Schematron Query Language Binding

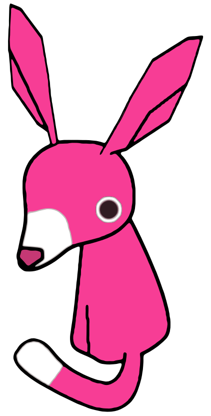
XATAPULT
CONTENT ENGINEERING

Erik Siegel
www.xatapult.com
erik@xatapult.nl

Codes must start with the department code

```
<inventory-list depcode="IMP">
  <article code="IMP0001">
    <name>Bolts</name>
    <description>Nuts to secure things with</description>
  </article>
  <article code="EXP0234">
    <name>Bananas</name>
    <description>Delicious ripe bananas</description>
  </article>
</inventory-list>
```

Schematron



```
<schema xmlns="http://purl.oclc.org/dsdl/schematron">
  <let name="department-code" value="/inventory-list/@depcode"/>
  <pattern>
    <rule context="article">
      <assert test="starts-with(@code, $department-code)">
        The article code (<value-of select="@code"/>) must start with the right
        prefix (<value-of select="$department-code"/>) for <value-of select="name"/>
      </assert>
    </rule>
  </pattern>
</schema>
```

```
The article code (EXP0234) must start with the right prefix (IMP) for Bananas
```

Schematron is a container language with a *configurable* query language for its expressions:

```
<schema xmlns="http://purl.oclc.org/dsdl/schematron">

  <let name="department-code" value="/inventory-list/@depcode"/>
  <pattern>
    <rule context="article">
      <assert test="starts-with(@code, $department-code)">
        The article code (<value-of select="@code"/>) must start with the right
        prefix (<value-of select="$department-code"/>) for <value-of select="name"/>
      </assert>
    </rule>
  </pattern>

</schema>
```

Query Language Binding or QLB

The **queryBinding** attribute

Specify the Query Language Binding using the **`queryBinding`** attribute on the root element (default value: **`xslt`**):

```
<schema xmlns=http://purl.oclc.org/dsdl/Schematron" queryBinding="…">
  …
</schema>
```

Reserved and **defined** Query Language Binding names:

- **`exslt`**
- **`stx`**
- **`xslt`**, **`xslt2`**, **`xslt3`**
- `xpath`, **`xpath2`**, **`xpath3`**, `xpath31`
- `xquery`, `xquery3`, `xquery31`

Supported Query Language Bindings (by SchXslt):

- **`xslt`**, **`xslt2`**, **`xslt3`**

How to create a new Query Language Binding?

1. Choose a (free) identifier for your query language binding
2. Write a specification document
   - The contents of such a document is defined in the Schematron standard
3. Write a processor that can handle the binding

Query Language Binding in practice…

- Only for XSLT (and maybe XPath)
- Advice:
  - Use **`xslt2`** or **`xslt3`**
  - Always specify the QLB (using the **`queryBinding`** attribute)
    - Otherwise, you get the default value: **`xslt`**
      - which means XPath 1.0
        - which is rather limiting…

The xslt2/xslt3 Query Language Binding

- Use XPath 2.0 or 3.1 expressions
- Use **`xsl:key`** to define keys and the **`key()`** function for lookups
- Add your own XSLT functions with **`xsl:function`** and use them in XPath expressions

