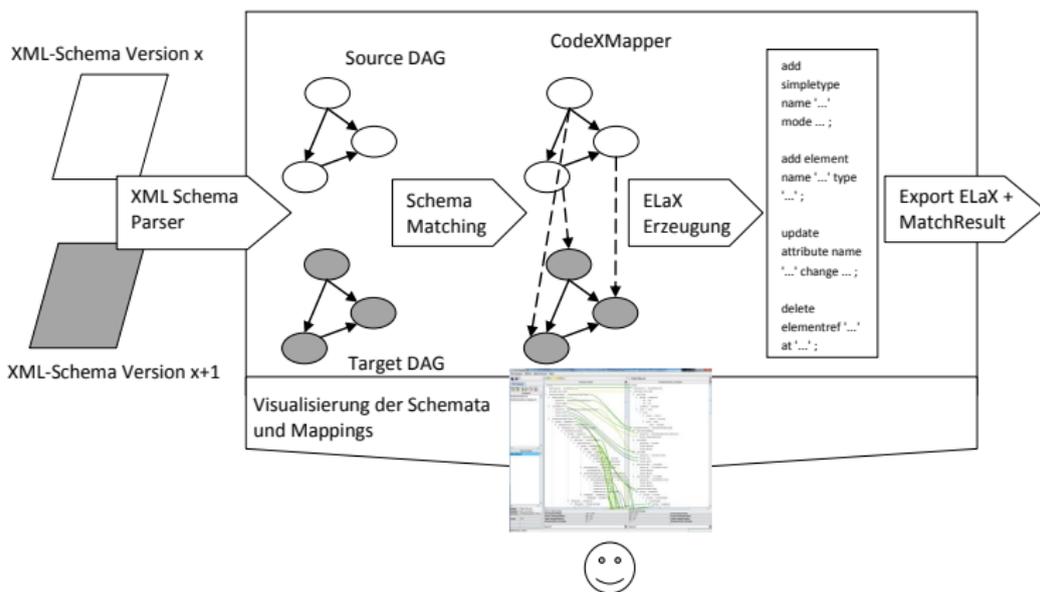


Figure: Allgemeine Architektur von CodeXMapper

- Programmiersprache: Java
- Werkzeuge
 - Eclipse
 - GWT (Google Web Toolkit)
- Anforderungen
 - XML-Schemata befinden sich im Modellierungsstil *Garden of Eden* (gewährleistet durch BA von Kaping)
 - automatische Erzeugung von Mappings und Möglichkeit der manuellen Korrigierung und Ergänzung
 - Ergebnis: Liste von ELaX-Ausdrücken
 - Vergleich Ergebnis mit den ELaX-Ausdrücken von CodeX

