

Schemaevolution und Adaption von XML-Dokumenten und XQuery-Anfragen

Tobias Tiedt

```
<add  
select="//xs:element"  
content="maxOccurs=5"  
>
```

Übersicht

- Ziel der Diplomarbeit
- Übersicht

Evolution

XSEL

Implementierung

Zusammenfassung / Ausblick

Übersicht

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Ziel der Diplomarbeit

Übersicht

● Ziel der Diplomarbeit

● Übersicht

Evolution

XSEL

Implementierung

Zusammenfassung / Ausblick

- Entwurf einer Änderungssprache
- ↔ XSEL
- allg: Umsetzung der Evolution
- ↔ Umsetzung der Schema-Änderungen
- ↔ Anpassung der XML-Dokumente
- ↔ Anpassung von XQuery-Anfragen

```
<add  
select="//xs:element"  
content="maxOccurs=5"  
>
```

Übersicht

Übersicht

● Ziel der Diplomarbeit

● **Übersicht**

Evolution

XSEL

Implementierung

Zusammenfassung / Ausblick

- Evolution
 - ◆ Erläuterung und Gründe
 - ◆ Ablauf
- XSEL (**X**ML **S**chema **E**volution **L**anguage)
 - ◆ Operationen
 - ◆ Constraints
 - ◆ inkrementelle Validierung
- Implementierung
 - ◆ Übersicht
 - ◆ Beispiel
- Zusammenfassung / Ausblick

```
<add  
select="//xs:element"  
content="maxOccurs=5"  
>
```

Übersicht

Evolution

- Erklärung
- Gründe für Evolution im XML-Bereich
- Evolutionsschritt

XSEL

Implementierung

Zusammenfassung / Ausblick

Evolution

```
<add  
select="//xs:element"  
content="maxOccurs=5"  
>
```

Erklärung

Übersicht

Evolution

● Erklärung

- Gründe für Evolution im XML-Bereich
- Evolutionsschritt

XSEL

Implementierung

Zusammenfassung / Ausblick

- Notwendigkeit von Änderungen auf Schema-Ebene
- Gründe:
 - ◆ veränderte Anforderungen durch Nutzer
 - ◆ veränderte Anforderungen über die Zeit
 - ◆ ungenügender / fehlerhafter Entwurf des Schemas
- Evolutionsschritt:
 - ◆ Änderung des Schemas
 - ◆ Adaption der Daten
 - ◆ Anpassung der Anfragen
- fortwährender Prozess
- Folge von Evolutionsschritten

```
<add  
select="//xs:element"  
content="maxOccurs=5"  
>
```

Gründe für Evolution im XML-Bereich

Übersicht

Evolution

● Erklärung

● Gründe für Evolution im
XML-Bereich

● Evolutionsschritt

XSEL

Implementierung

Zusammenfassung / Ausblick

- XML als Metasprache
- Einsatz von XML als Datenaustauschformat
- Daten zunehmend im XML-Format gespeichert
- Gründe:
 - ◆ Abbildung von XML-Daten auf relationale Datenbanken ist ungenügend
 - ◆ Abbildung von XML-Daten auf objektorientierte Datenbanken besser
 - Verlust der Vorteile und dem Nutzen von XML
- ↪ gleiche Anforderungen wie bei Datenbanksystemen

