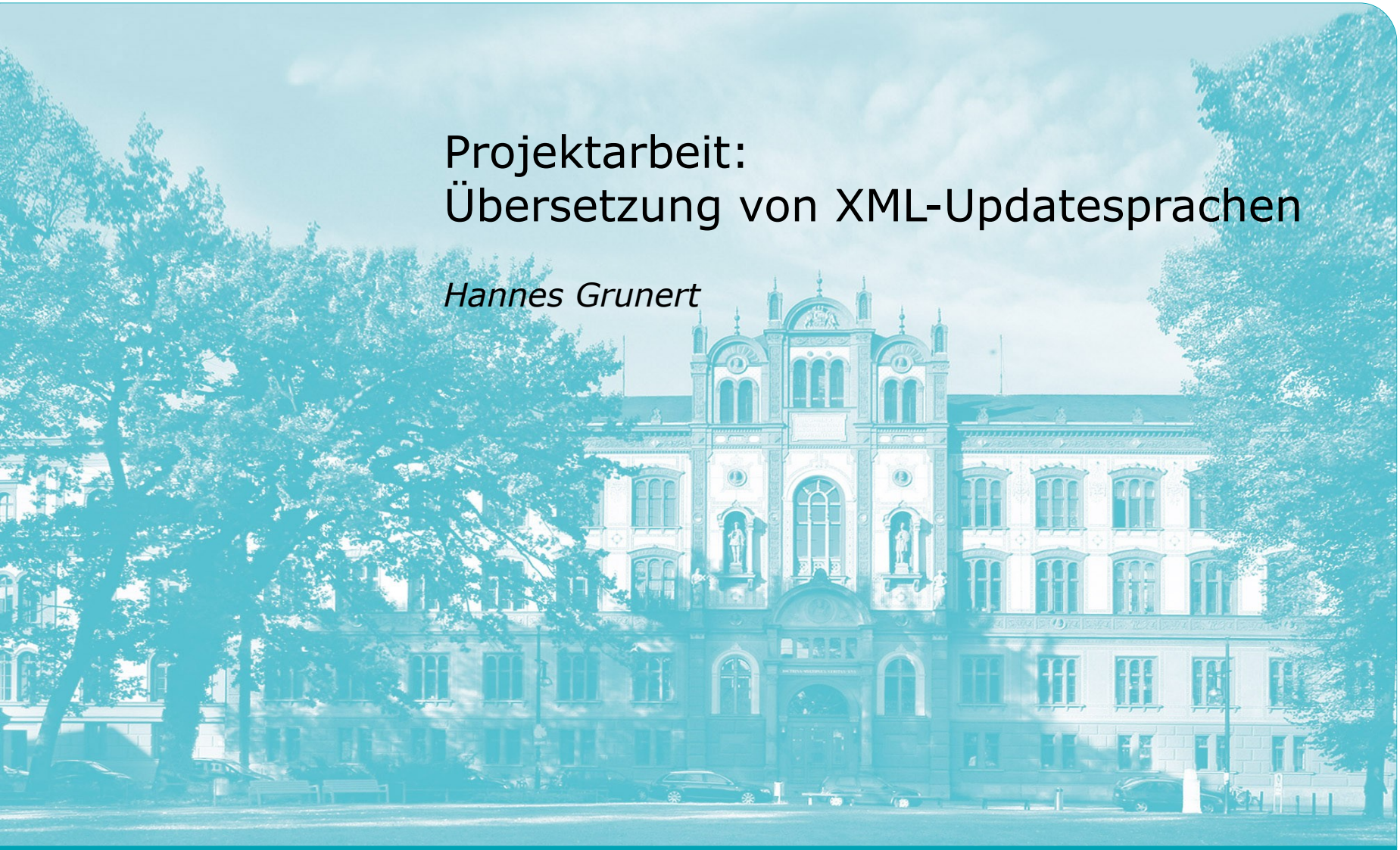
Universität Rostock

Traditio et Innovatio



Projektarbeit: Übersetzung von XML-Updatesprachen

Hannes Grunert

Inhalt

Inhalt

Aufgabenstellung

- Übersetzung von XSEL[1]-Ausdrücken in XML-Updatesprache
  - Vergleich von XUpdate und XSLT
  - Algorithmus zur Übersetzung in die gewählte Sprache
  - Einbindung in CodeX[2]
  - Lauffähigkeit an einem Beispiel zeigen

1 Xml-Schema-Evolution Language
2 COnceptual Design and Evolution for Xml-schema

XSEL

- Diplomarbeit von Tobias Tiedt
- Orientiert sich am XUpdate-Sprachvorschlag
- Operationen:
  - **add, insert_before, insert_after**: Hinzufügen neuer Elemente und Attribute
  - **delete**: Löschen von Elementen und Attributen
  - **rename**: Umbenennen von Elementen und Attributen
  - **replace, change**: Ersetzen von Elementen und Attributen
  - **move, move_before, move_after**: Verschieben von Elementen

XSEL - Beispiel

```
<add
  select="//film[@film-id='42']/filmdaten"
  content="&lt;xs:attribute
           name=&quot;originalsprache&quot;
           &gt;Deutsch&lt;/xs:attribute&gt;"
/>


<Operationsname Attribut=Wert … />
```

Inhalt

XUpdate

- Ursprung: XML:DB-Initative
- Working Draft seit 2000
- Update von XML-Dokument-Kollektionen
  - wenige Operationen, die alle Änderungen überdecken
  - leicht verständlich
  - eignet sich auch für Evolution, da XSD ⊂ XML
- XPath zur Selektion von Knoten
- Beispiel:

```
<xupdate:append select="//film[@film-id='42']/filmdaten">
   <xupdate:attribute name="originalsprache">Deutsch</xupdate:attribute>
</xupdate:append>
```

XUpdate - Operationen

- Namespace: http://www.xmldb.org/xupdate
- **insert-before, insert-after, append:** neue Elemente und Attribute einfügen
- **update**: Verändern von Elementen und Attributen
- **remove**: Elemente und Attribute löschen
- **rename**: Umbenennen von Elementen und Attributen
- **variable**: Binden eines Wertes an eine Variable
- **value-of**: Zum Auswerten von Variablen und XPath-Ausdrücken
- **if**: Für bedingte Anweisungen (wird im Working Draft nicht erklärt)

XUpdate - Nachteile

- keine Recommendation beim W3C
- Arbeit am Working Draft scheinbar eingestellt
- wenige Implementationen
  - Lexus
  - Xindice
  - Jaxup
- Implementationen nicht mehr lauffähig
  - Änderungen in abhängigen Bibliotheken
  - Cast-Fehler, fehlende Funktionen
  - keine Namespace-Unterstützung → keine XML-Schema-Elemente möglich
- Abhilfe: alternative Implementierung (Vorstellung später)

