

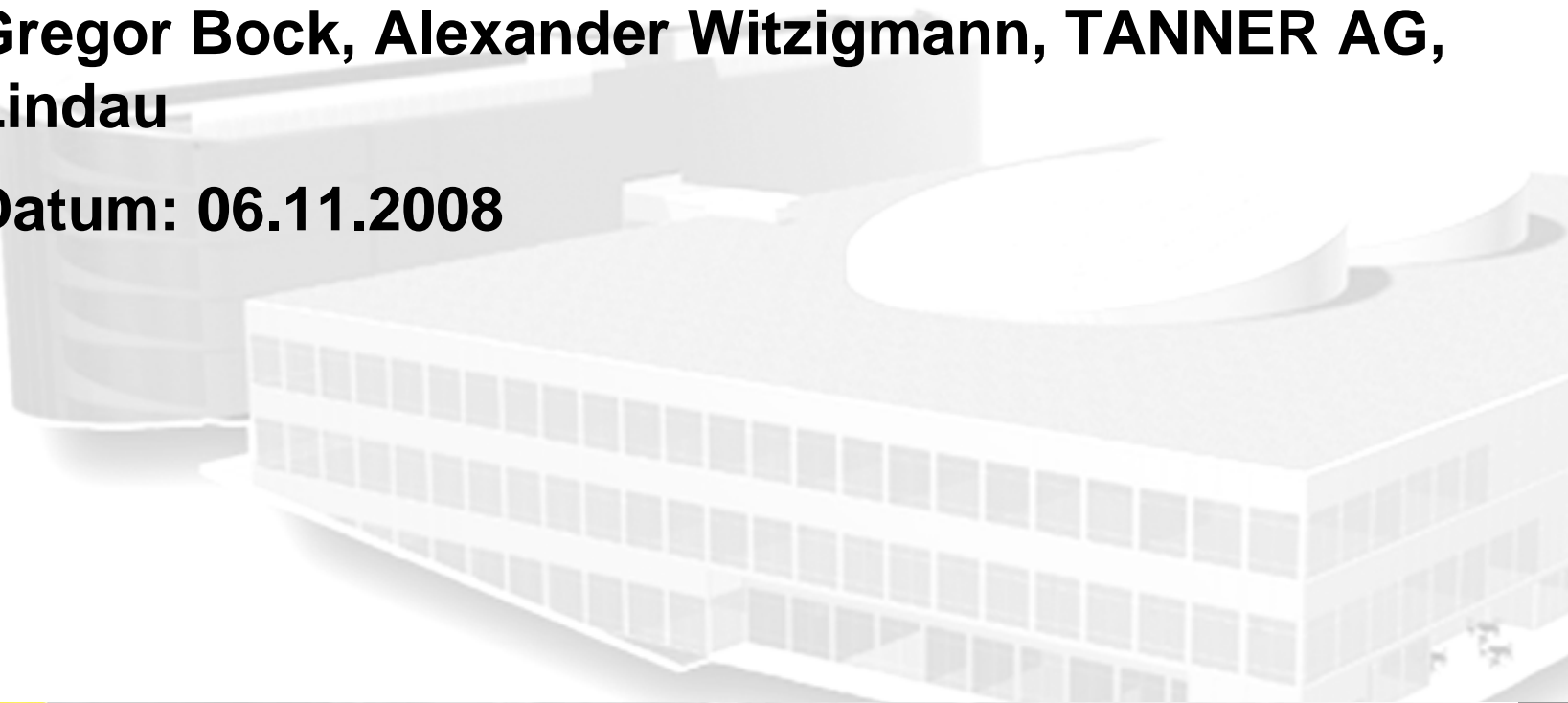


Automatisierte Dokumentation mit XML-Pipelining und XProc

Autoren:

**Gregor Bock, Alexander Witzigmann, TANNER AG,
Lindau**

Datum: 06.11.2008





Inhalt

- **Motivation**
- **Fliessbandprinzip**
- **Pipeline - Anwendungen aus der Praxis**
- **XProc – Der neue Standard**
- **Praxisbeispiel Calabash**
- **Zusammenfassung**



Motivation

- **These 1 – Nutzen von Information**
 - Nutzen einer Informationseinheit steigt mit der Anzahl der Verwendungen
 - Anzahl der Verwendungen ist identisch mit der Anzahl der Prozesse auf einer Informationseinheit
 - Anzahl der Prozesse erhöht die zu beherrschende Gesamtkomplexität

Je höher der Nutzen
umso
höher die zu beherrschende Gesamtkomplexität

Motivation

- **These 2 – Prozesse auf Information**
 - Prozesse auf Informationseinheiten sind in der Gesamtheit spezifisch
 - Unterschiede bzw. spezifische Anteile sind marginal
 - Zerlegung der Prozesse führt zu
 - Hoher Anzahl identischer Prozessbestandteile
 - Geringe Anzahl spezifischer Prozessbestandteile