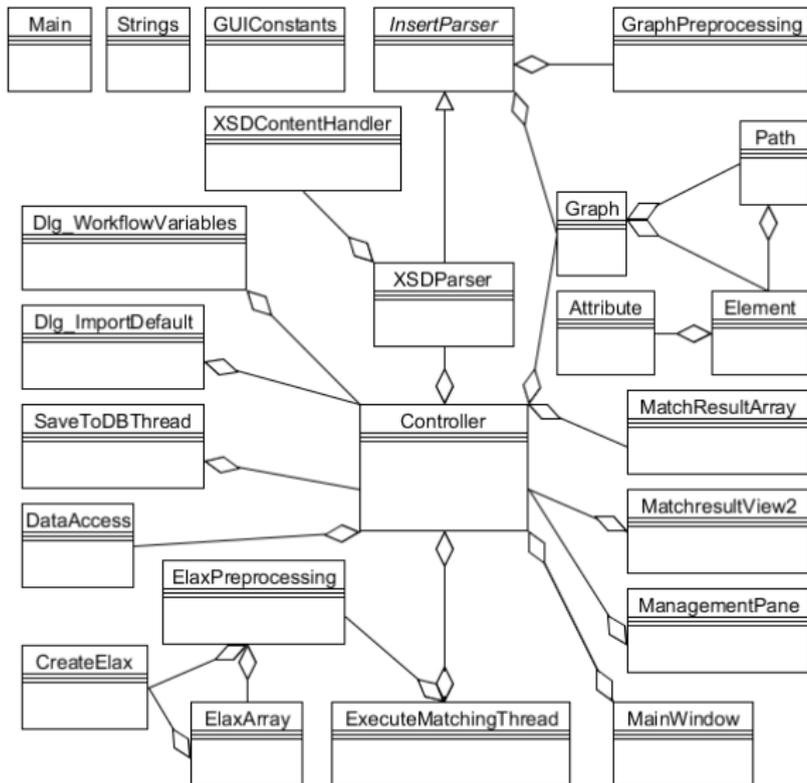


Figure: Klassendiagramm von CodeXMapper

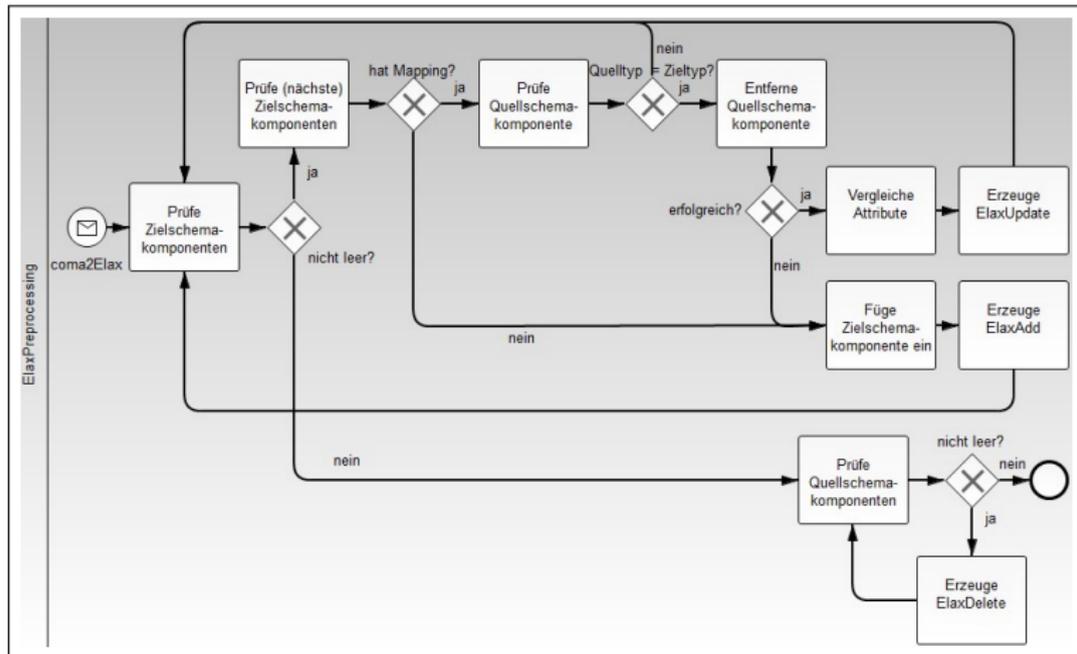


Figure: Ablaufprozess der ELaX-Erzeugung

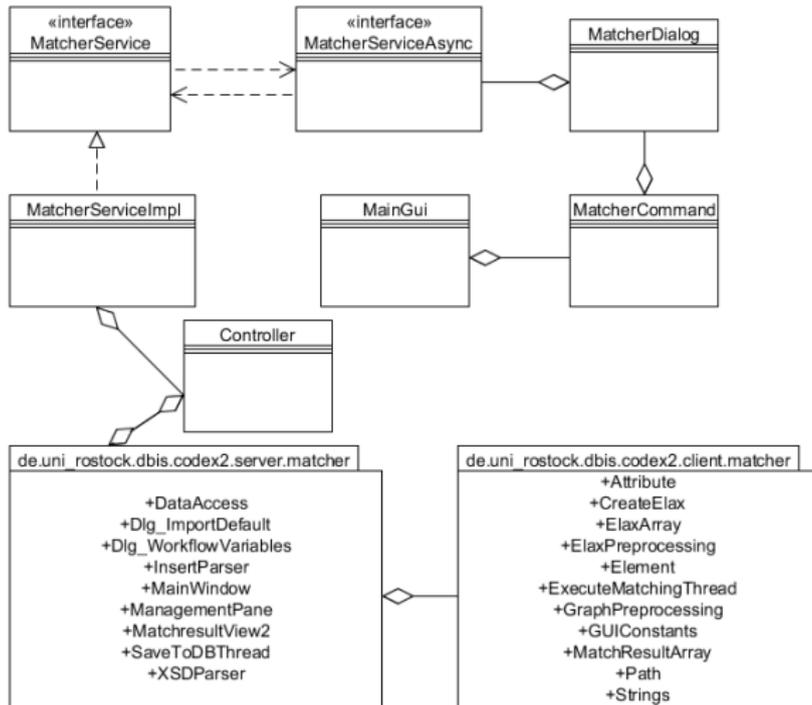


Figure: Klassendiagramm nach Integration von CodeXMapper in CodeX

Kurzpräsentation

- 1 CodeXMapper
- 2 Codex

Entwicklung eines Matching- und Mappingverfahrens zur Verbesserung der XML-Schemaevolution

Jan Deffke

Einstieg in die Thematik

XML-Schemaevolution
Verwendete Techniken/Umgebung
Ziele der Masterarbeit

Schema Matching

Klassifikation
Umsetzung in der Praxis
Eignung für eigenen Ansatz

Integration in CodeX

Allgemeine Architektur
Implementierung

Evaluierung

Evolutionszenario
CodeXMapper vs. Evolutionszenario
CodeXMapper vs. CodeX

Zusammenfassung

Umsetzung der Ziele
Ausblick



Figure: Evolutionszenario: Fernbuslinien, entnommen aus <http://meinfernbus.de/>

Selbstgewähltes Beispiel

- Anlass
 - Lockerung Zulassung Fernbuslinien seit 01.01.2013
- Szenario
 - Altes XML-Schema anpassen an neue Anforderungen
 - Z.B. Optimierung der Speicherstruktur für Suchanfragen