XSLT

- E**x**tensible **S**tylesheet **L**anguage **T**ransformations
- W3C-Recommendation
  - Version 1.0: November 1999
  - Version 2.0: Januar 2007
- eigentlich zur Formatierung/Anpassung von XML-Dokumenten an verschiedene Ausgabemedien/-formate
- Template-Rules passen Teile eines Dokumentes an
  - Knotenauswahl erfolgt über Pattern-Matching
    - Selektion über XPath
  - Regeln lassen sich so spezifizieren, dass sie Updates auf XML-Dokumenten ausführen
- Viele ausgereifte Implementierungen (Apache Xalan)

XSLT – Beispiel (Einfügen eines Attributes)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http:/www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" intent="yes" encoding="UTF-8">
    <xsl:template match="//film[@film-id='42']/filmdaten">
        <xsl:copy>
            <xsl:apply-templates select="attribute::*" />
            <xsl:attribute name="originalsprache">Deutsch</xsl:attribute>
            <xsl:apply-templates select="child::node()" />
        </xsl:copy>
    </xsl:template>
    <xsl:template match="@*|node()">
        <xsl:copy>
            <xsl:apply-templates select="attribute::* | child::node()">
        </xsl:copy>
    </xsl:template>
</xsl:stylesheet>
```

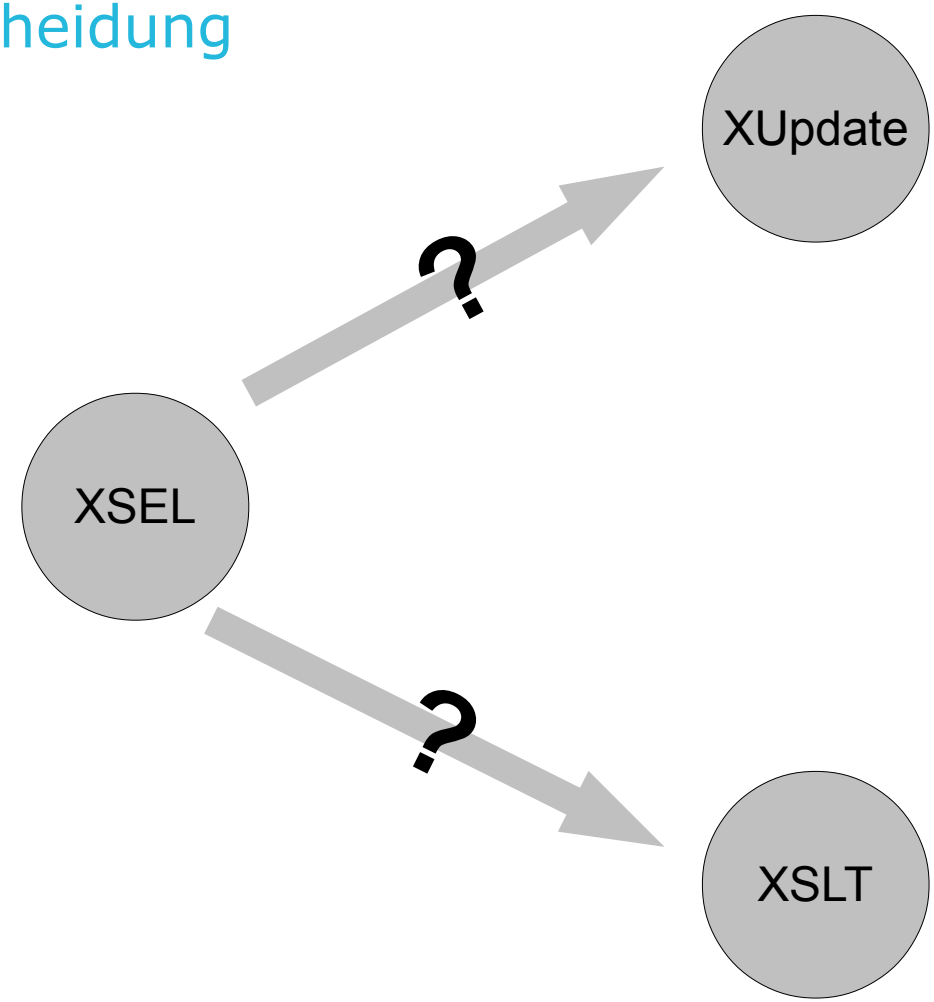
XSLT - Nachteile

- Knoten lassen sich maximal einmal verändern
  - Zwei verschiedene Updates auf einem Element lassen sich in einer Transformation nicht umsetzen
  - Updates müssen somit sequentiell durchgeführt werden
  - Transformationsaufwand steigt somit
- Template-Rules lang und unübersichtlich
  - Extremfall: Template-Rule für insert-before benötigt das 10-fache an LoC (Lines of Code) im Vergleich zu XUpdate
- Nicht veränderte Knoten benötigen extra Kopier-Regel, da sie sonst nicht mit übernommen werden
  - Transformationsaufwand steigt weiter