Hohe Anzahl an spezifischen Prozessen
wird durch marginale Unterschiede hervorgerufen



Schlüssel zum Erfolg

- **Nutzenmaximierung durch**
 - Zerlegung der Prozesse in seine atomaren Bestandteile und dadurch
 - Maximaler Nutzen der Prozessbestandteile bzw. des darin enthaltenen Wissens

Nutzenmaximierung durch
Prozesszerlegung und -wiederverwendung

Fließbandprinzip

- **Warum?**
 - Automatisierung und Arbeitsteilung in industrialisierten Prozessen ist erfolgreich
- **Was?**
 - „Jeder macht nur das, was er besonders gut kann“
→ Arbeitsteilung
 - „und das möglichst häufig!“
→ Wiederverwendung
 - „und in höherer Taktrate!“
→ Automatisierung

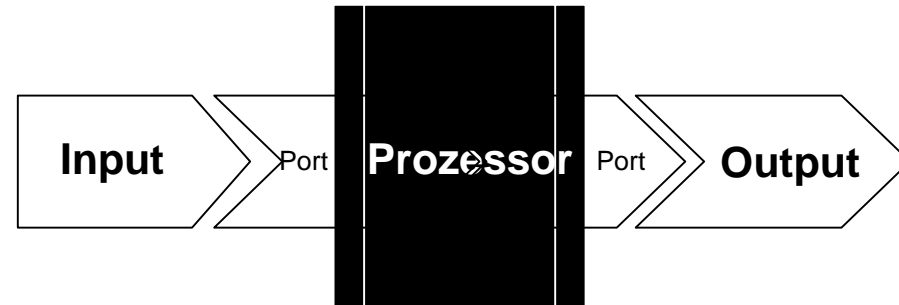




Fließbandprinzip in Informationsprozessen

- **Analogien**

- Dampfmaschine = Pipeline / Pipeline-Komponente
- Dampf/Wasser = XML Information
- Loch = Port



- **Prozessor / Pipeline-Komponente**

- Hat Eingänge (input ports)
- Hat zusätzlichen Eingang zur Konfiguration über Parameter
- Hat Ausgänge (output ports)

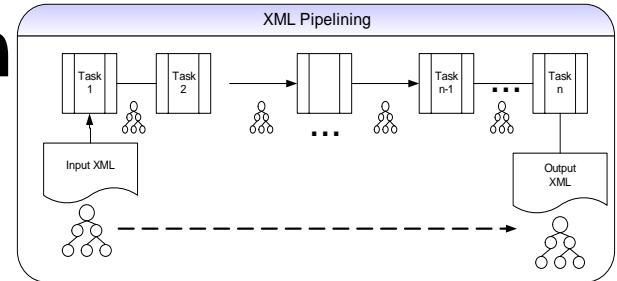


Was ist eine XML Pipeline?

- **Fliessbandprinzip angewendet auf XML basierten Informationen**
- **Abfolge von Prozessschritten (Steps)**
- **Jeder Prozessschritt modifiziert XML Information durch ein festgelegtes Regelwerk**
 - XSLT
 - XQuery
 - Java
- **Steuerung der Abfolge von Prozessschritten**
 - Sequenziell
 - Parallel
 - Wiederholend



XML Pipelining - Vorgehen



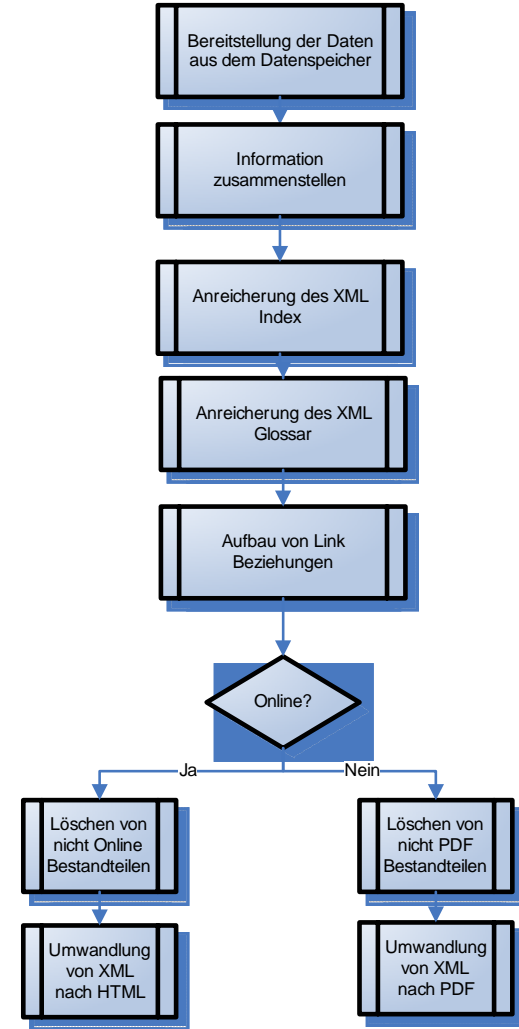
- Daten zum Zeitpunkt T sind gültig gegen die Spezifikation A
- Daten zum Zeitpunkt T2 sind gültig gegen die Spezifikation B
- Zerlegung der Transformation von A -> B in einzelne, unabhängige Teilschritte
- Überprüfen, ob Teilschritte bereits vorliegen
- Bereitstellung von noch nicht vorliegenden Teilschritten