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Evolutionsschritt

Übersicht

Evolution

- Erklärung
- Gründe für Evolution im XML-Bereich
- Evolutionsschritt

XSEL

Implementierung

Zusammenfassung / Ausblick

- Änderung der Schemas
- Anpassung der Dokumente
 - ◆ Dokumente nicht mehr valide bezüglich neuem Schema
 - ◆ Schemaänderung muss sich in Dokumenten widerspiegeln
 - ◆ minimaler Informationsverlust
- Anpassung von Anfragen (z.B. XQuery-Anfragen)
 - ◆ Anfragen nicht mehr ausführbar
 - ◆ Anfragen liefern falsche Ergebnisse


```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Übersicht

Evolution

XSEL

- Überblick
- Operationen (1)
- Operationen (2)
- Constraints (1)
- Constraints (2)
- inkrementelle Validierung (1)
- inkrementelle Validierung (2)
- Generierung von
Standardwerten
- Generierung für komplexe
Typen

Implementierung

Zusammenfassung / Ausblick

XSEL

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Überblick

Übersicht

Evolution

XSEL

● Überblick

● Operationen (1)

● Operationen (2)

● Constraints (1)

● Constraints (2)

● inkrementelle Validierung (1)

● inkrementelle Validierung (2)

● Generierung von
Standardwerten

● Generierung für komplexe
Typen

Implementierung

Zusammenfassung / Ausblick

- Änderungssprache
 - ◆ 10 Änderungsoperationen auf XML-Schema
- Sprachkonstrukte in XML-Syntax
- keine speziellen Konstruktoren
 - ◆ jedes XML-Dokument änderbar
- Abbildung Schema-Dokument
- Constraints
- inkrementelle Validierung

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Operationen (1)

Übersicht

Evolution

XSEL

● Überblick

● **Operationen (1)**

● Operationen (2)

● Constraints (1)

● Constraints (2)

● inkrementelle Validierung (1)

● inkrementelle Validierung (2)

● Generierung von

Standardwerten

● Generierung für komplexe
Typen

Implementierung

Zusammenfassung / Ausblick

- **add**: fügt an einen Knoten neues Element oder Attribut (allg: Knoten) hinzu