- Optimierte Prozessoren gleichen erhöhten Aufwand aus

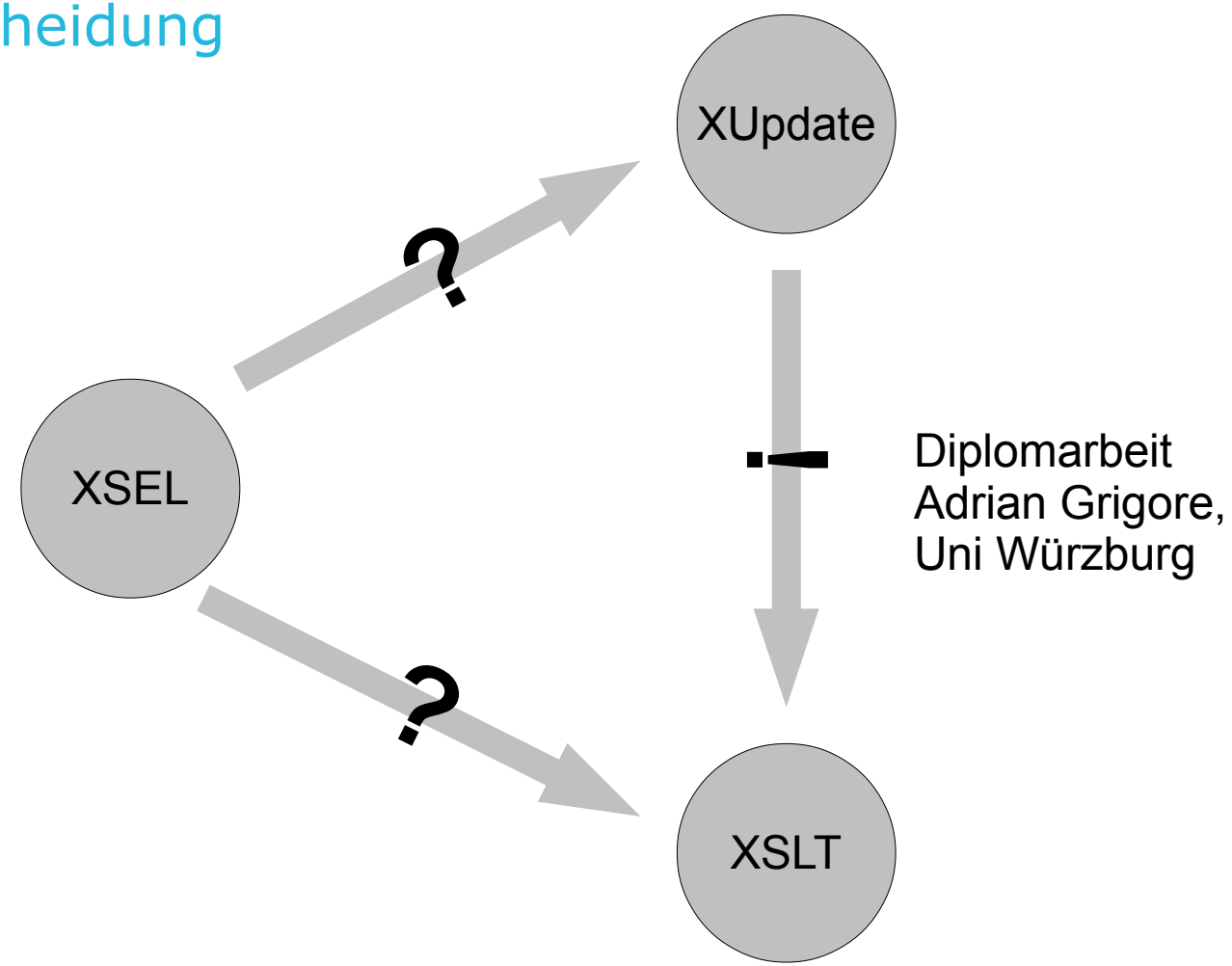
Vergleich

Kriterium	XUpdate	XSLT
Verständlichkeit	leicht, Quellcode übersichtlich	schwer, ca. 10-mal soviele LoC wie XUpdate
Standard	nein, Working Draft seit 2000	ja, Recommendation seit 1999
verfügbare Prozessoren	wenige, nicht lauffähig	viele, weit verbreitet
Funktionalität	alle Update- und Evolutionsschritte umsetzbar	alle Update- und Evolutionsschritte umsetzbar
Übersetzungs-aufwand	gering	hoch

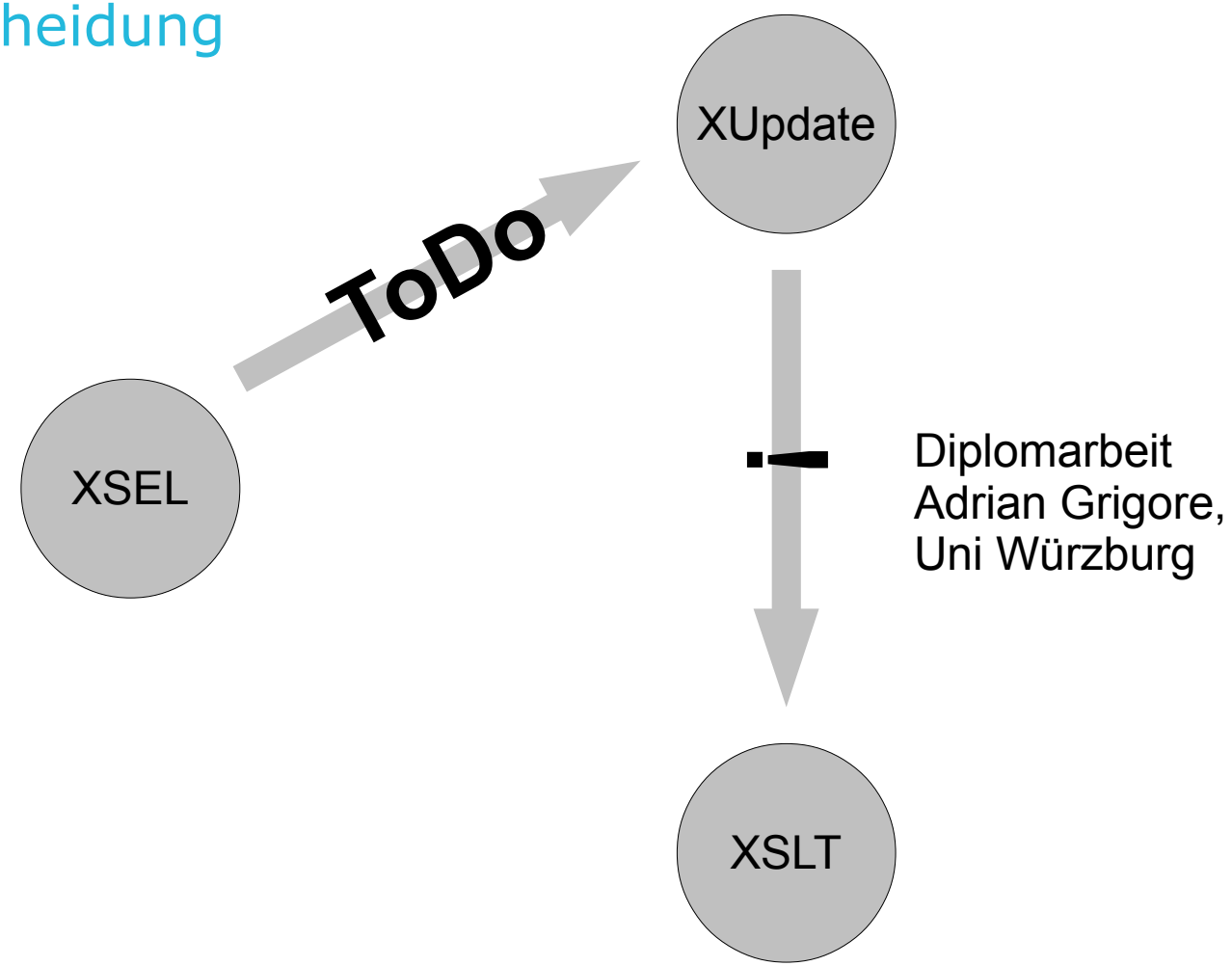
Entscheidung



Entscheidung



Entscheidung



Inhalt

XSEL → XUpdate

- Teilaufgaben der Übersetzung
  1) Mapping der Tagnamen
  2) Zusammensetzen des *select*-Attributwertes
  3) Konvertierung des content/value-Attributes in XUpdate-Elemente
  4) Zerlegung des move(_before/_after)-Operators

XSEL → XUpdate

- 1) Mapping der Tag-Namen:
  - add → xupdate:append
  - insert_before → xupdate:insert-before
  - insert_after → xupdate:insert-after
  - delete → xupdate:remove
  - rename → xupdate:rename
  - replace → xupdate:update
  - change → xupdate:update
  - move → ???
  - move_before → ???
  - move_after → ???

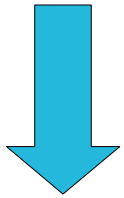
XSEL → XUpdate

- 2) Zusammensetzen des *select*-Attributwertes
  - notwendig nur für insert_before und insert_after
  - XSEL:
    - *select*-Attribut gibt Vaterknoten an, unter welchen neuer Knoten eingefügt werden soll
    - before/after-Attribut gibt Kindknoten an, vor/hinter dem der neue Knoten eingefügt werden soll
  - XUpdate:
    - select-Attribut gibt Knoten an unter/vor/hinter dem der neue Knoten eingefügt werden soll
    - Position abhängig vom jeweiligen Operator
  - → Zusammenfassen von select und before/after

XSEL → XUpdate

- 2) Zusammensetzen des *select*-Attributwertes
  - Beispiel:

```
<insert_before
select="/xs:schema/xs:complexType[@name='firma']/xs:sequence"
before="xs:element[@name='Telefon']"...
```



```
<xupdate:insert-before
select="/xs:schema/xs:complexType[@name='firma']/xs:sequence/
xs:element[@name='Telefon']">
```

  - select+"/"+before

XSEL → XUpdate

- 3) Konvertierung des content/value-Attributes in XUpdate-Elemente
  - XSEL: neue Elemente und Attribute liegen als content/value-Attributwerte inkl. Markup vor
  - XUpdate: eigene Operatoren zum Einfügen
    - xupdate:element
    - xupdate:attribute

XSEL → XUpdate

- 3) Konvertierung des content/value-Attributes in XUpdate-Elemente
  - Beispiel:

```
<add select="//xs:complexType[@name='kontakt']/xs:sequence"
content="&lt;xs:element name=&quot;Nachname&quot;
type=&quot;xs:string&quot;/&gt;"/>
```

```
<xupdate:append
select="//xs:complexType[@name='kontakt']/xs:sequence">
    <xupdate:element name="xs:element">
        <xupdate:attribute name="name">Nachname</xupdate:attribute>
        <xupdate:attribute name="type">xs:string</xupdate:attribute>
    </xupdate:element>
</xupdate:append>
```

XSEL → XUpdate

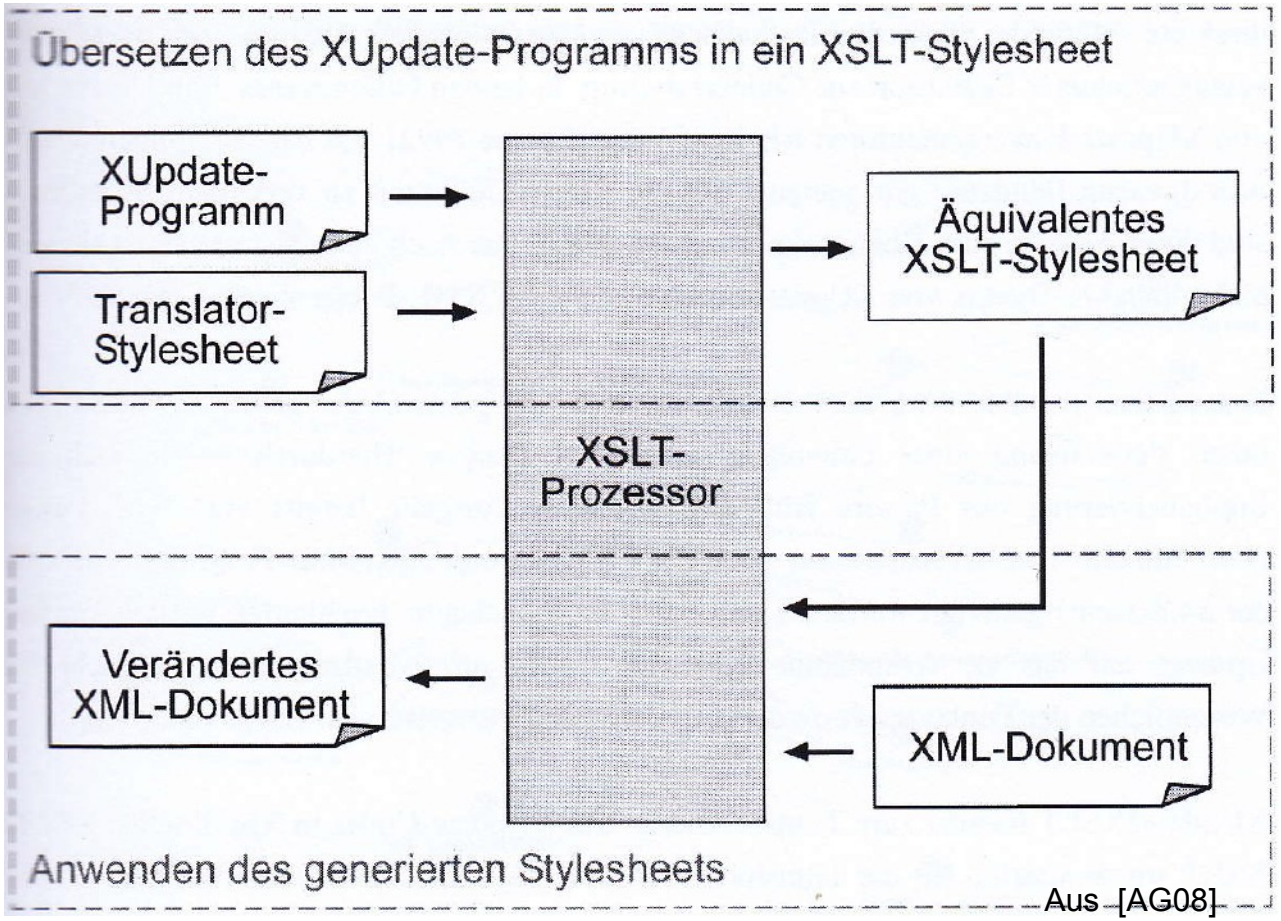
- 4) Zerlegung des move(_before/_after)-Operators
  - keine Entsprechung in XUpdate
