DITA NG
A RelaxiNG implementation of DITA
- Part 3 -

George Cristian Bina

@georgebina
george@oxygenxml.com

syncro soft

oXygen XML Editor

Part 1

A proof of concept showing that it is possible to have a DITA implementation based on Relax NG

Presented at

- XML Prague 2011

and

- DITA North America 2011

Part 2

DITA in Relax NG in action and a focus on its advantages over the DTD implementation

Presented at

- DITA North America 2012

  and

- Information Energy 2012

Part 3

- Quick overview
- Current state
- What is next

Hands-on introduction

A hands-on introduction on how you can use DITA with Relax NG support with DITA-OT

Download distributions

- Get a DITA OT distribution

  http://dita-ot.sourceforge.net/latest/

  Follow any of

  - DITA-OT stable release
  - DITA-OT latest development build

  at the bottom of that page and download the full_easy_install distribution

- Get a DITA-NG distribution

  http://code.google.com/p/dita-ng/downloads/list

  Get the latest/featured distribution

Installation

Place DITA-OT and DITA-NG distributions in a folder, for example you may have

- DITA-OT1.6.3_full_easy_install_bin.zip
- Dita-ng20121108.zip

Unzip the two archive

- DITA-OT1.6.3
- org.dita-ng.doctypes

Move org.dita-ng.doctypes inside DITA-OT1.6.M3/plugins

DITA-NG setup

Open the README.txt file from the RelaxNG folder and follow the setup instructions:

- Edit startcmd.sh to add the following line

  NEW_CLASSPATH="$DITA_DIR/plugins/org.dita-ng.doctypes/lib/dita-ng.jar:$DITA_DIR/plugins/org.dita-ng.doctypes/lib/jing.jar:$NEW_CLASSPATH"

- Start a terminal and cd to the DITA-OT1.6.3 folder
- Set DITA_HOME for example using

  export DITA_HOME=.

- Run startcmd.sh to start a new shell and integrate the plugin

  ant -f integrator.xml

Test the integration

- ant -f build.xml

  -Dargs.input=plugins/org.dita-ng.doctypes/demo/flowers/flowers.ditamap

  -Doutput.dir=plugins/org.dita-ng.doctypes/demo/flowers/out

  -Dtranstype=xhtml

  The result will be in plugins/org.dita-ng.doctypes/demo/flowers/out/index.html

- ant -f build.xml

  -Dargs.input=plugins/org.dita-ng.doctypes/demo/flowers/flowers.ditamap

  -Doutput.dir=plugins/org.dita-ng.doctypes/demo/flowers/out

  -Dtranstype=pdf

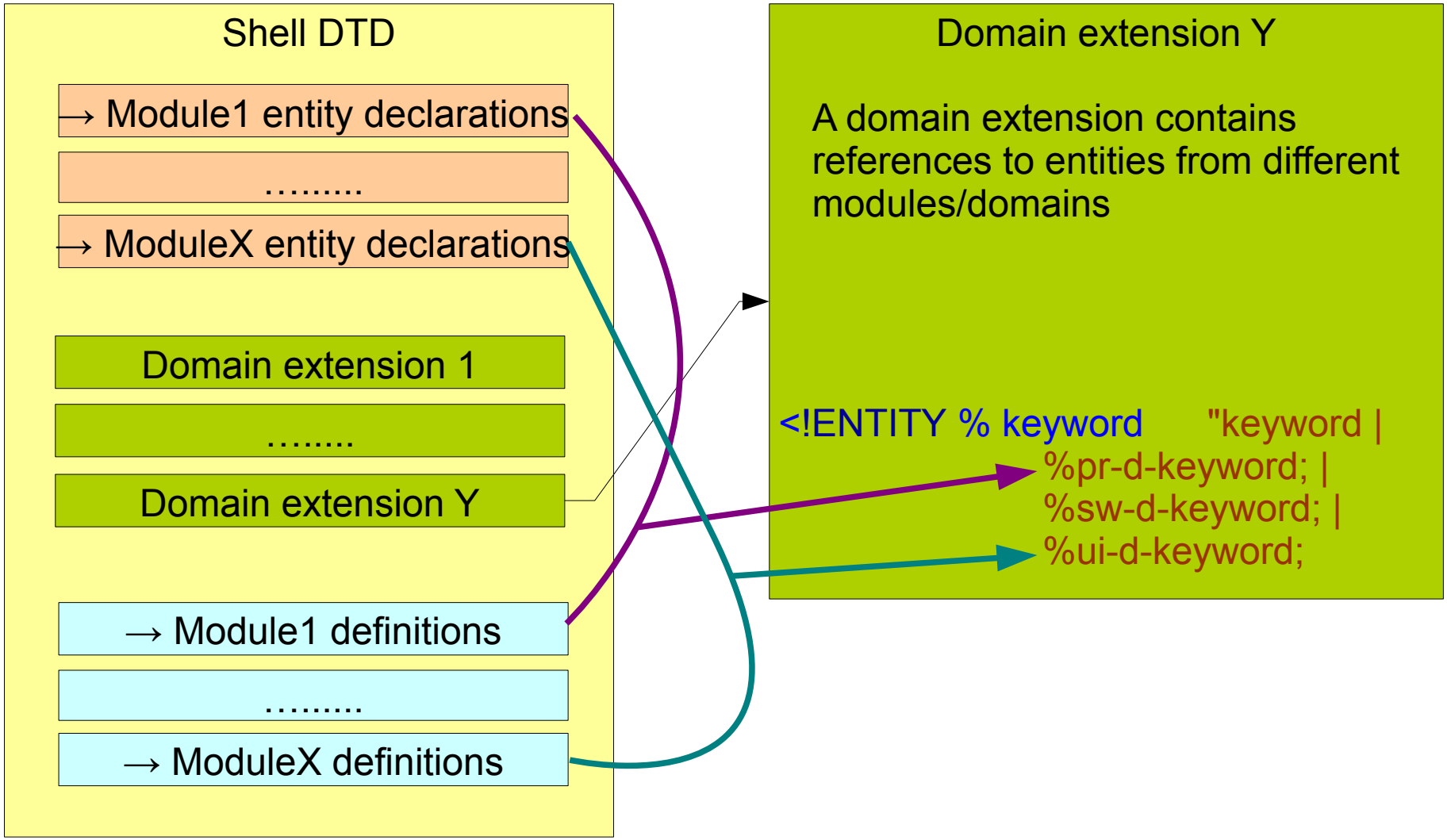
  The result will be in plugins/org.dita-ng.doctypes/demo/flowers/out/flowers.pdf