- **insert_before**: fügt Knoten vor eine bestimmte Position ein
- **insert_after**: fügt Knoten hinter eine bestimmte Position ein
- **move**: Verschieben von Dokumentteilen im Dokument
- **move_before**: Verschieben von Dokumentteilen vor eine bestimmte Position
- **move_after**: Verschieben von Dokumentteilen hinter eine bestimmte Position

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Operationen (2)

Übersicht

Evolution

XSEL

- Überblick
- Operationen (1)
- **Operationen (2)**
- Constraints (1)
- Constraints (2)
- inkrementelle Validierung (1)
- inkrementelle Validierung (2)
- Generierung von Standardwerten
- Generierung für komplexe Typen

Implementierung

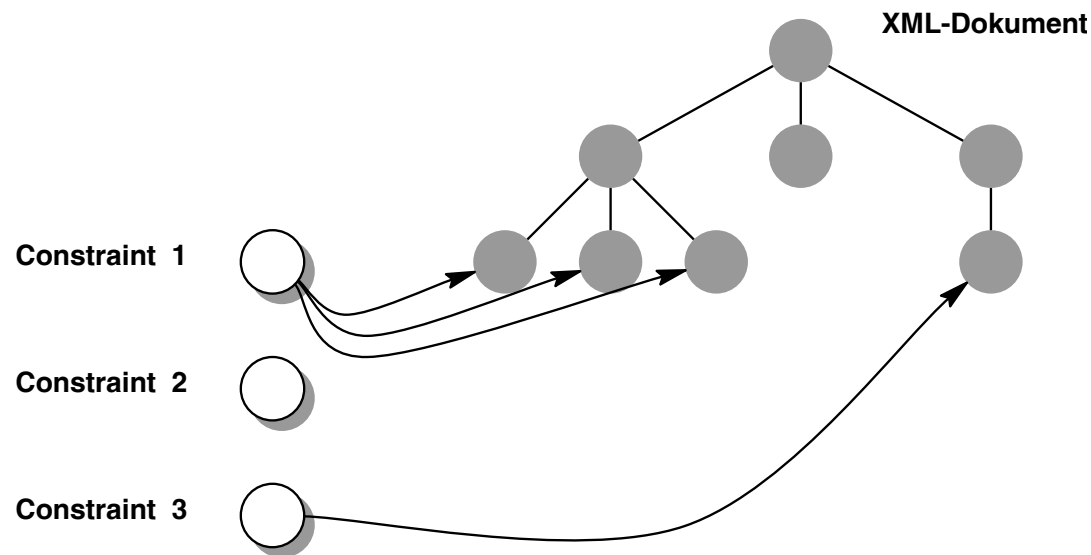
Zusammenfassung / Ausblick

- **delete**: Löschen von Knoten
- **replace**: Ersetzen von Knoten
- **rename**: Umbenennen von Knoten
- **change**: Änderung des Knotenwerts
- weitere Konzepte: Gruppierung (Queries) und Transaktionen

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Constraints (1)

- Bedingung für einen Knoten
- für jeden deklarierten Knoten ein Constraint
- beinhaltet Deklarations- und Typinformationen
- definiert Abbildung vom Schema ins Dokument
- Einsatz bei der inkrementellen Validierung



Übersicht

Evolution

XSEL

- Überblick
- Operationen (1)
- Operationen (2)
- Constraints (1)
- Constraints (2)
- inkrementelle Validierung (1)
- inkrementelle Validierung (2)
- Generierung von Standardwerten
- Generierung für komplexe Typen

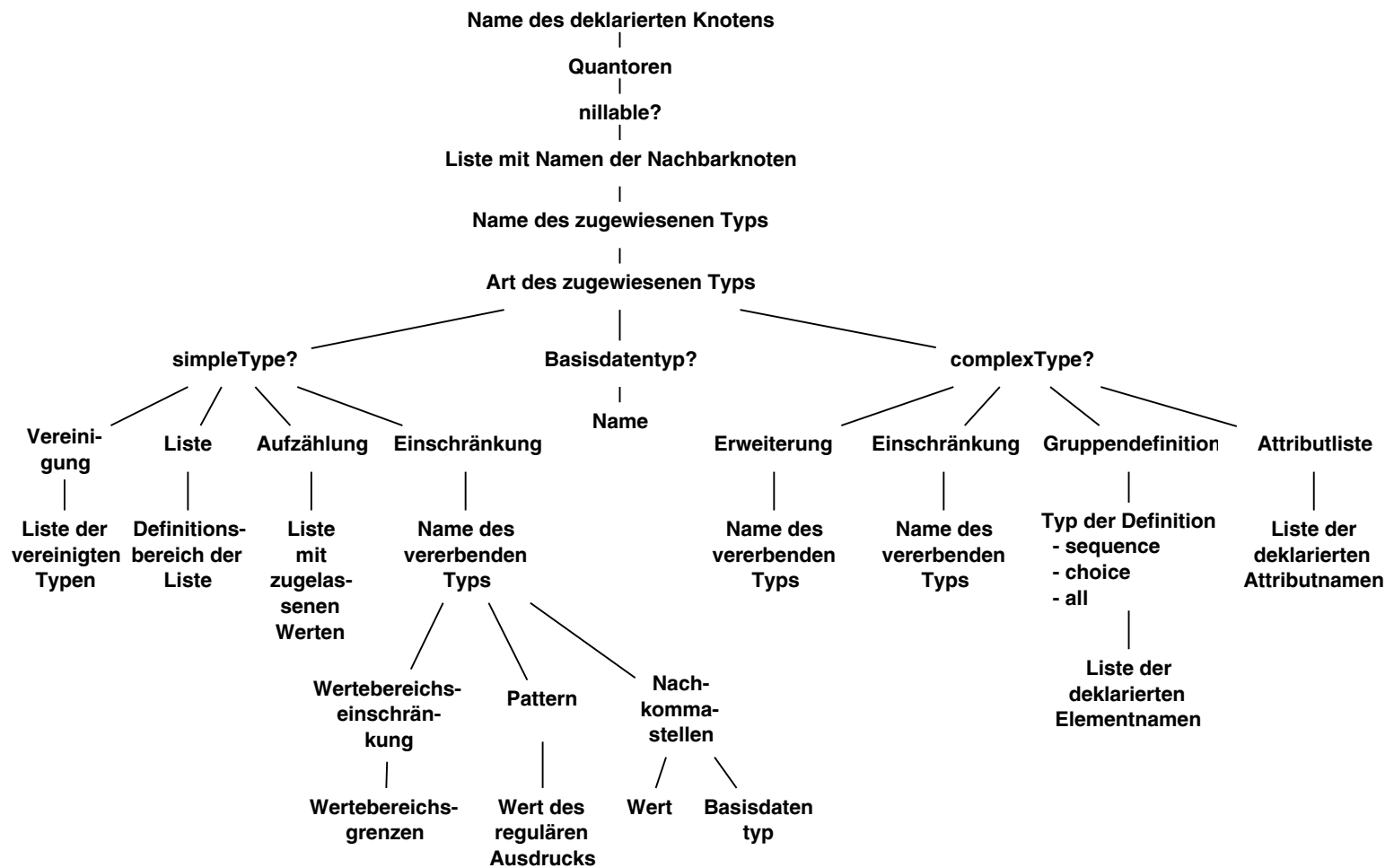
Implementierung

Zusammenfassung / Ausblick

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Constraints (2)

- Übersicht
- Evolution
- XSEL
 - Überblick
 - Operationen (1)
 - Operationen (2)
 - Constraints (1)
 - **Constraints (2)**
 - inkrementelle Validierung (1)
 - inkrementelle Validierung (2)
 - Generierung von Standardwerten
 - Generierung für komplexe Typen
- Implementierung
- Zusammenfassung / Ausblick



