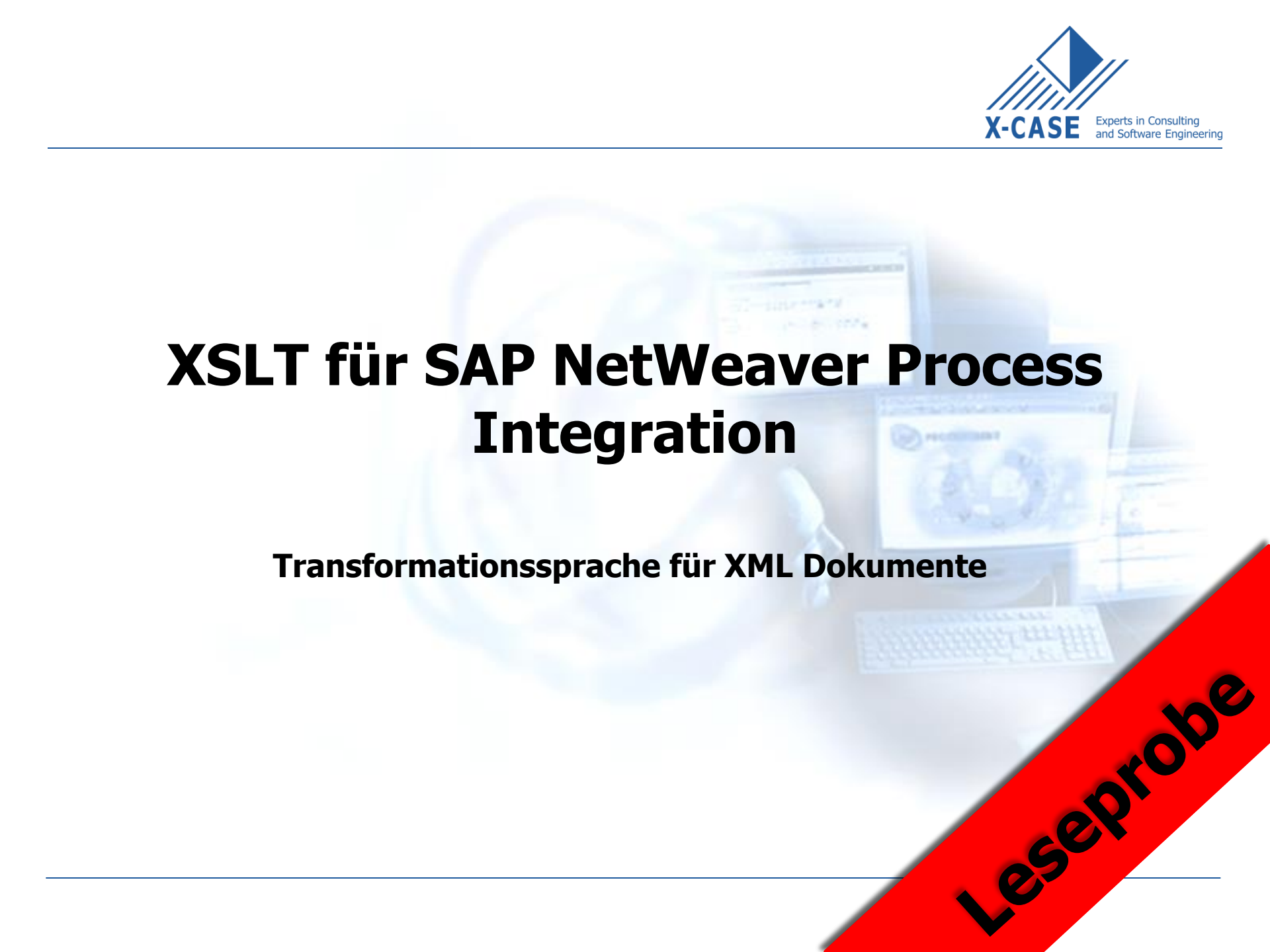
X-CASE Experts in Consulting and Software Engineering