Advantages of implementing DITA in Relax NG

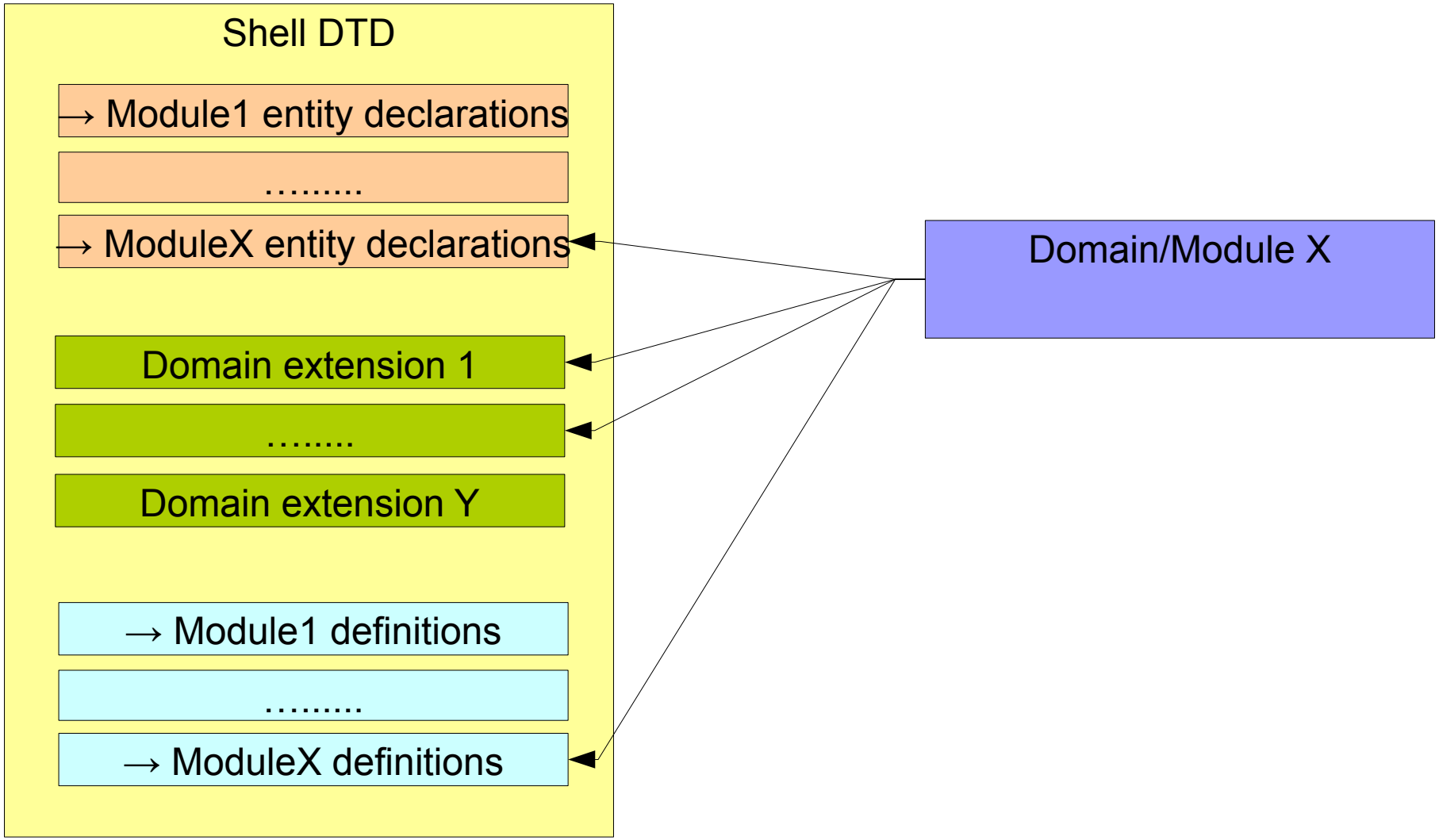
Advantages of implementing DITA in Relax NG