  - Aufsplittung des Befehls
    - move = append + remove
    - move_before = insert-before + remove
    - move_after = insert-after + remove
  - Aber: bisher wurden mit append/insert nur neue Elemente und Attribute eingefügt
    - <xupdate:value-of select="..." />
    - select-Attribut zur Selektion eines Knoten (via XPath)
    - xupdate:value-of fügt innerhalb von append/insert den kompletten Teilbaum ein

XUpdate → XSLT

- Übersetzungsskript von Adrian Grigore (Diplomarbeit)
- XUpdate-Prozessor auf Basis von XSLT
  - Somit funktionsfähige XUpdate-Implementierung
- Nutzt zur Übersetzung ebenfalls XSLT
- Zum Update einfacher Dokumente
  - Namespace-Erweiterung
    - Dadurch Evolution von Schemata möglich
    - Sowie Dokumentkollektionen mit Namespace

Umsetzung in CodeX

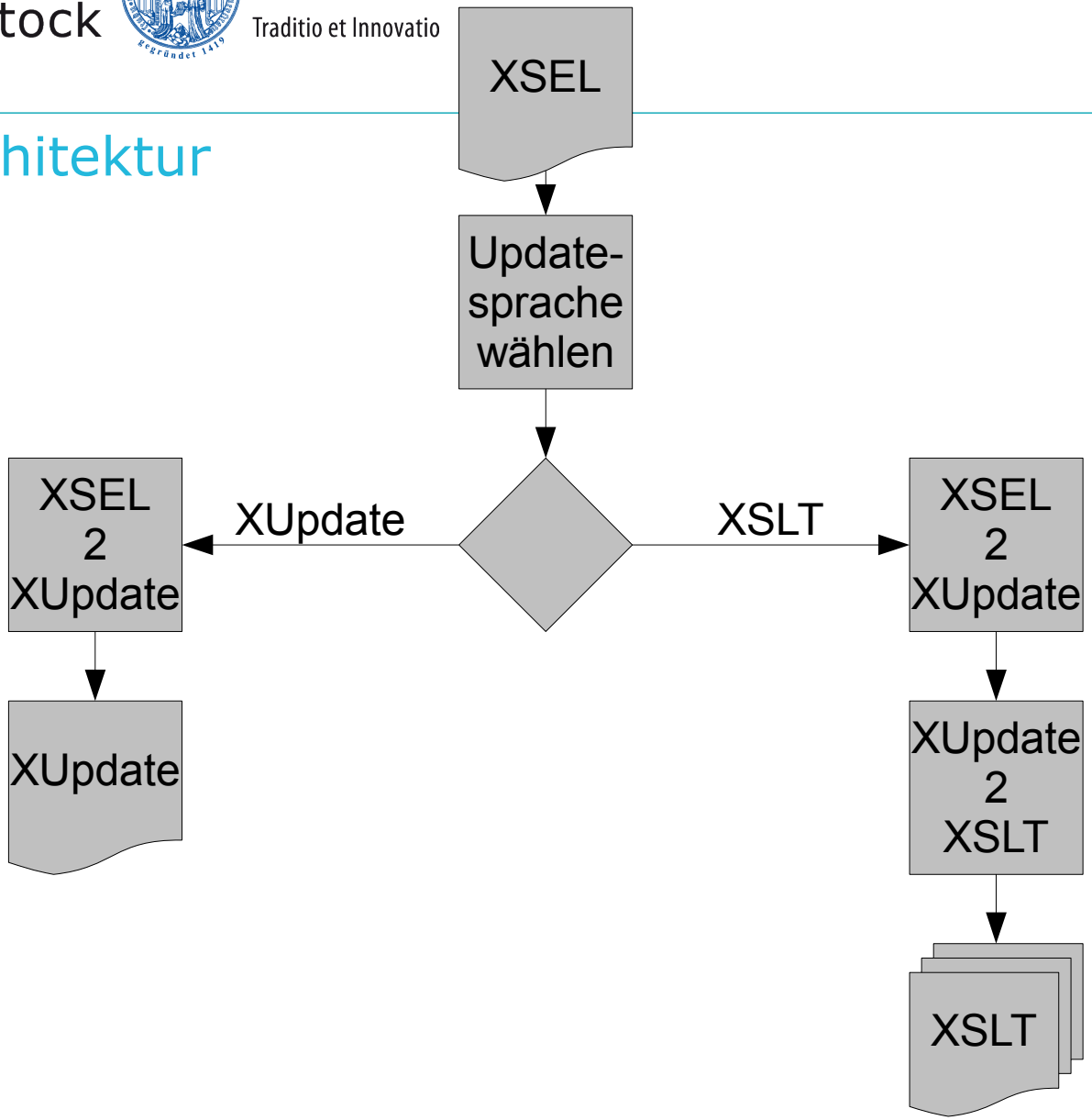
- Übersetzungsskript von Adrian Grigore nutzt ebenfalls XSLT um XUpdate in XSLT zu übersetzen



Aus [AG08]

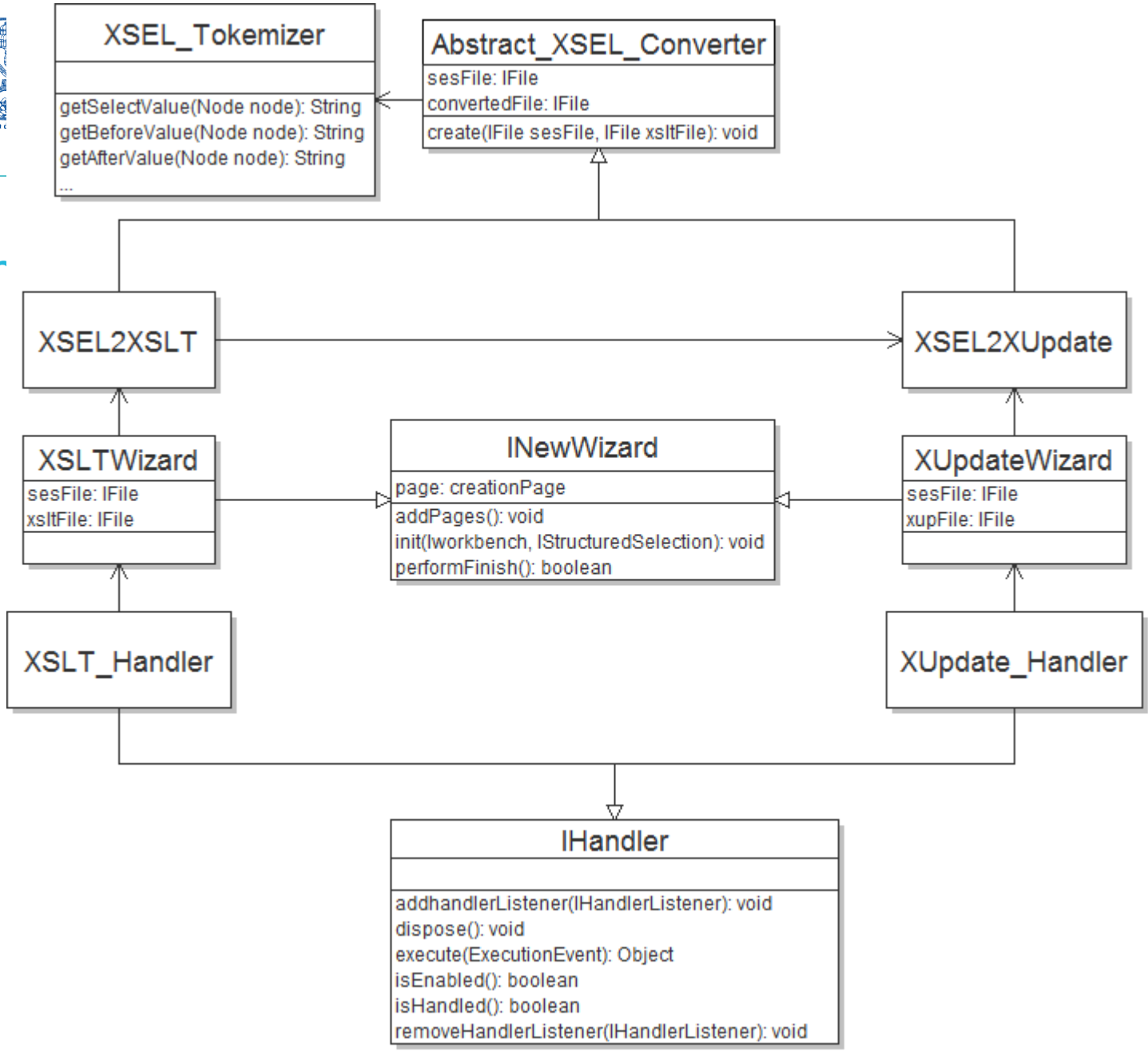
Inhalt

Architektur



Universität Rostock

Klassen-Diagramm



Umsetzung in CodeX – Quellcode XSEL2XUpdate

- Grundsätzlich:
  - Elemente und Attribute über DOM auslesen
  - Diese Informationen zum Aufbau der XUpdate-Datei nutzen

Alle Änderungen erfassen