XSLT für SAP NetWeaver Process Integration

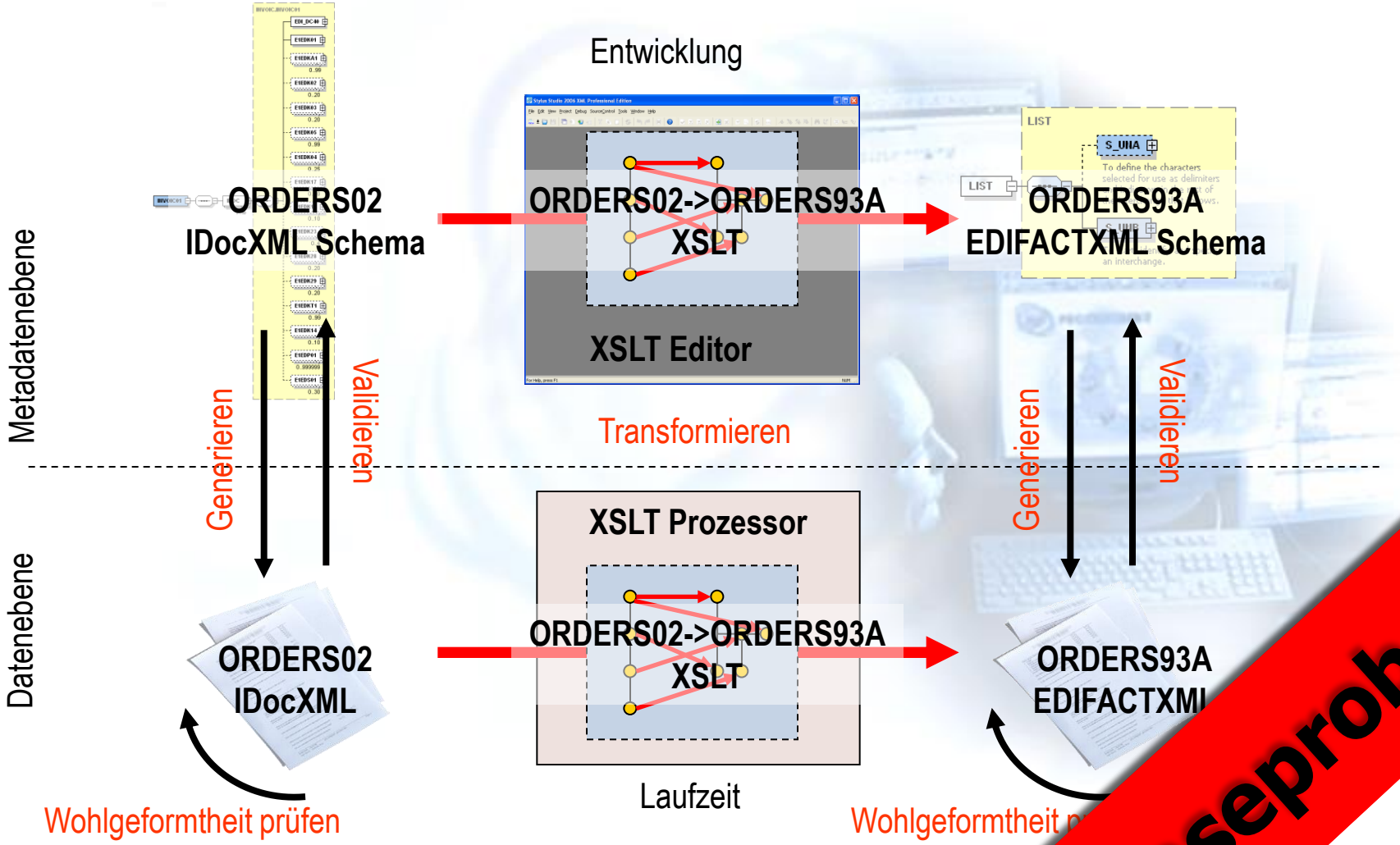
Transformationssprache für XML Dokumente