- Simplified shell structure

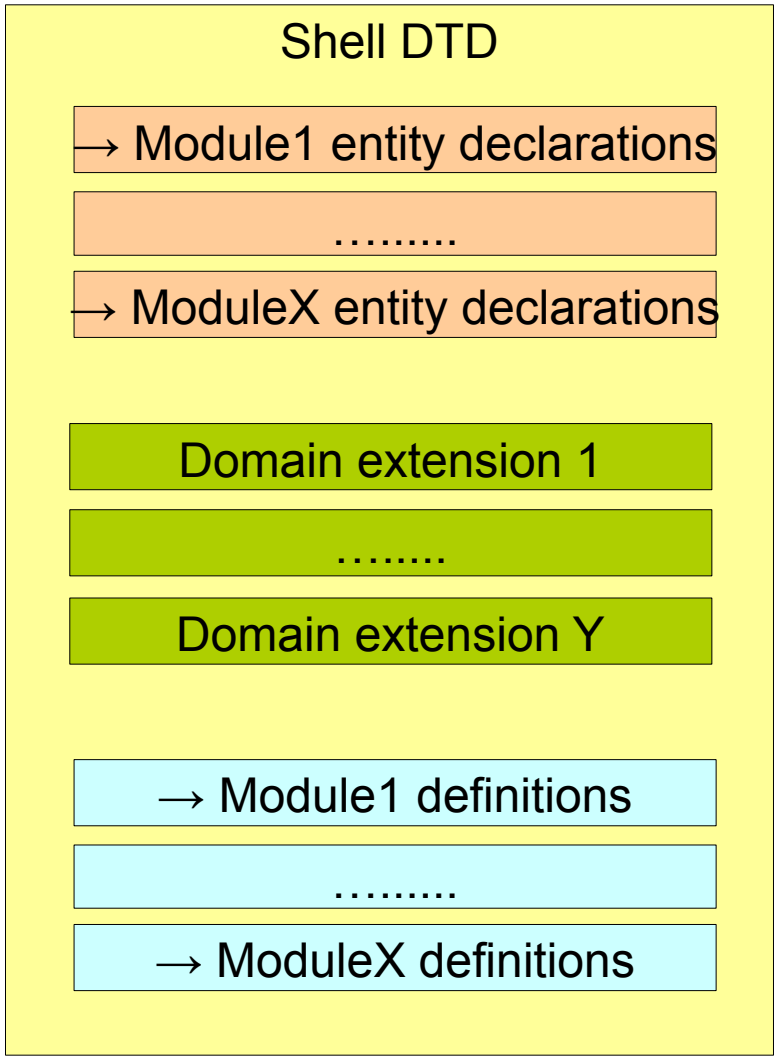
DITA implementation with DTDs

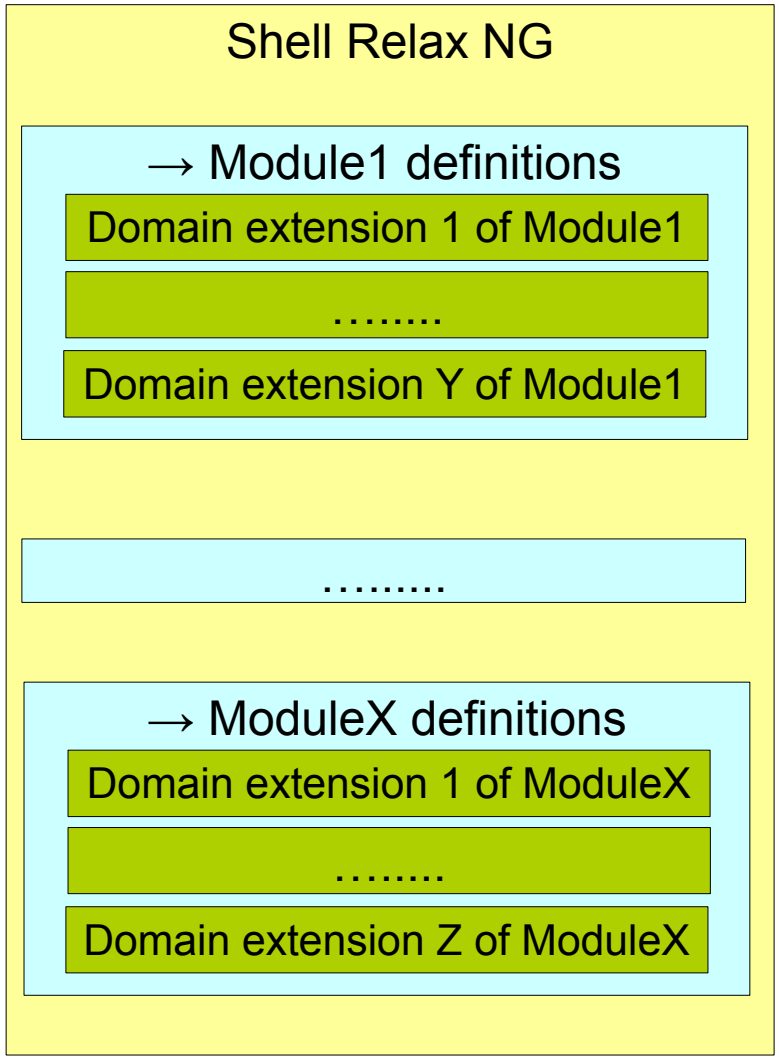


DITA implementation with DTDs



DITA DTDs vs Relax NG





Example shell

```
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0">
  <!-- Define the root elements -->
  <start>
    <ref name="bookmap.element"/>
  </start>

  <!-- The DITA domains attribute -->
  <define name="domains-atts" combine="interleave">
    <optional>
      <attribute name="domains"
        a:defaultValue="(map bookmap) (topic delay-d) (topic indexing-d) (map mapgroup-d) (topic xnal-d)
        (topic hi-d) (topic ut-d) (topic hazard-d) (topic abbrev-d) (topic ui-d) (topic pr-d) (topic sw-d)"/>
    </optional>
  </define>

  <!-- Include modules -->
  <include href="bookmap.mod.rng"/>
  <include href="../../base/rng/map.mod.rng"/>
  <include href="../../base/rng/delayResolutionDomain.mod.rng"/>
  <include href="../../base/rng/indexingDomain.mod.rng"/>
  <include href="../../base/rng/mapGroup.mod.rng"/>
  <include href="../../xnal/rng/xnalDomain.mod.rng"/>
  <include href="../../base/rng/highlightDomain.mod.rng"/>
  <include href="../../base/rng/utilitiesDomain.mod.rng"/>
  <include href="../../base/rng/hazardstatementDomain.mod.rng"/>
  <include href="../../technicalContent/rng/abbreviateDomain.mod.rng"/>
  <include href="../../technicalContent/rng/uiDomain.mod.rng"/>
  <include href="../../technicalContent/rng/programmingDomain.mod.rng"/>
  <include href="../../technicalContent/rng/softwareDomain.mod.rng"/>

  <!-- Define the any pattern to exclude elements with ID attributes
    from the wildcard and refer them expliceitely  -->
  <define name="any"> [22 lines]
</grammar>
```