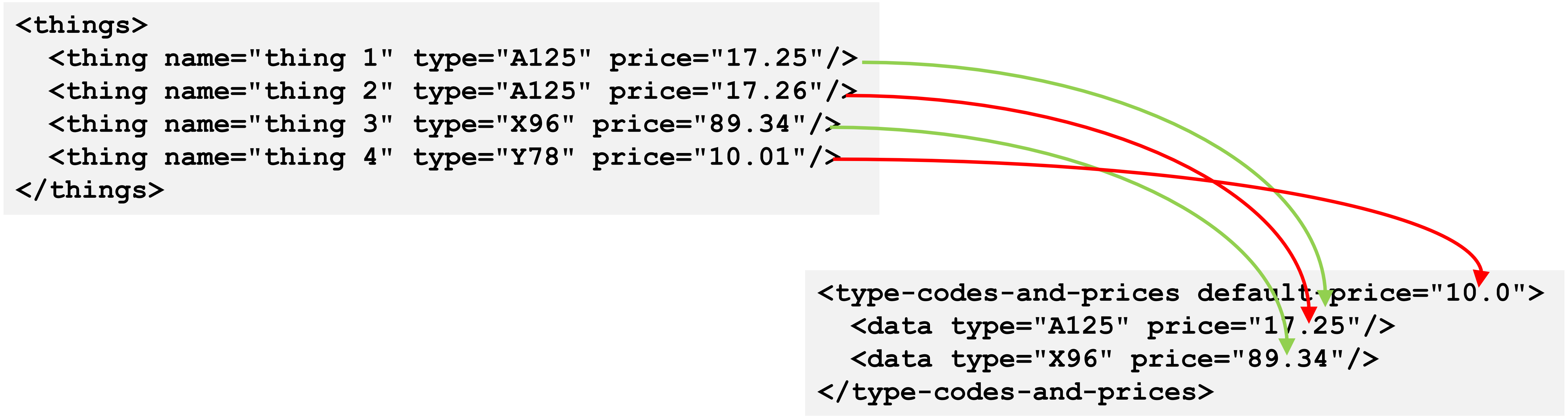
- Officially no other XSLT constructs, like:
  - **`xsl:include/xsl:import`**
  - **`xsl:template`**
  - Global XSLT variables
- But this usually works fine…

Example of using xsl:key

```
<orders>
  <item id="bolts" price="5.49">A box with 20 bolts</item>
  <item id="nuts" price="3.78">A box with 20 nuts</item>
  <!-- … many, many more items… -->
  <order>
    <ordered-item id-ref="bolts" quantity="5"/>
    <ordered-item id-ref="nuts" quantity="10"/>
  </order>
  <!-- … many, many more orders… -->
</orders>
```

```
<schema xmlns="http://purl.oclc.org/dsdl/schematron"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" queryBinding="xslt3">

  <xsl:key name="item-ids" match="/*/item" use="@id"/>

  <pattern>
    <rule context="ordered-item">
      <assert test="exists(key('item-ids', @id-ref))">
        The referenced item <value-of select="@id-ref"/> does not exist
      </assert>
    </rule>
  </pattern>
</schema>
```

Example of defining a function - 1



```
<things>
  <thing name="thing 1" type="A125" price="17.25"/>
  <thing name="thing 2" type="A125" price="17.26"/>
  <thing name="thing 3" type="X96" price="89.34"/>
  <thing name="thing 4" type="Y78" price="10.01"/>
</things>
```

```
<type-codes-and-prices default-price="10.0">
  <data type="A125" price="17.25"/>
  <data type="X96" price="89.34"/>
</type-codes-and-prices>
```

Example of defining a function - 2

```
<schema xmlns="http://purl.oclc.org/dsdl/schematron"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform" queryBinding="xslt3">

 <ns uri="#functions" prefix="f"/>

 <xsl:function name="f:get-price" as="xs:double">
   <xsl:param name="type" as="xs:string"/>
   <xsl:variable name="prices-document" as="document-node()"
     select="doc('type-codes-and-prices.xml')"/>
   <xsl:variable name="data-element-for-type" as="element(data)?"
     select="$prices-document//data[@type eq $type]"/>
   <xsl:choose>
     <xsl:when test="exists($data-element-for-type)">
       <xsl:sequence select="xs:double($data-element-for-type/@price)"/>
     </xsl:when>
     <xsl:otherwise>
       <xsl:sequence
         select="xs:double($prices-document/type-codes-and-prices/@default-price)"/>
     </xsl:otherwise>
   </xsl:choose>
 </xsl:function>
```