```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

inkrementelle Validierung (1)

Übersicht

Evolution

XSEL

- Überblick
- Operationen (1)
- Operationen (2)
- Constraints (1)
- Constraints (2)
- inkrementelle Validierung (1)
- inkrementelle Validierung (2)
- Generierung von Standardwerten
- Generierung für komplexe Typen

Implementierung

Zusammenfassung / Ausblick

- Validierung von evtl. abzuändernden Dokumentteilen
- Nutzung der Constraints und der Schema-Dokument-Abbildung
- Überprüfung der Eigenschaften des Constraints
 - ◆ alle Eigenschaften sind erfüllt
 - ◆ \hookrightarrow Knoten ist valide
 - ◆ manche Eigenschaften sind nicht erfüllt
 - ◆ \hookrightarrow Knoten ist nicht valide
- Ableitung der durchzuführenden Dokumentadaption
 - ◆ welche Constrainteigenschaften sind nicht erfüllt?
 - ◆ \hookrightarrow Einfügen, Löschen, Ändern von Knoten

```

<add
select="//xs:element"
content="maxOccurs=5"
/>

```

inkrementelle Validierung (2)

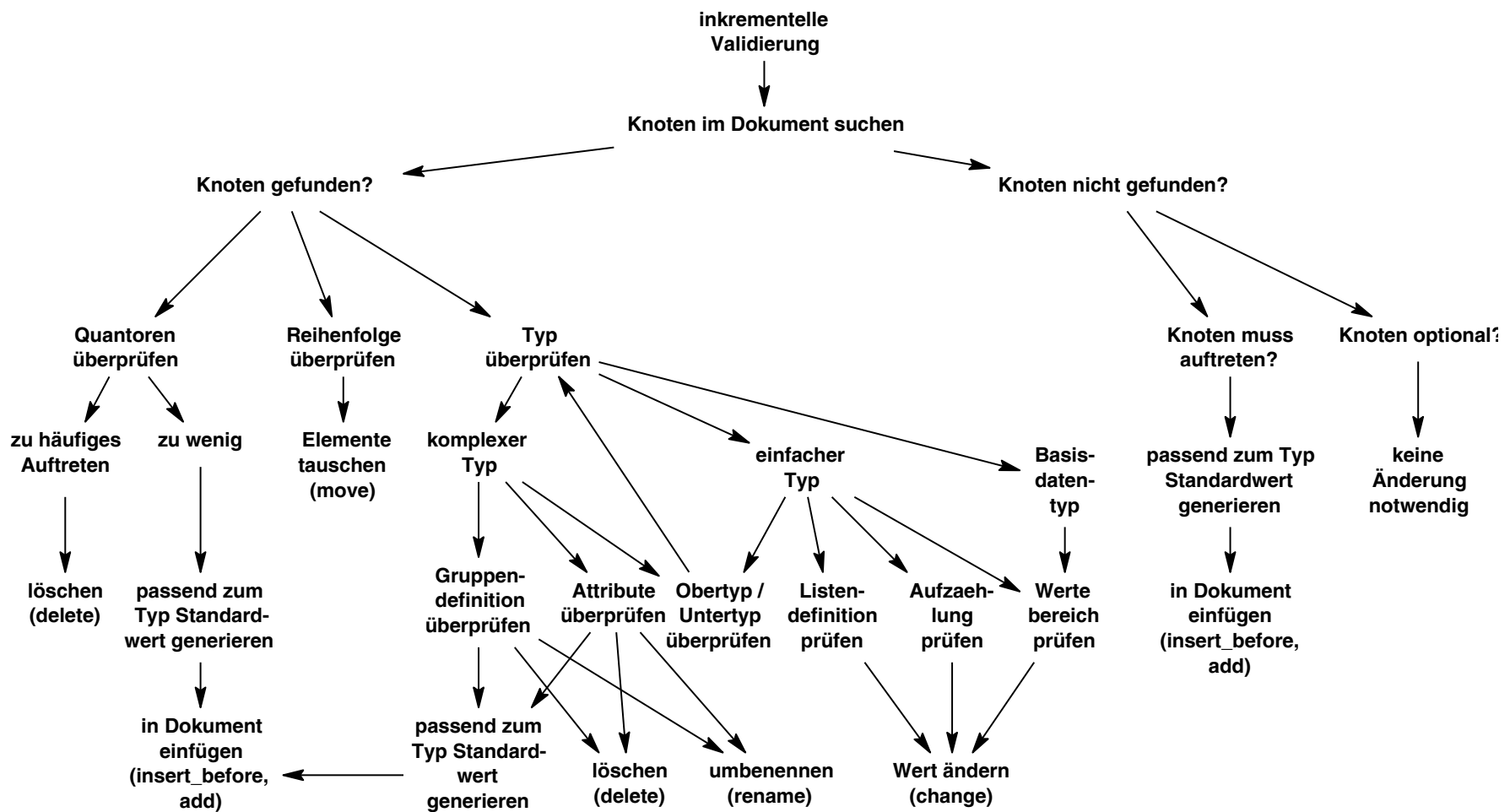
- Übersicht

- Evolution

- XSEL
 - Überblick
 - Operationen (1)
 - Operationen (2)
 - Constraints (1)
 - Constraints (2)
 - inkrementelle Validierung (1)
 - **inkrementelle Validierung (2)**
 - Generierung von Standardwerten
 - Generierung für komplexe Typen