Domain extensions

Domain extensions stay inside each domain file

hightlightDomain.mod.rng

```
<!-- Define domain extension patterns -->
<define name="hi-d-ph">
  <choice>
    <ref name="b.element"/>
    <ref name="i.element"/>
    <ref name="sup.element"/>
    <ref name="sub.element"/>
    <ref name="tt.element"/>
    <ref name="u.element"/>
  </choice>
</define>
<!-- Extend the patterns with the domain contribution -->
<define name="ph" combine="choice">
  <ref name="hi-d-ph"/>
</define>
```

Advantages of implementing DITA in Relax NG

- Simplified shell structure
- Validation of the domains attribute value

Validation of the domains attribute

The domains attribute contains values contributed from the included domains

Example:

(topic hi-d) (topic ut-d) (topic indexing-d) (topic hazard-d) (topic abbrev-d) (topic ui-d) (topic pr-d) (topic sw-d)

No validation error in case of DTDs

Validation though Schematron in case of Relax NG

Validation of the domains attribute

Missing (topic sw-d) from the domains attribute value gives this error:

```
49  <include href="../../technicalContent/rng/uiDomain.mod.rng"/>
50  <include href="../../technicalContent/rng/programmingDomain.mod.rng"/>
51  <include href="../../technicalContent/rng/softwareDomain.mod.rng"/>
52
53  <!-                                                  with ID attributes
54                                                        -->
55  <de
56
57
58
59
```

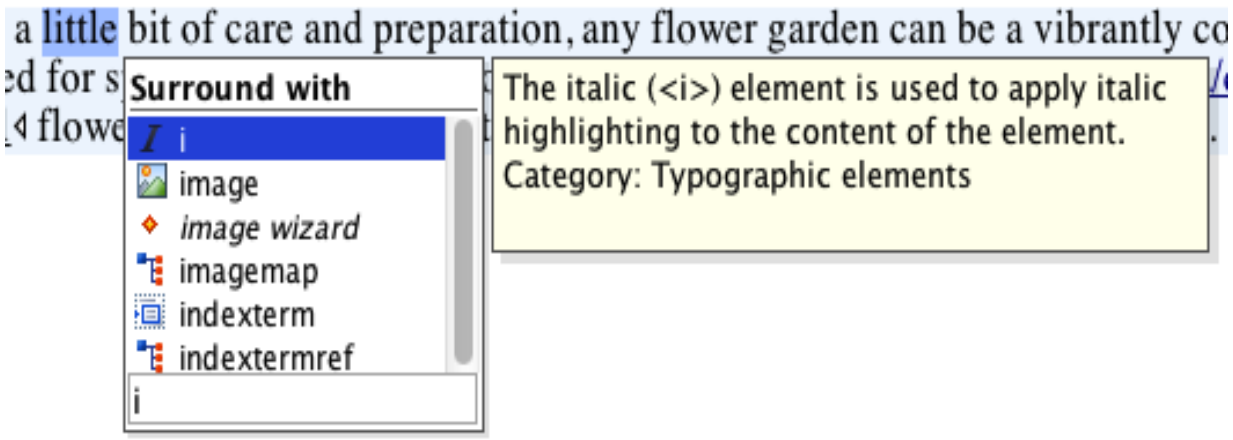
E [ISO Schematron (XSLT 2.0)] The domain values defined in an included domain file should be present in the domains attribute default value. (min(document(@href)/rng:grammar/rng:define[@name='domains-atts-value']/rng:value/contains($domains, .))) [assert]