```
public void parse()
{
    NodeList changes = doc.getChildNodes().item(0).getChildNodes();
    for(int i=0; i<changes.getLength(); i++)
    {
        Node change = changes.item(i);
        parseNode(change);
    }
}
```

Umsetzung in CodeX - Quellcode

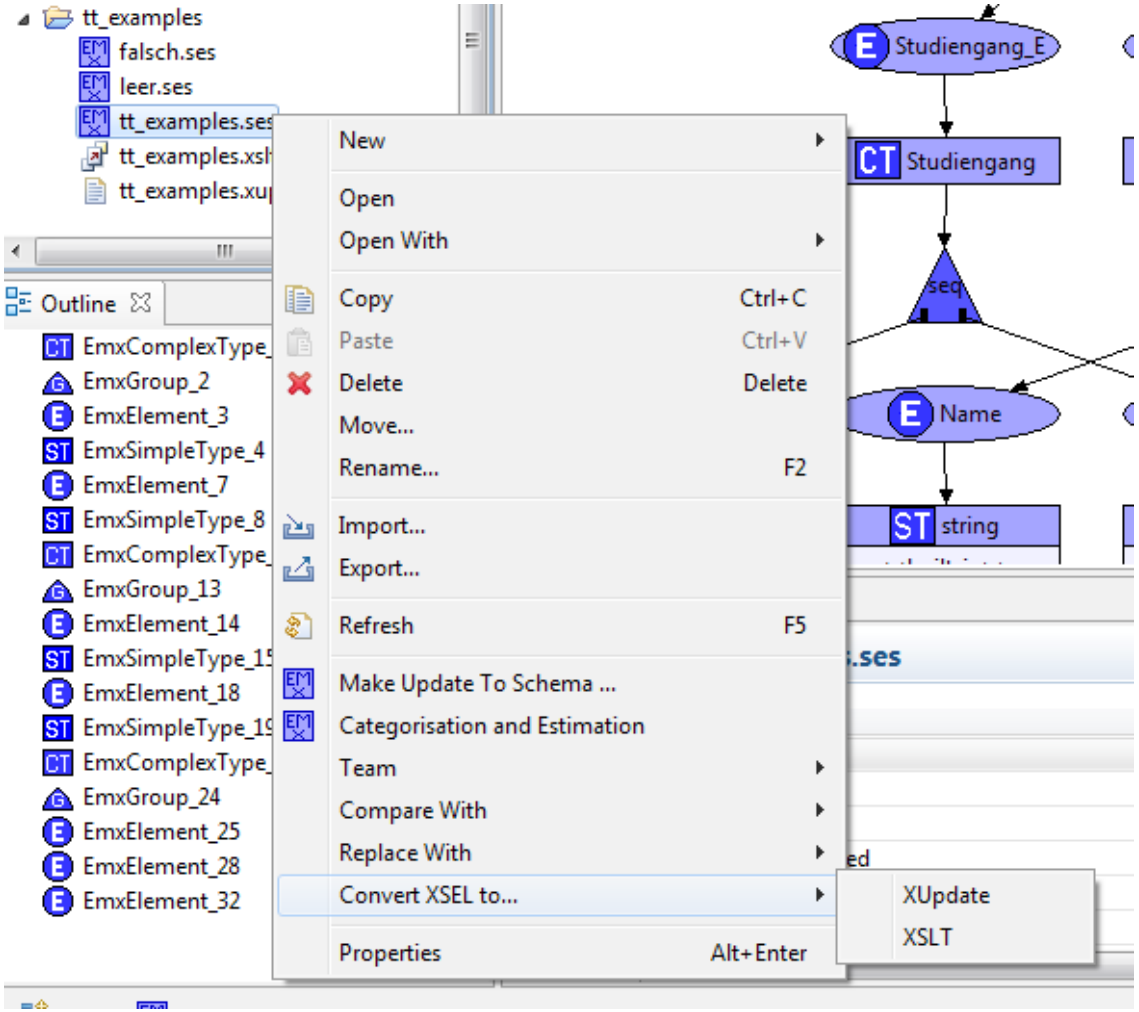
Operation ermitteln