Eine Ansammlung von kleinen, verwaltbaren und in sich abgeschlossenen Prozessen mit definierter Input und Output Spezifikation



XML Pipelining in der TD

- **Aufgabe: Automatisierte Erzeugung von PDF und Online Help**
- **Jede automatisierte / halbautomatisierte Erstellung von Informationsprodukten der TD ist das Ergebnis einer mehr oder weniger modularen Pipeline**

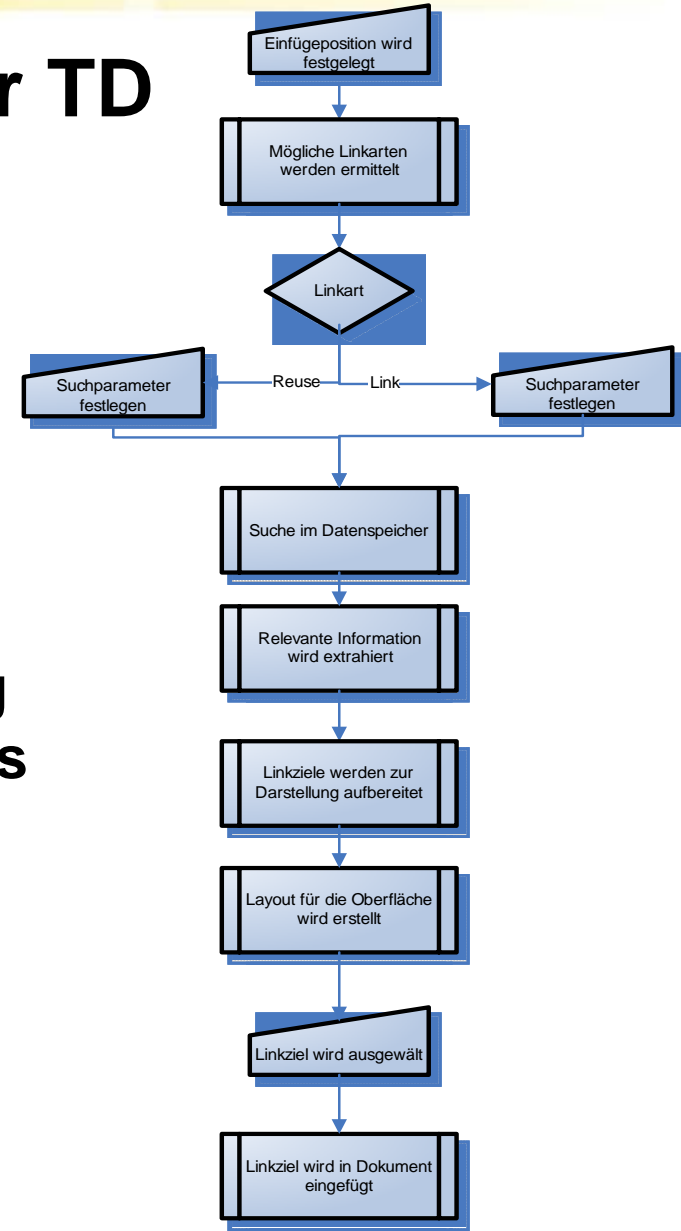


Vereinfachte Darstellung



XML Pipelining in der TD

- Aufgabe: Setzen von Beziehungen zwischen Informationsmodulen
- Jede automatisierte / halbautomatisierte Unterstützung bei der redaktionellen Anreicherung von Inhalten ist das Ergebnis einer mehr oder weniger modularen Pipeline



Vereinfachte Darstellung





Praxisbeispiele

- **DITA-OT**
 - XML Pipeline zur automatisierten Erstellung von Informationsprodukten aus DITA Quellen
 - ANT basiert
- **TANNER**
 - Serverbasierte XML Pipeline zur Erstellung von Informationsprodukten aus XML Quellen
 - Serverbasierte XML Pipeline zur Erstellung von Benutzeroberflächen aus XML Quellen
 - Orbeon (XPL basiert)