Aside: Namespaces for expressions in Schematron

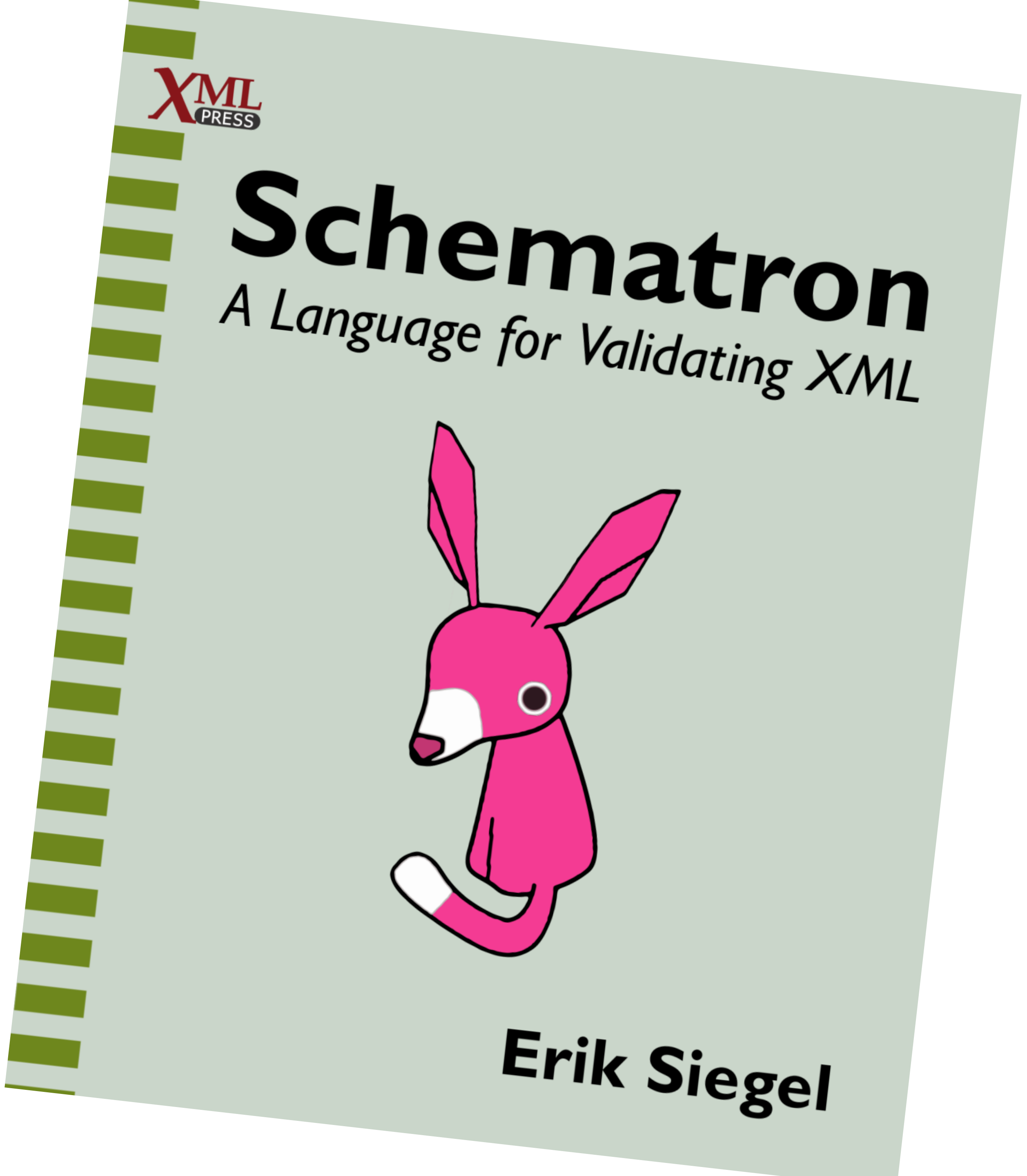
```
<schema xmlns="http://purl.oclc.org/dsdl/schematron"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" queryBinding="xslt3"
  xmlns:f="#functions"
>

  <xsl:function name="f:get-price" as="xs:double">
    …
  </xsl:function>
```