- Implementierung

- Zusammenfassung / Ausblick




```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Generierung von Standardwerten

Übersicht

Evolution

XSEL

- Überblick
- Operationen (1)
- Operationen (2)
- Constraints (1)
- Constraints (2)
- inkrementelle Validierung (1)
- inkrementelle Validierung (2)
- **Generierung von Standardwerten**
- Generierung für komplexe Typen

Implementierung

Zusammenfassung / Ausblick

- **baseType**
 - ◆ Festlegung eines Wertes für jeden Typ
 - ◆ z.B.: long 0, double 0.0, string "."
- **simpleType**
 - ◆ Analyse des regulären Ausdrucks
 - ◆ \hookrightarrow Substitution der Klassen durch Werte
 - ◆ \hookrightarrow z.B.: $\backslash d "0"$, $[a-z] "a"$, $(aaa | bbb) "aaa"$
 - ◆ Wertebereichsgrenze als Wert
 - ◆ erster Wert der Aufzählung als Standardwert
- **complexType**
 - ◆ Generierung von XML-Literalen
 - ◆ Abstieg in der Hierarchie des Schemas
 - ◆ deklarierte Blattknoten haben Standardtyp oder einfachen Typ

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Generierung für komplexe Typen

Übersicht

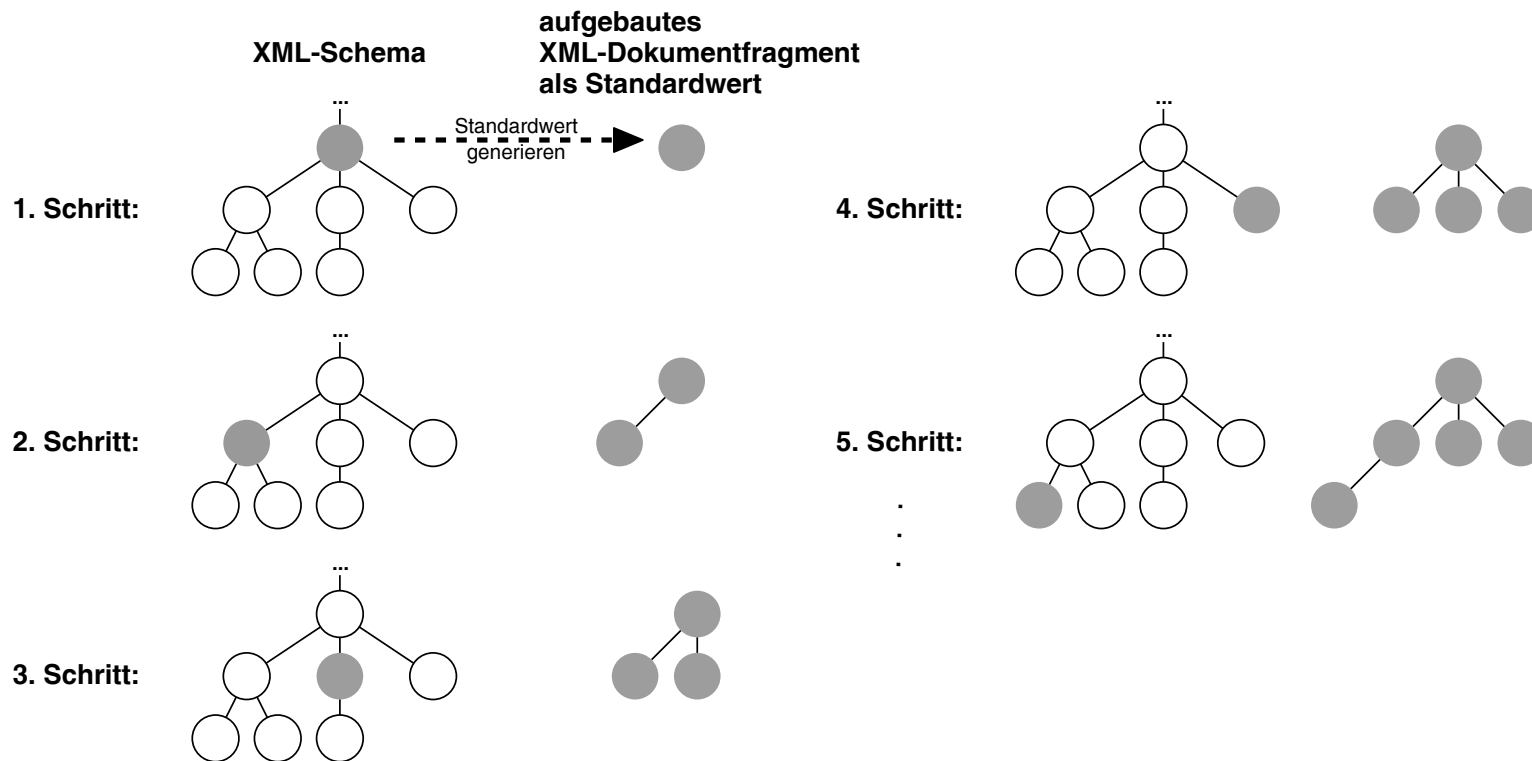
Evolution

XSEL

- Überblick
- Operationen (1)
- Operationen (2)
- Constraints (1)
- Constraints (2)
- inkrementelle Validierung (1)
- inkrementelle Validierung (2)
- Generierung von Standardwerten
- Generierung für komplexe Typen

Implementierung

Zusammenfassung / Ausblick



```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

[Übersicht](#)

[Evolution](#)

[XSEL](#)

Implementierung

- Übersicht
- Framework
- Ablauf eines
Evolutionsschrittes (1)
- Ablauf eines
Evolutionsschrittes (2)
- Ablauf eines
Evolutionsschrittes (3)
- Ablauf eines
Evolutionsschrittes (4)
- Ablauf eines
Evolutionsschrittes (5)