```
private void parseNode(Node node)
{
    String changeName = node.getNodeName();
    if(changeName.equals("insert_before"))
    {
        pw.print("\t<"+XUP+"insert-before ");
        pw.println(convertSelectValue(node)+convertBeforeValue(node)
            +">");
        writeElement(node);
        pw.println("\t</"+XUP+"insert-before>");
    }
    if(changeName.equals("insert_after"))
    …
```

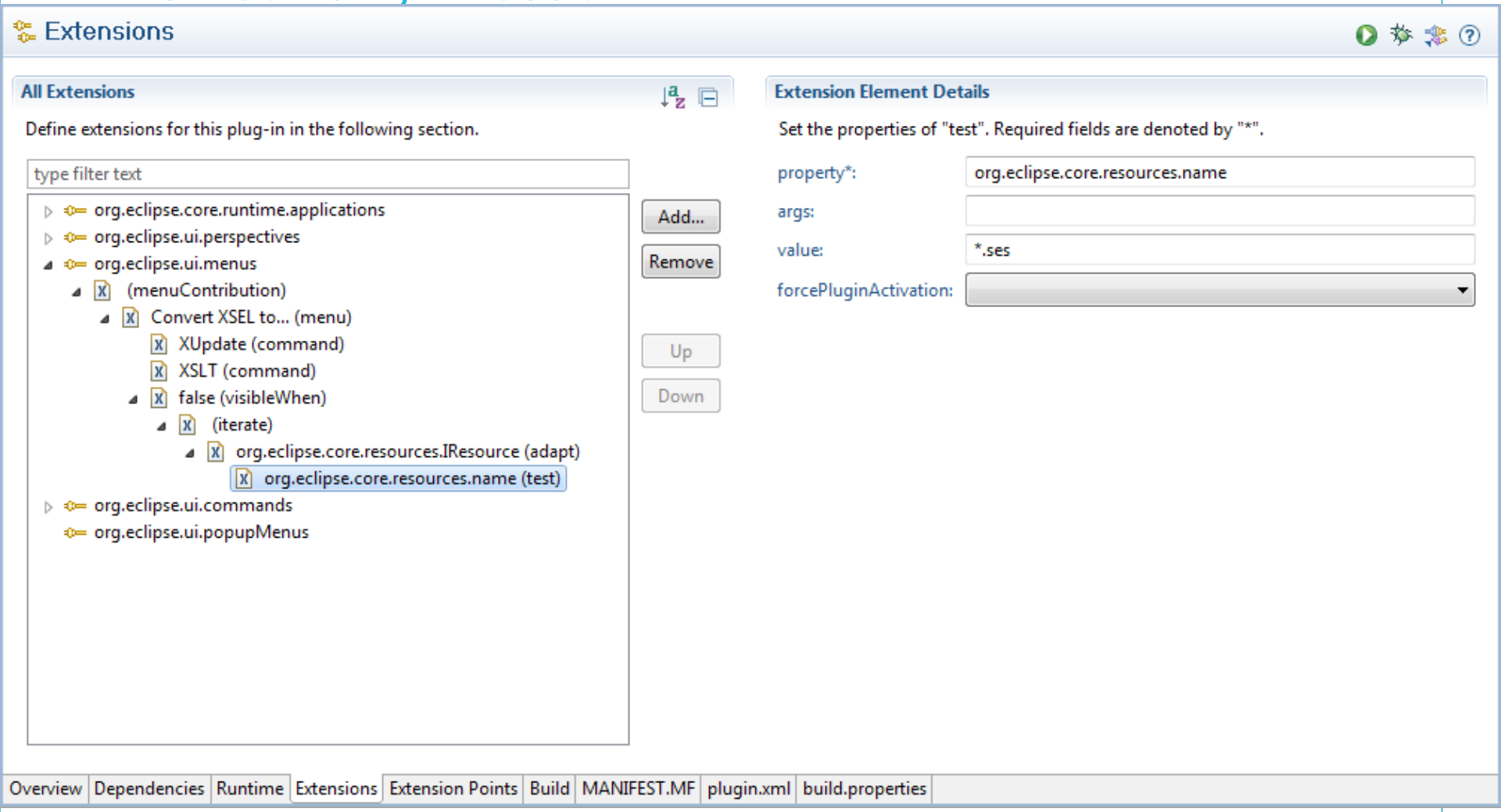
Sub-Funktionen nutzen DOM-Funktionalitäten aus, um XSEL-Attribute auszulesen und XUpdate-Anweisungen zu erstellen.

Umsetzung in CodeX

- Zugriff über Kontextmenü (Rechtsklick auf .ses-Datei)
- Dazu Erweiterung des Eclipse-Menüs notwendig

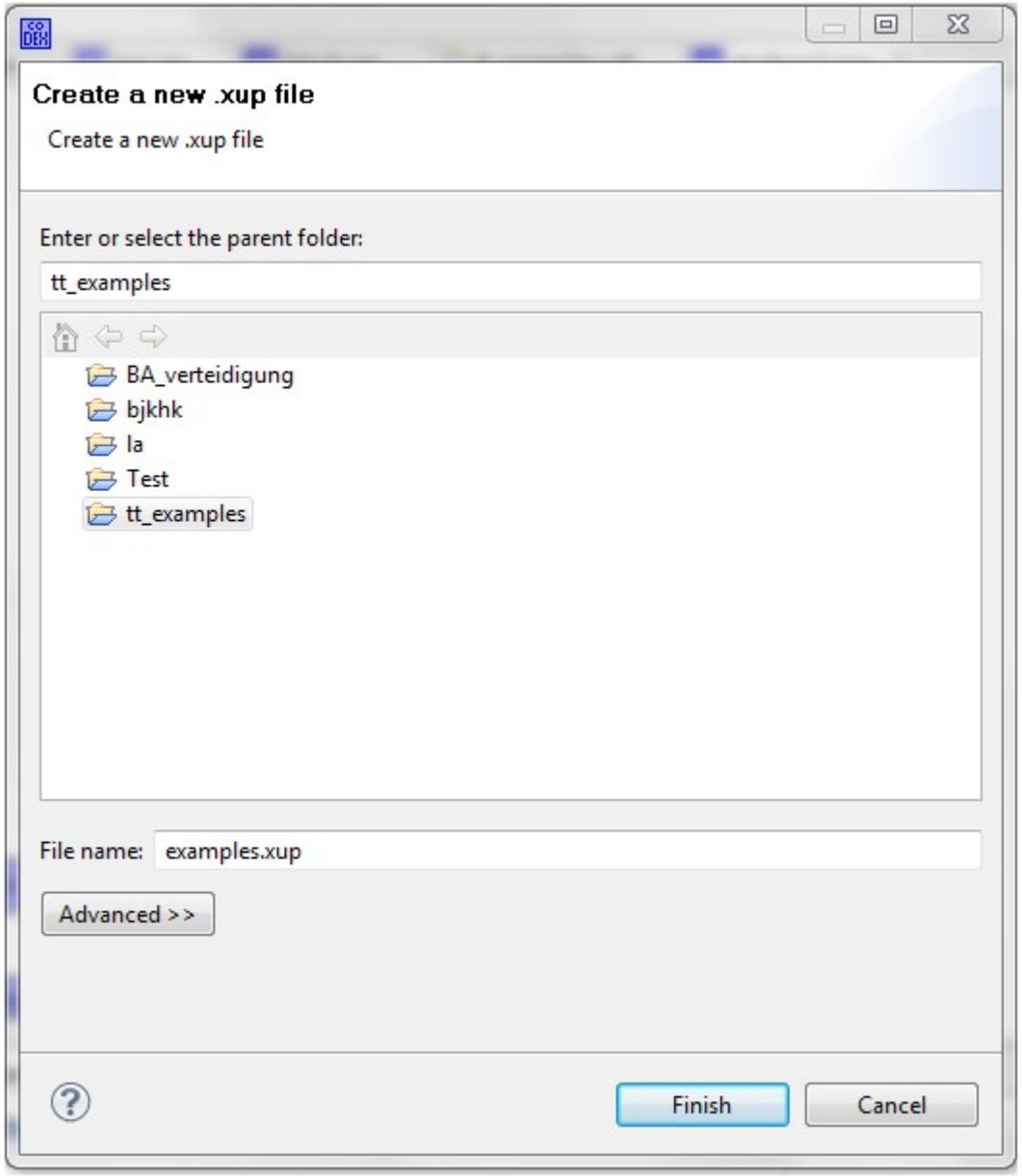


Umsetzung in CodeX



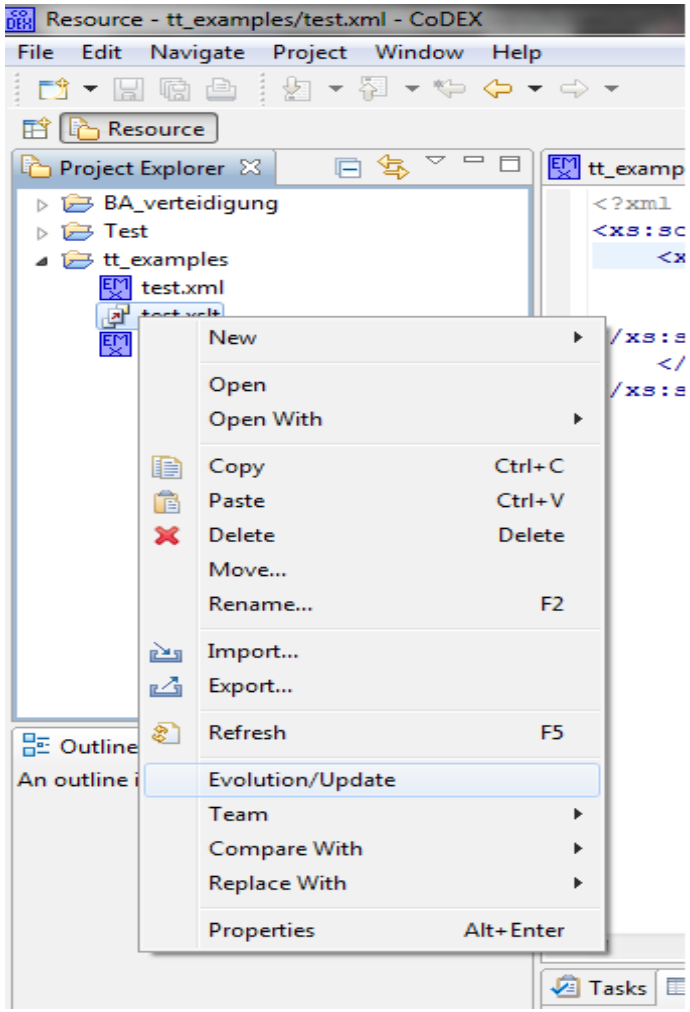
Umsetzung in CodeX

- Anlegen einer neuen Datei über Wizard
- Übersetzung erfolgt automatisch nachdem [Finish] ausgewählt wurde
- Datei erscheint nach erfolgreicher Übersetzung im Project-Explorer