Leseprobe

Entwicklung und Laufzeit

X-CASE Experts in Consulting and Software Engineering



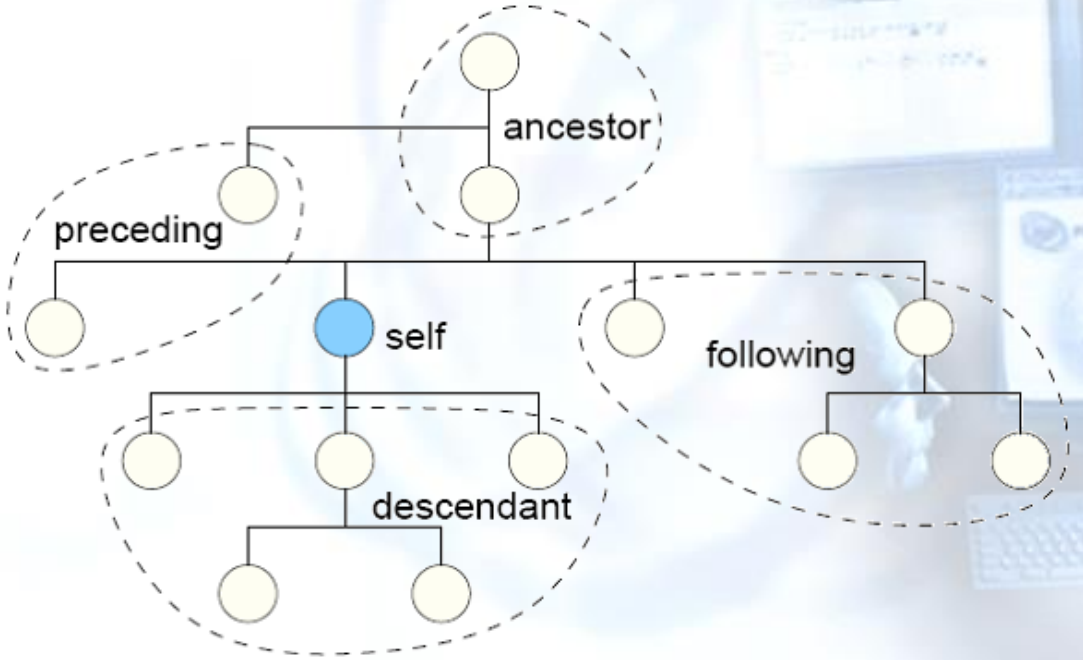
Prinzipien

X-CASE Experts in Consulting and Software Engineering

- **Ein XPath-Ausdruck setzt sich zusammen aus:**
  - einem oder mehreren Lokalisierungsschritten (location step), getrennt durch das Zeichen /.
- **Ein Lokalisierungsschritt besteht aus:**
  - Achse (axis) und
  - Knotentest (node-test),
  - optional gefolgt von einem oder mehreren Prädikaten (predicates).
- **Beliebig viele XPath-Ausdrücke lassen sich mit dem Zeichen | mengenmäßig vereinigen.**

Achsen

- Über Achsen erfolgt die Navigation vom Kontextknoten in andere Teile des XML-Dokumentes, das dabei als (DOM-)Baum betrachtet wird.



- Die fünf Achsen self, ancestor, descendant, preceding und followi
einem beliebigen Knoten den gesamten Dokumentbaum ab.

Knotentests

X-CASE Experts in Consulting and Software Engineering

- **Knotentests (geschrieben Achse::Knotentest) schränken die Elementauswahl einer Achse ein.**
  - Angabe eines Elementnamens wählt alle entsprechenden Elemente. (Beispiel: /descendant-or-self::E1EDP01 wählt alle Elemente im Dokument, die den Namen „E1EDP01“ haben.)
  - Mit dem Zeichen * wählt man beliebige Elemente. (Beispiel: /descendant-or-self::E1EDP01/child::* wählt alle Elemente im Dokument, die Kinder von Elementen mit dem Namen „E1EDP01“ sind.)
  - Mit text(), comment() und processing-instruction() lassen sich Knoten bestimmten Typs wählen.
  - Zu beachten ist, dass Attribute und Namensräume nicht in Knotentests, sondern als eigene Achse gewählt werden.