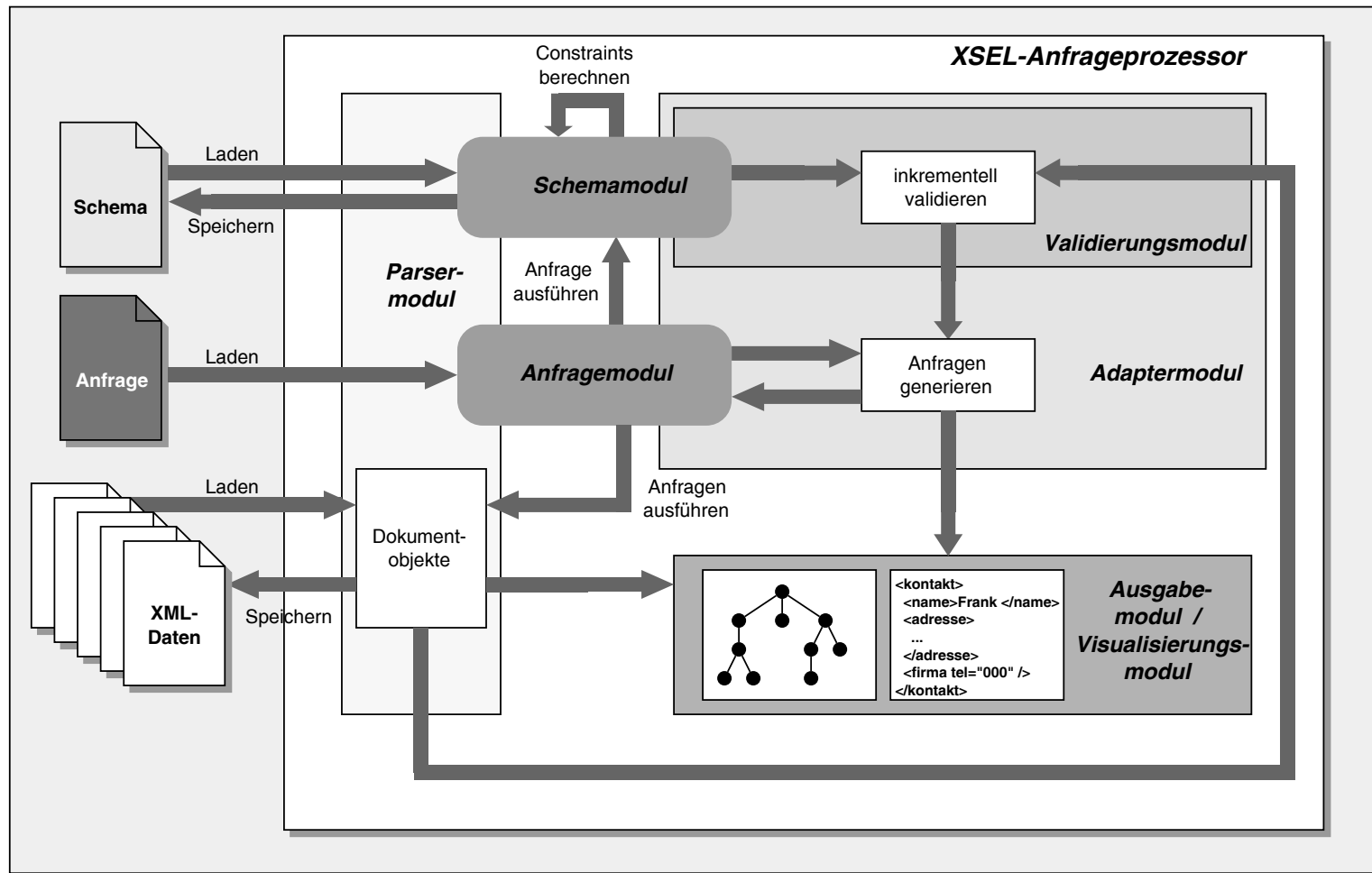
[Zusammenfassung / Ausblick](#)

Implementierung

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Übersicht

■ Implementierung: Java SDK 1.4.2 mit Xerces 2.6.2



Übersicht

Evolution

XSEL

Implementierung

● Übersicht

● Framework

● Ablauf eines

Evolutionsschrittes (1)

● Ablauf eines

Evolutionsschrittes (2)

● Ablauf eines

Evolutionsschrittes (3)

● Ablauf eines

Evolutionsschrittes (4)

● Ablauf eines

Evolutionsschrittes (5)

Zusammenfassung / Ausblick

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Framework

Übersicht

Evolution

XSEL

Implementierung

● Übersicht

● **Framework**

● Ablauf eines

Evolutionsschrittes (1)

● Ablauf eines

Evolutionsschrittes (2)

● Ablauf eines

Evolutionsschrittes (3)

● Ablauf eines

Evolutionsschrittes (4)

● Ablauf eines

Evolutionsschrittes (5)

Zusammenfassung / Ausblick

- **Parsermodul:** beinhaltet Parser für XML-Dok., Schema, Anfragen und XQuery-Anfragen
 - ◆ **Schemamodul:** Schema-Klasse beinhaltet zusätzlich Constraint- und TypeInfo-Objekte
 - ◆ **Querymodul:** bildet Anfrage auf DOM-Operationen ab und wendet diese an
- **Validierungsmodul:** realisiert inkrementelle Validierung
- **Adaptermodul:** generiert Anfragen zur Dokumentadaption
 - ◆ Anwendung der gen. Anfragen durch **Querymodul**

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Ablauf eines Evolutionsschrittes (1)

Übersicht

Evolution

XSEL

Implementierung

● Übersicht

● Framework

● Ablauf eines
Evolutionsschrittes (1)

● Ablauf eines

Evolutionsschrittes (2)

● Ablauf eines

Evolutionsschrittes (3)

● Ablauf eines

Evolutionsschrittes (4)

● Ablauf eines

Evolutionsschrittes (5)

Zusammenfassung / Ausblick

- Beispiel: Kontaktdaten für Personen
- Fehlen des Namens bei Firma \Rightarrow ungenügender Entwurf

```
<xs:schema>
  <xs:complexType name="firma">
    <xs:sequence>
      <xs:element name="Adresse" type="adresse" minOccurs="0"/>
      <xs:element name="Telefon" type="telefon"/>
      <xs:element name="e-mail" type="email"/>
      <xs:element name="Webseite" type="webseite" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="kontakt">
    <xs:sequence>
      <xs:element name="Nachname" type="xs:string"/>
      <xs:element name="Vorname" type="xs:string"/>
      ...
      <xs:element name="Firma" type="firma"/>
    </xs:sequence>
  </xs:complexType>
  ...
  <xs:element name="Kontakt" type="kontakt"/>
</xs:schema>
```

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Ablauf eines Evolutionsschrittes (2)