Advantages of implementing DITA in Relax NG

- Simplified shell structure
- Validation of the domains attribute value
- Annotations for elements, attributes and values

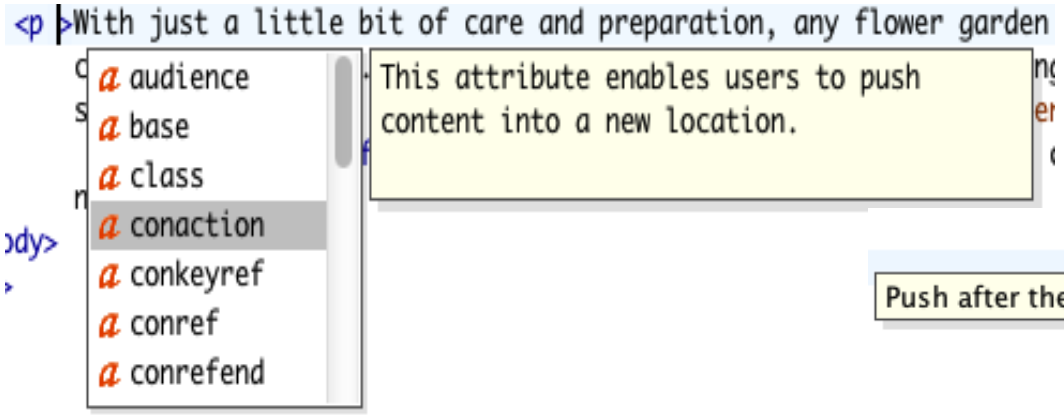
Annotations for elements

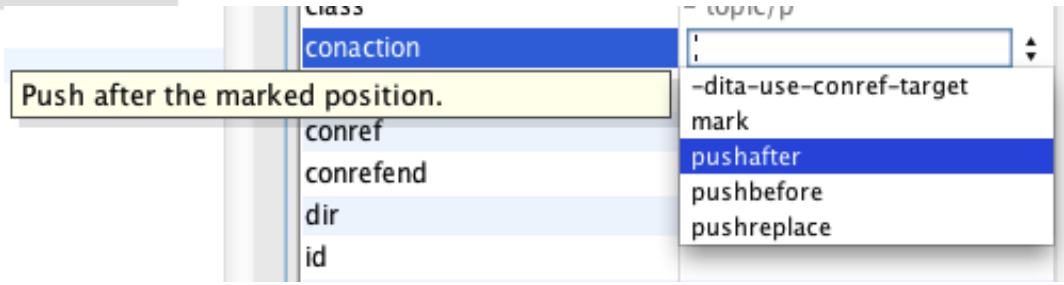
```
<element name="i">
  <a:documentation>The italic (&lt;i>) element is used to apply italic highlighting to the
    content of the element.
    Category: Typographic elements</a:documentation>
  <ref name="i.attlist"/>
  <ref name="i.content"/>
</element>
```



Annotations for attributes and values

```
<attribute name="conaction">
  <a:documentation>This attribute enables users to push content into a new location.</a:documentation>
  <choice>
    <value>mark</value>
    <a:documentation>Marks the reference position.</a:documentation>
    <value>pushafter</value>
    <a:documentation>Push after the marked position.</a:documentation>
      ...
  </choice>
</attribute>
```





Advantages of implementing DITA in Relax NG

- Simplified shell structure
- Validation of the domains attribute value
- Annotations for elements, attributes and values
- Possibility to embed Schematron rules

Embedded Schematron

Schematron rules can be embedded in Relax NG schemas

- This enabled adding additional checks to make sure the documents conform to the DITA specification
- The Schematron validation will pick the rules from all the Relax NG schemas referred from the shell, so the set of rules to be checked is assembled dynamically, based on what modules are included

Sample embedded Schematron

The Schematron rules stay with the element they check:

```
<element name="ul">
  <a:documentation>In an unordered list (&lt;ul>), the order of the list items is not
    significant. List items are typically styled on output with a "bullet" character, depending
    on nesting level.
    Category: Body elements</a:documentation>
  <sch:pattern name="atLeastTwoChildren">
    <sch:rule context="ul">
      <sch:assert test="count(*) > 1" role="warning">
        Please make sure you have at least 2 items for this list!
      </sch:assert>
    </sch:rule>
  </sch:pattern>
  <ref name="ul.attlist"/>
  <ref name="ul.content"/>
</element>
```

Schematron validation result

topic body p ul li

Flowers by Season

The various climatic changes that occur in cyclic pattern are termed as 'Seasons'. There are four general seasons occurring on Earth

- Spring

Summer, Autumn and Winter.

Flowers and seasons are intimately bound to each other. When Most

W [ISO Schematron (XSLT 2.0)] Please make sure you have at least 2 items for this list!