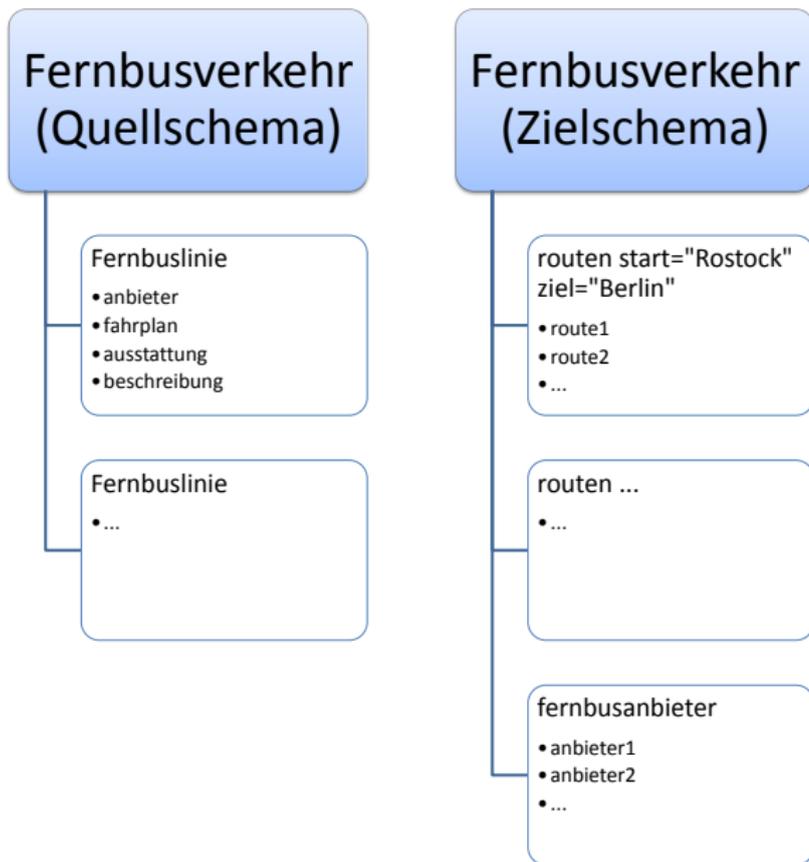


Figure: Selbstgewähltes Beispiel

- CodeXMapper vs. Evolutionsszenario
 - ELaX-Ausdrücke ergeben nach Ausführung gewünschtes Zielschema
 - Jedoch einige Mappings nicht erkannt \Rightarrow Folge: Delete/Add anstatt Update
 - \Rightarrow fehlende Mappings durch Benutzer hinzufügen
 - CodeXMapper: grafisch/textuell
 - CodeX: textuell

CodeXMapper vs. CodeX

- CodeXMapper bietet Alternativen zu CodeX, z.B. move-Befehl anstatt Namensänderung
- Feinheiten in der XPath-Adressierung
 - add elementref 'ab' as first into '[...]/node()/. ' ;
 - add elementref 'ab' in '[...]/node()/. ' ;

- Mögliche Kombinationen zwischen CodeXMapper und CodeX
 - CodeXMapper Mappings und ElaX-Ausdrücke auf Schemaebene erzeugen und im EMX von CodeX anzeigen lassen
 - Quell- und Ziel-EMX zur Verfügung stellen ⇒ Ableitung XML-Schemata ⇒ CodeXMapper
 - Logik des CodeXMapper auf konzeptionelle Ebene übertragen

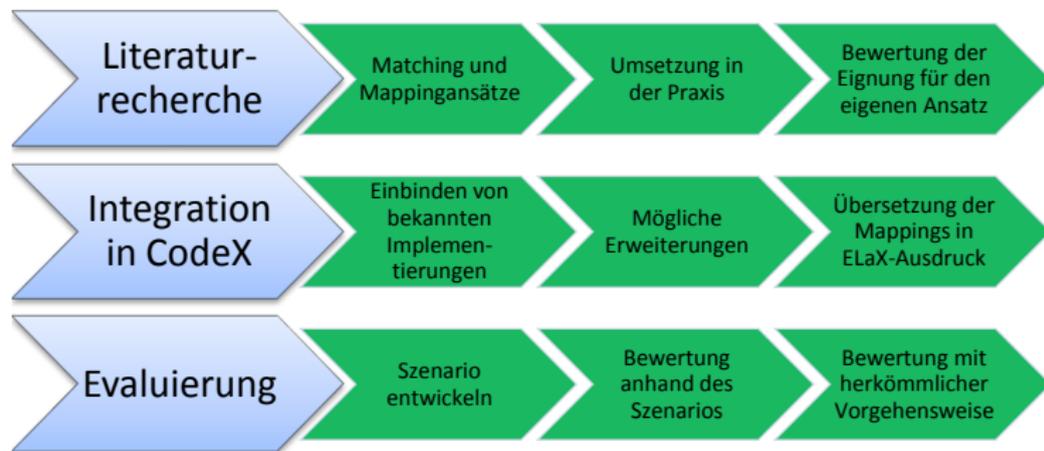


Figure: Zusammenfassung: Umsetzug der Ziele

- Automatisches Matching und Mapping nicht trivial
- Zusammenstellung der richtigen Matching und Mappingverfahren ist abhängig vom Anwendungsfall und bedarf meist Expertenerfahrung
- Keine allumfassende Lösung \Rightarrow Trend: Integration Vorteile mehrerer Tools
- Erweiterungsmöglichkeiten von CodeXMapper
 - Automatisches Mapping beim ELaX-Vergleich
 - Verwendung von Ontologien und/oder Synonymwörterbücher
 - Wiederverwendbare Mappingergebnisse

Gibt es noch Fragen?