Übersicht

Evolution

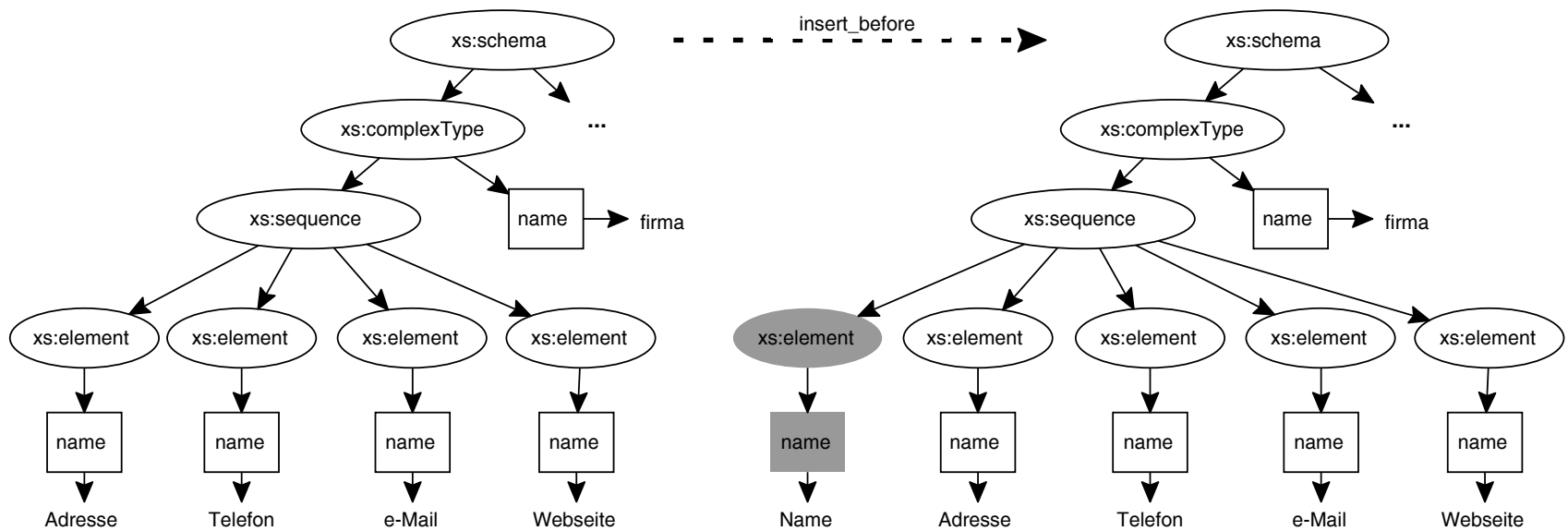
XSEL

Implementierung

- Übersicht
- Framework
- Ablauf eines Evolutionsschrittes (1)
- Ablauf eines Evolutionsschrittes (2)
- Ablauf eines Evolutionsschrittes (3)
- Ablauf eines Evolutionsschrittes (4)
- Ablauf eines Evolutionsschrittes (5)

Zusammenfassung / Ausblick

```
<insert_before
select="//xs:schema/xs:complexType[@name='firma']/xs:sequence"
before="xs:element[@name='Adresse']"
content="&lt;xs:element
name=&quot;Name&quot;
type=&quot;xs:string&quot;/&gt;"
/>
```



```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Ablauf eines Evolutionsschrittes (3)

Übersicht

Evolution

XSEL

Implementierung

- Übersicht
- Framework
- Ablauf eines Evolutionsschrittes (1)
- Ablauf eines Evolutionsschrittes (2)
- **Ablauf eines Evolutionsschrittes (3)**
- Ablauf eines Evolutionsschrittes (4)
- Ablauf eines Evolutionsschrittes (5)

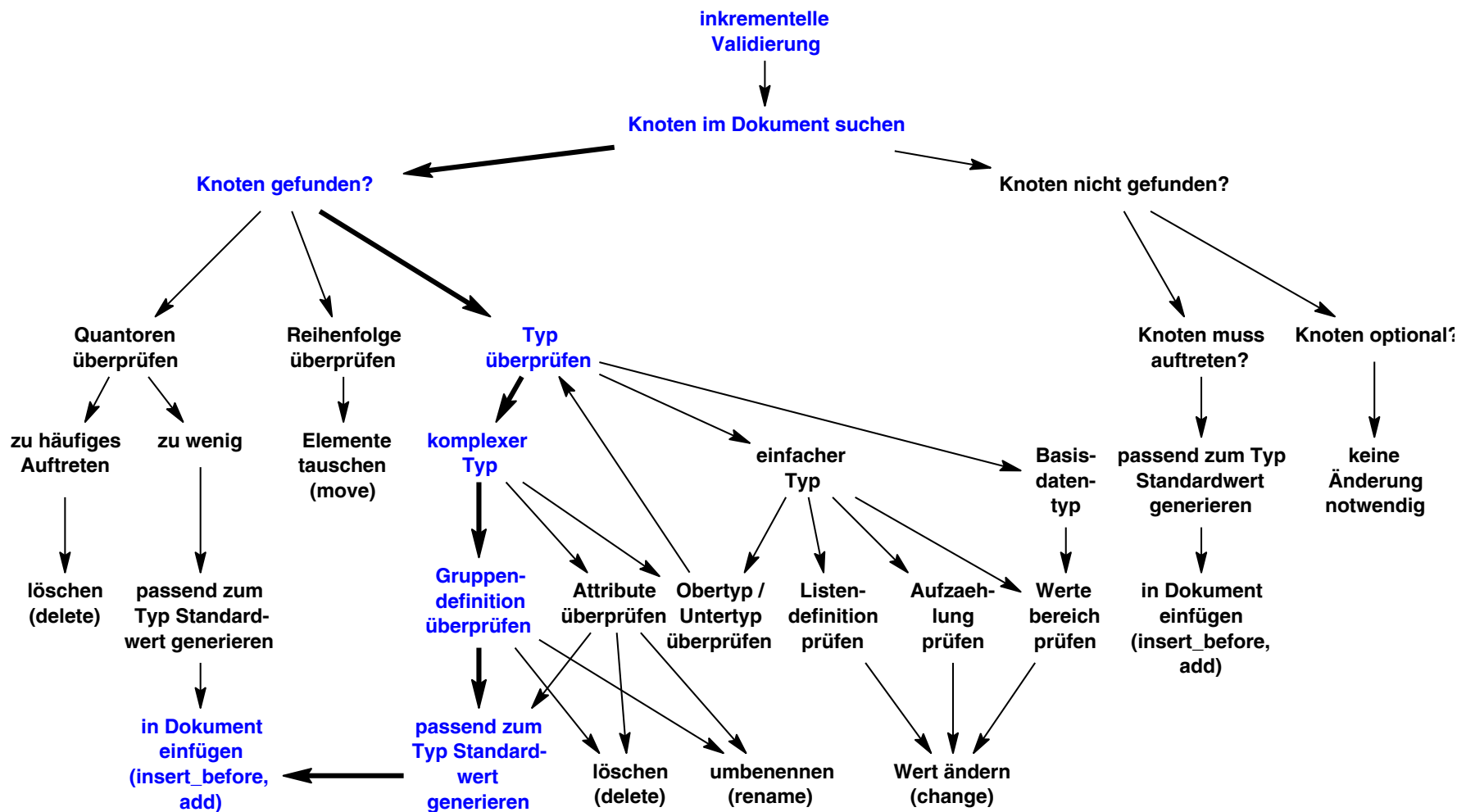
Zusammenfassung / Ausblick

- Anwendung der Anfrage auf das Schema
- Neuberechnung der Constraints im **Schemamodul**
 - ◆ Generierung von XPath-Ausdrücken zur Selektion in den Instanzdokumenten
 - ◆ Aufstieg in der Hierarchie des Schemas
 - ◆ \hookrightarrow /Kontakt/Firma
- **Adaptermodul** lädt Dokumente
 - ◆ inkrementelle Validierung durch **Validierungsmodul**
 - ◆ \hookrightarrow Überprüfung der Constraint-Eigenschaften des Elementes Firma