Text | Grid | Author

DITA-NG project

- Apache 2.0 license (same as DITA-OT)
- Available on Google Code

  http://code.google.com/p/dita-ng/
- Contains
  - DITA 1.2 Relax NG schemas (XML and Compact syntax)
  - XML Catalogs
  - Java code to transparently add default values to Xerces based on Relax NG schemas
  - DITA-OT plugin integration

DITA OT Plugin integration

Just a **plugin.xml** file with the following content:

```
<plugin id="com.oxygenxml.dita.relaxng">
  <feature extension="dita.specialization.catalog.relative"
    value="catalog.xml" type="file"/>
</plugin>
```

Integrating DITA-NG into DITA OT

Relax NG branch of DITA-OT

https://github.com/georgebina/dita-ot

Automatic Conversion to DTDs

Enables authoring DITA in Relax NG and deploy a DTD version for tools that cannot handle Relax NG

Two possible approaches

- Just get a DTD
- Get the modular DTSs

Just get a DTD

Use XProc to orchestrate processing:

- Convert the modular schema to a simplified version
- Apply fixes to allow DTD conversion
- Use Trang to convert to DTD
- Apply fixes on the DTD

Demo

The same support can be easily integrated also in an XML Authoring tool

Preview of oXygen with Relax NG support

- Create new DITA/Relax NG documents
- Edit DITA documents based on Relax NG
- Transform DITA/Relax NG maps to different formats
  - EPUB
  - WebHelp
  - PDF

New Relax NG schemas

DITA 1.3 proposals come with Relax NG schemas

Some work for the future

- Add documentation annotations for attributes and values
- Add embedded Schematron rules
- Include the Relax NG schemas in DITA-OT
- Show more specializations using Relax NG (DITA4Publishers maybe)

Part 1 Quick Overview

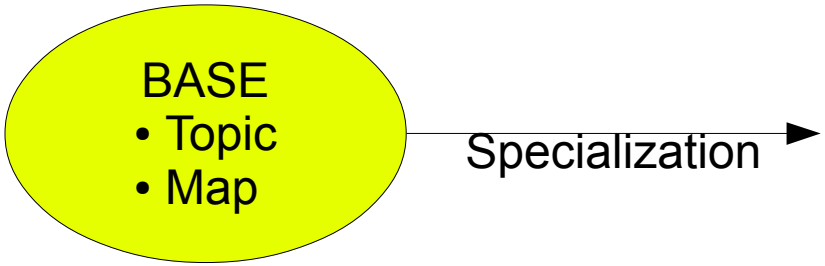
- Short introduction to DITA
- Short introduction to Relax NG
- Why DITA/Relax NG did not work before
- How we made DITA/Relax NG work
- Support for a:defaultValue annotations
- Creating the Relax NG schemas for DITA 1.2