Example of defining a function - 3

```
<schema xmlns="http://purl.oclc.org/dsdl/schematron"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" queryBinding="xslt3">

  <ns uri="#functions" prefix="f"/>

  <xsl:function name="f:get-price" as="xs:double">
    …
  </xsl:function>

  <pattern>
    <rule context="thing">
      <let name="expected-price" value="f:get-price(@type)"/>
      <assert test="$expected-price eq xs:double(@price)">
        The price for <value-of select="@name"/> should be
        <value-of select="$expected-price"/>
      </assert>
    </rule>
  </pattern>

</schema>
```

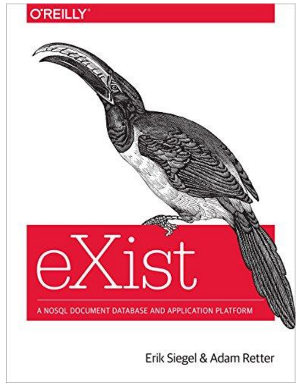
Wrap-up

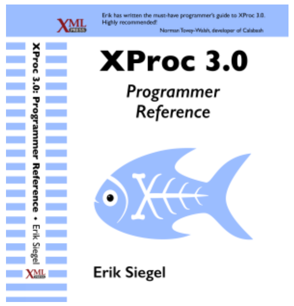
- Schematron is defined as flexible with regards to its query language
  - The Query Language Binding or QLB
- In practice: **`xslt`** (and therefore XPath) only
- Set this using the **`queryBinding`** attribute. Recommended values:
  - **`queryBinding="xslt2"`**
  - **`queryBinding="xslt3"`**
- Allows for XPath 2.0 or 3.1 expressions
- XSLT2 and XSLT3 type query bindings also allow XSLT keys and functions
- Other XSLT features are officially unsupported but usually work



Available: September 2022

My other books


O'REILLY
eXist
A NOSQL DOCUMENT DATABASE AND APPLICATION PLATFORM
Erik Siegel & Adam Retter

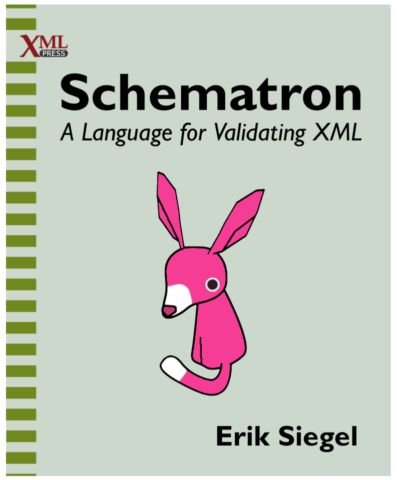

XML PRESS
Erik has written the must-have programmer's guide to XProc 3.0. Highly recommended!
Norman Tovey-Walsh, developer of Calabash
XProc 3.0
Programmer Reference
XProc 3.0: Programmer Reference • Erik Siegel
Erik Siegel

Book contents

- Introductions
- Schematron in context (of other validation languages)
- How to apply Schematron (with SchXslt, in Oxygen, etc.)
- Basics (patterns, rules, asserts/reports, value-of, variables, namespaces)
- Advanced (diagnostics, phases, abstract rules/patterns, includes)
- Query language binding
- Additional features (markup, flags, roles, etc.)
- Some further examples and recipes

Appendices:

- XPath technology primer
- Introduction to namespaces
- Schematron & SVRL reference
- Introduction to SQF
- Additional reading



Questions?

XATAPULT
CONTENT ENGINEERING

Erik Siegel
www.xatapult.com
erik@xatapult.nl