Umsetzung in CodeX - Evolution

- Analog zur Übersetzung
- PopUp-Menü auf xslt-Datei
- Eintrag: Evolution/Update
- Auswahl der XML-Datei erfolgt über Dialog



Beispiel

- Anhand einer Live-Präsentation

Quellen

- [AG08] Adrian Grigore: XUpdate mittels XSLT, VDM Verlag Dr. Müller, 2008
- [TN12] Thomas Nösinger: persönliche Mitteilung vom 13.12.2011
- [TT05] Tobias Tiedt: Schemaevolution und Adaption von XML-Dokumenten und XQuery-Anfragen, Diplomarbeit, Universität Rostock, 2005
- [XML:DB] http://xmldb-org.sourceforge.net/index.html, zuletzt aufgerufen am 13.03.2012
- [XSLT1] http://www.w3.org/TR/xslt, zuletzt aufgerufen am 13.03.2012
- [XSLT2] http://www.w3.org/TR/xslt20/, zuletzt aufgerufen am 13.03.2012
- [Vog12] http://www.vogella.de/articles/EclipseRCP/article.html, zuletzt aufgerufen am 08.04.2012
- [XQu12] http://www.w3.org/TR/xquery-update-10/, zuletzt aufgerufen am 12.02.2012

Danke für Eure Aufmerksamkeit.

Fragen?