XProc, der neue Standard

- **XProc (XML Processing)**
 - Standardisierter Syntax für die Beschreibung von XML Pipelines
 - Keine standardisierte Ablaufumgebung
- **Alter Wein in neuen Schläuchen?**
 - XML Pipelining-Sprachen gab es schon vorher: XPL, XPD, Cocoon Sitemaps, SXPipe, Xpipe, DPML und Ant
 - Nun: Standardisierung durch W3C
 - Okt 2005 - W3C XML Processing Model Working Group
 - Aug 2008 - Status: W3C Working Draft
 - Sept 2008 – second „last Call ended
 - Nächster Schritt: Candidate Recommendation



XProc - Vorteile

- **Standardisierter Syntax**
 - Langfristig hohe Unterstützung von Toolherstellern aus unterschiedlichen Bereichen
 - Ablaufumgebung
 - Grafische Design Tools
 - Dedizierte Prozessoren für diverse Sprachen und Techniken
 -
 - Langfrist höhere Verbreitung des notwendigen Know-Hows
- **Vollständigkeit des Sprachumfangs**
 - Fehler und Einschränkungen von bestehenden Standards wurden berücksichtigt



XProc - Nachteile

- **Keine Ablaufumgebung mit produktivem Anspruch verfügbar**
- **Hoher, standardisierter Funktionsumfang erschwert die Umsetzung für Toolhersteller**

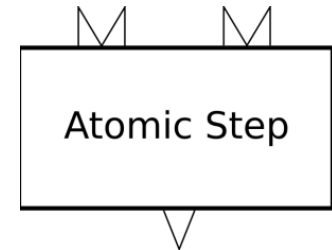


XProc Vokabular

- **Elemente zur Definition von Steps (Standard Step Library)**
- **Hauptelemente zur Steuerung des Datenflusses**
 - Schleifen
 - Bedingungen
 - Fehlerbehandlung
 - ...
- **zusätzliche Elemente zur Angabe von Input, Output, Optionen, Variablen und Parametern**

XProc, Steps

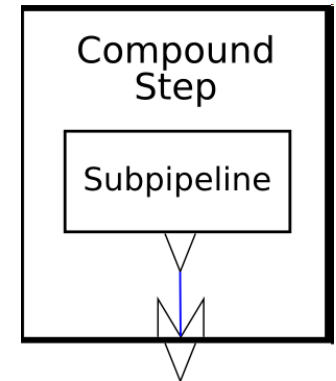
- **Atomare Steps**
Verarbeitungsschritte, die keine Subpipeline enthalten, wie z.B.
 - Filtern `<p:xslt>`
 - Inkludieren `<p:xinclude>`
 - Aufteilen `<p:split-sequence>`
 - Validieren `<p:validate-with-xml-schema>`



XProc, Kontrollstrukturen

- **Bündelung von Verarbeitungsschritten in Form von Kontrollstrukturen oder Subpipelines**

- „compound steps“,
ist ein Step, der eine Subpipeline enthält.
 - Schleifen `<p:for-each>`
 - Gruppe von Steps `<p:group>`
- „multi-container steps“
ist ein Step, der mehrere alternative Subpipelines enthält.
 - Fallunterscheidung `<p:choose>`: enthält subpipelines für `p:when` and `p:otherwise`
 - Fehlerbehandlung `<p:try>`: enthält subpipelines für `p:catch`





XProc Implementierungen

- **Calabash von Norman Walsh (JAVA)**
 - Referenzimplementierung des Standards
 - Aktuell nicht “Production Ready”
- **Yax (JAVA)**
 - Aktuell nicht “Production Ready”
- **EMC XML Store (ehemals X-Hive)**
- **Ankündigung: Implementierung von XProc in Orbeon**



Syntaxbeispiel Calabash

```
<?xml version="1.0"?>
```

```
<p:declare-step xmlns:p="http://www.w3.org/ns/xproc">
```

```
  <p:input port="source">
```

```
    <p:inline>
```

```
      <doc>Congratulations! You've run your first pipeline!</doc>
```

```
    </p:inline>
```

```
  </p:input>
```

```
  <p:output port="result"/>
```

```
  <p:identity/>
```

```
</p:declare-step>
```



Zusammenfassung

- Nutzenmaximierung durch Prozesszerlegung und –wiederverwendung in der TD wird durch XML Pipelining ermöglicht
- Standardisierung der dafür notwendigen Beschreibung schafft die notwendige Basis für
 - Höhere Anzahl von verfügbaren Tools und Know-How ohne spezifische Einarbeitung in Syntax
 - Unternehmensübergreifende Verwendung von zentralen Bausteinen und Transformationen für spezifische Anwendungsfälle

Erst die Zukunft verbindet beide genannten Punkte