```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Ablauf eines Evolutionsschrittes (4)

- Übersicht
- Evolution
- XSEL
- Implementierung
 - Übersicht
 - Framework
 - Ablauf eines Evolutionsschrittes (1)
 - Ablauf eines Evolutionsschrittes (2)
 - Ablauf eines Evolutionsschrittes (3)
 - **Ablauf eines Evolutionsschrittes (4)**
 - Ablauf eines Evolutionsschrittes (5)
- Zusammenfassung / Ausblick



```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Ablauf eines Evolutionsschrittes (5)

Übersicht

Evolution

XSEL

Implementierung

- Übersicht
- Framework
- Ablauf eines Evolutionsschrittes (1)
- Ablauf eines Evolutionsschrittes (2)
- Ablauf eines Evolutionsschrittes (3)
- Ablauf eines Evolutionsschrittes (4)
- Ablauf eines Evolutionsschrittes (5)

Zusammenfassung / Ausblick

- **Adaptermodul** generiert Standardwert
- **Adaptermodul** generiert Anfrage(n) für Adaption und führt diese aus

```
<insert_before
  select="/Kontakt/Firma" before="Adresse"
  content="&lt;Name&gt;.&lt;/Name&gt;" />
```

- **Anpassung von XQuery-Anfragen**

```
<Person>{
  LET $a IN /Kontakt/Firma/child::*[1],
      $n IN /Kontakt/Nachname
  RETURN
    <Name>{$n/text()}</Name>
    <Firmenkontakt>{$a/text()}</Firmenkontakt>
}</Person>
```

```
<add  
select="//xs:element"  
content="maxOccurs=5"  
>
```

Übersicht

Evolution

XSEL

Implementierung

Zusammenfassung / Ausblick

- Zusammenfassung
- ?

Zusammenfassung / Ausblick

```
<add
select="//xs:element"
content="maxOccurs=5"
/>
```

Zusammenfassung

Übersicht

Evolution

XSEL

Implementierung

Zusammenfassung / Ausblick

● Zusammenfassung

● ?

- Definition von Änderungsoperationen
- Definition Schema-Dokument-Abbildung
- inkrementelle Validierung
- Umsetzung der Änderungen auf Schema-Ebene
- automatische Adaption der Dokumente
- manuelle Adaption der XQuery-Anfragen
- Erweiterungsmöglichkeiten:
 - ◆ vollständige Umsetzung eines Transaktionskonzeptes
 - ◆ vollständige Umsetzung des Typkonzeptes von XML-Schemas

```
<add  
select="//xs:element"  
content="maxOccurs=5"  
>
```



Fragen?

Übersicht

Evolution

XSEL

Implementierung

Zusammenfassung / Ausblick

● Zusammenfassung

● ?