DITA

DITA

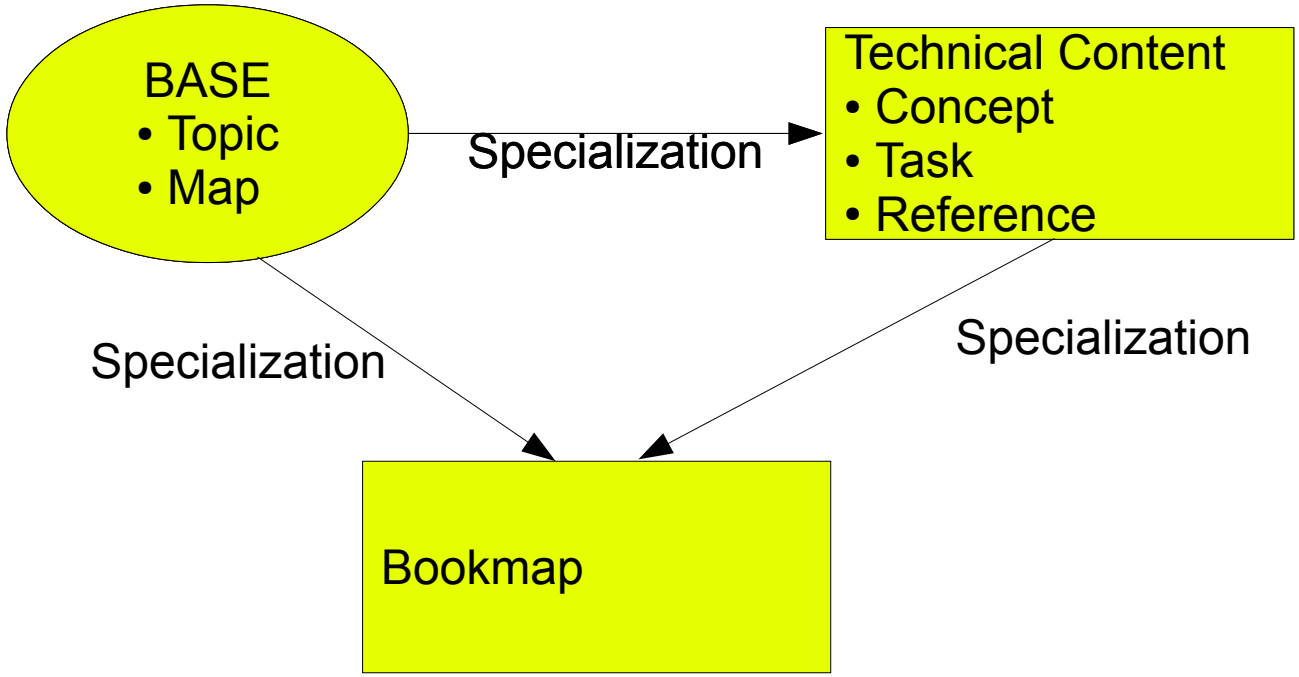
BASE
- Topic
- Map

DITA

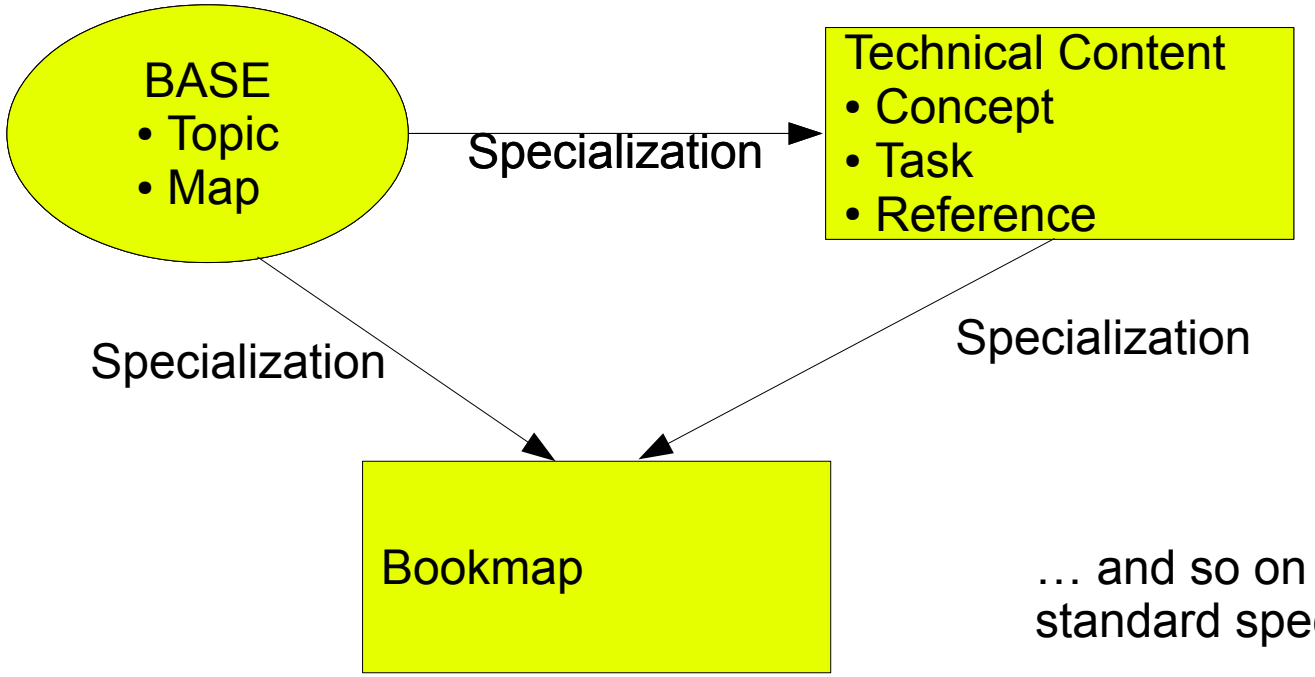


DITA

DITA

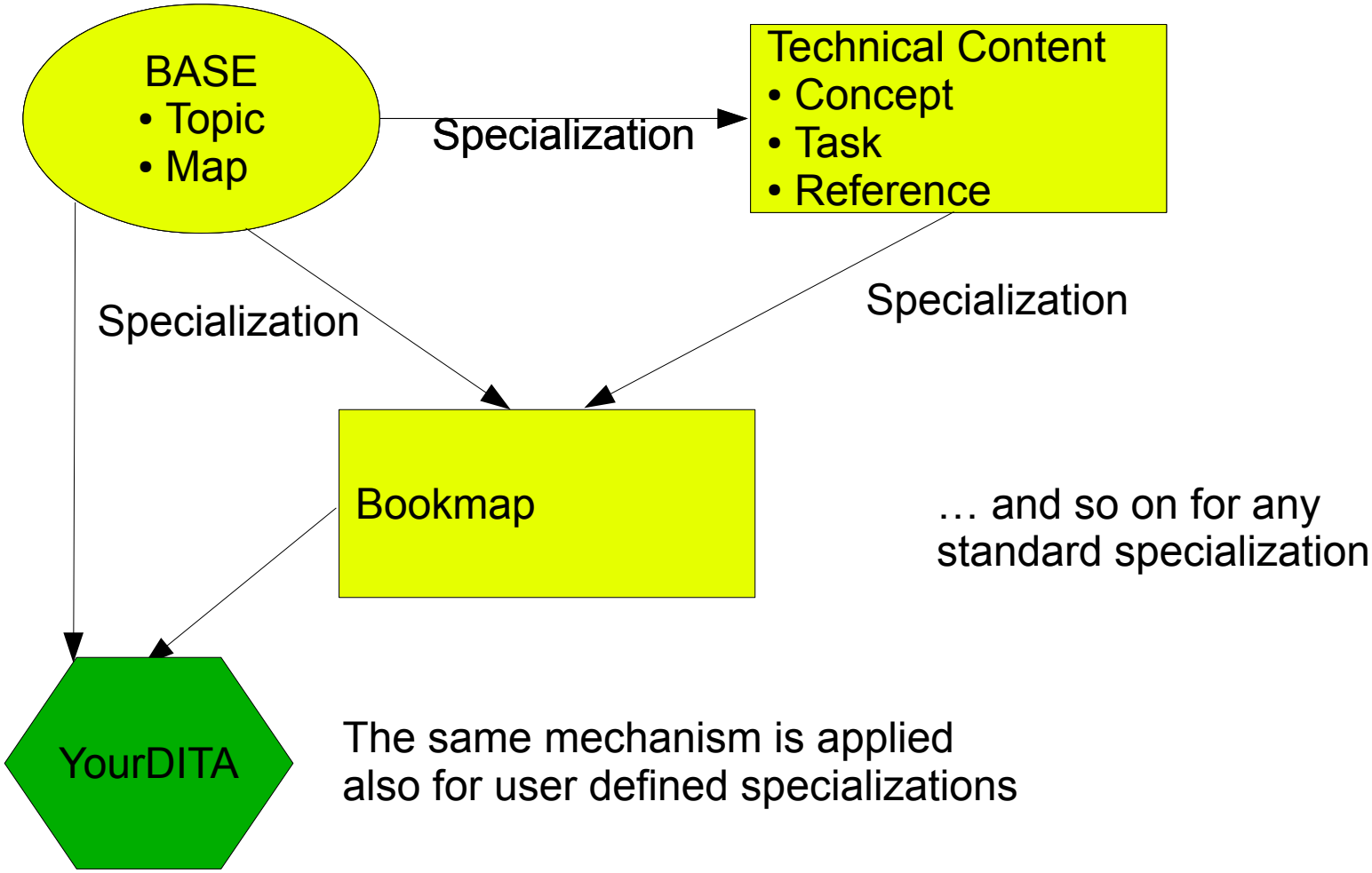


DITA



… and so on for any standard specialization

DITA



DITA samples - concept

```
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="toolsconcept" xml:lang="en-us">
 <title>Tools</title>
 <shortdesc>Invest in a good set of tools for doing all kinds of tasks around the house.</shortdesc>
 <conbody>
  <p>Useful tools include the following items:</p>
  <ul>
   <li>Hammer</li>
   <li>Screw driver set</li>
   <li>Wrench set</li>
   <li>Nails and screws</li>
   <li>Level</li>
   <li>Saws</li>
   <li>Drill</li>
   <li audience="expert">Air pressure gauge</li>
   <li>Spade</li>
   <li>Rake</li>
  </ul>
  <p>Keep your tools organized in a tool box which you can store in the garage.</p>
 </conbody>
 <related-links>
  <link href="toolbox.xml" format="dita" type="concept"/>
  <link href="../tasks/organizing.xml" format="dita" type="task"/>
 </related-links>
</concept>
```

DITA samples – learning overview

```
<!DOCTYPE learningOverview PUBLIC "-//OASIS//DTD DITA Learning Overview//EN"
"learningOverview.dtd">
<learningOverview id="overview">
 <title>Learning Overview topic</title>
 <shortdesc>Working outline for learning overview topic design</shortdesc>
 <learningOverviewbody>
  <lcObjectives>
   <title>Objectives</title>
   <lcObjectivesStem>When you complete this lesson, you'll know how to do the
    following:</lcObjectivesStem>
   <lcObjectivesGroup>
    <lcObjective>Creating a good learning overview topic.</lcObjective>
    <lcObjective>Identifying clear learning objectives.</lcObjective>
    <lcObjective>Adding good test items to assess knowledge gained.</lcObjective>
