

An illustrative coupled transformation of a I O I implementation

Ralf Lämmel
on behalf of IOI companies.org

Let's transform XML document, XML schema, and XSLT program simultaneously!

Let's familiarize with l0limplementation:xslt!

Headline

--- XML processing with XSLT ---

Languages

- ▶ XML
- ▶ XSD
- ▶ XSLT

Technologies

- ▶ xsltproc
- ▶ Saxon

Testing and packaging is tailored to serve these two alternatives for an XSLT implementation.

In principle, any XSLT implementation should do.

Features

- ▶ Tree structure
- ▶ Type-driven query
- ▶ Type-driven transformation
- ▶ Structure-driven query
- ▶ Data import
- ▶ Data export

Eventually, we want to rename
“salary” to “income”.

The plan:

Rename “salary” to “income”

- **Instance** level:
actual company documents
- **Schema** level:
the schema of company documents
- **Program** level:
XSLT transformations of company documents

Renaming -- *Instance* level

“salary”
to become
“income”.

```
<employee>  
  <name>Ralf</name>  
  <address>Koblenz</address>  
  (<salary>1234</salary>)  
</employee>
```

A fragment of the sample company.

Renaming -- *Schema* level

```
<xs:complexType name="employee">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="address"/>
    <xs:element ref="salary"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="name" type="xs:string"/>
<xs:element name="address" type="xs:string"/>
<xs:element name="salary" type="xs:double"/>
```

A fragment of the XML schema for companies.

Renaming -- *Program* level

```
<xsl:template match="com:salary">  
  <xsl:copy>  
    <xsl:value-of select=". div 2"/>  
  </xsl:copy>  
</xsl:template>
```

A fragment of the XSLT
program for cutting salaries.

Summary of transformation

- *XML* serves for object representation at all levels.
- *Java* as the meta-language using *JDOM* API
- *Instance* level is straightforward.
- *Type* level “involves” *XML Schema* details.
- *Program* level “involves” *XSLT* details.

The demo is available from github:

<https://github.com/rlaemmel/couplingI0Idemo/downloads>

*... or you git clone / fork,
if you are familiar with git/github.*

See here specifically:

[https://github.com/rlaemmel/couplingI0Idemo/blob/master/
transformation/RenameElementName.java](https://github.com/rlaemmel/couplingI0Idemo/blob/master/transformation/RenameElementName.java)

Disclaimer

- Low-level, ad-hoc implementation.
- Not all XSLT patterns covered.
- The regexp patterns are not safe.

Further reading

See “coupled software transformations”, e.g.:

- The CSXW 2011 workshop
- Work by J.Visser et al., e.g., [BerdaguerCPV07]

Summary:

What's the point of this demo?

- To demonstrate “meta-level” value of I0I companies.
- To easily get started with re-engineering on top of I0I.
- To motivate more profound re-engineering experiments.
- (To enjoy the XML technological space.)

Thanks!

- *Contact I0I companies*
- *Email: I0Icompanies@gmail.com*
- *Twitter: @I0Icompanies*
- Material for this presentation:
- <https://github.com/rlaemmel/couplingI0Idemo>