Prädikate (1)

X-CASE Experts in Consulting and Software Engineering

- **Durch Angabe von Prädikaten kann das Ergebnis weiter eingeschränkt werden.**
  - Prädikate werden in eckige Klammern eingeschlossen und können in beliebiger Zahl hintereinander geschrieben werden, wobei die Reihenfolge wesentlich ist.
  - Prädikate können XPath-Ausdrücke enthalten, außerdem kann eine Vielzahl von Funktionen verwendet werden.
  - Eine Zahl ist auch ein Prädikat und gibt den Index in einer Knotenmenge an.

- **Relationszeichen:**
  - = != and or < > <= >=

- **Zeichenkettenfunktionen:**
  - concat(), translate(), substring(), substring-before(), substring-aft
    with(), contains(), string-length(), …

Prädikate (2)

X-CASE Experts in Consulting and Software Engineering

- **Boolesche Funktionen:**
  - true(), false(), not()
- **Numerische Funktionen:**
  - + - * div mod
  - floor(), ceiling(), round()
- **Konvertierungen:**
  - string(), number(), format-number(), normalize-space(), …
- **Knotenmengen-Funktionen:**
  - name(), position(), last(), count() , sum(), id(), …
- **…mehr unter http://www.w3.org/TR/xpath/**

Stylesheets

X-CASE Experts in Consulting and Software Engineering

- Jedes XSLT Dokument hat ein **xsl:stylesheet** Kopfelement, ein optionales **xsl:output** Ausgabesteuerelement und ein **xsl:template (match="/")** Hauptprogramm.

- Alle weiteren **xsl:template** Elemente sind Unterprogramme, die nach dem Match- oder Name-Verfahren aufrufbar sind. Jedes **xsl:template** setzt den Kontext auf den aktuell gültigen Quellknoten.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <!-- Output control -->
  <xsl:output method="xml" indent="yes" encoding="utf-8"/>
  <!-- ********************** E1EDL37 template ********************** -->
  <xsl:template match="E1EDL37">
  </xsl:template>
  <!-- ********************** /E1EDL37 template ********************** -->
  <!-- ************************ Root template ************************ -->
  <xsl:template match="/">
    <ORDERS93A>
    </ORDERS93A>
  </xsl:template>
  <!-- ************************ /Root template ************************
</xsl:stylesheet>
```

Templates

X-CASE Experts in Consulting and Software Engineering

- Templates mit Name-Verfahren werden durch **xsl:template (name="...")** definiert und wie ein benanntes Unterprogramm durch **xsl:call-template (name="...")** aufgerufen.

- Match-Templates spezifizieren mit **xsl:template (match="...")** eine Art Muster, das für einen Knoten gelten muss, um das Template via **xsl:apply-templates (select="")** aufzurufen. Diese Muster können auch sehr dynamisch und überlagernd definiert werden.

```
<!-- ************ copy template ************ -->
<xsl:template match="@*|node()">
  <xsl:copy>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:copy>
</xsl:template>
<!-- ************ /copy template ************ -->

<xsl:apply-templates select="E1EDK01"/>
```

Schleifen

X-CASE Experts in Consulting and Software Engineering

- Schleifen werden durch das Element **xsl:for-each (select="...")** aufgrund einer Quellknotenmenge definiert, über die dann iteriert wird.

- Schleifen sind den **xsl:template** Unterprogrammen sehr ähnlich, befinden sich allerdings am Ort der Iteration, sind exakter in der Definition und Selektion als **xsl:template** mit **Match**-Verfahren und im Gegensatz zum **Name**-Verfahren anonym. Exakt gleich ist das Setzen des Kontextes auf den aktuell gültigen Quellknoten.

```
<ORDERS93A>
  <!--<S_UNA/>-->
  <S_UNB></S_UNB>
  <xsl:for-each select="*/IDOC">
    <M_ORDERS></M_ORDERS>
  </xsl:for-each>
  <S_UNZ></S_UNZ>
</ORDERS93A>
```