   </lcObjectivesGroup>
  </lcObjectives>
  <section>
   <title>Additional sections</title>
   <p>You can add additional sections to cover any other content you'd like to include in a learning
    Overview.</p>
  </section>
 </learningOverviewbody>
</learningOverview>
```

Understanding DITA

There is one key feature I want to focus on:

element names are not important, all processing relies on class attribute values

Understanding DITA

```
<lcObjectivesGroup>
 <lcObjective>Creating a good learning overview topic.</lcObjective>
 <lcObjective>Identifying clear learning objectives.</lcObjective>
 <lcObjective>Adding good test items to assess knowledge gained.</lcObjective>
</lcObjectivesGroup>
```

Understanding DITA

```
<lcObjectivesGroup>
 <lcObjective>Creating a good learning overview topic.</lcObjective>
 <lcObjective>Identifying clear learning objectives.</lcObjective>
 <lcObjective>Adding good test items to assess knowledge gained.</lcObjective>
</lcObjectivesGroup>
```

lcObjectivesGroup/@class default value

- topic/ul learningBase/lcObjectivesGroup

lcObjective/@class default value:

- topic/li learningBase/lcObjective

Understanding DITA

```
<lcObjectivesGroup>
 <lcObjective>Creating a good learning overview topic.</lcObjective>
 <lcObjective>Identifying clear learning objectives.</lcObjective>
 <lcObjective>Adding good test items to assess knowledge gained.</lcObjective>
</lcObjectivesGroup>
```

lcObjectivesGroup/@class default value

- topic/ul learningBase/lcObjectivesGroup

lcObjective/@class default value:

- topic/li learningBase/lcObjective

Understanding DITA

```
<lcObjectivesGroup>
 <lcObjective>Creating a good learning overview topic.</lcObjective>
 <lcObjective>Identifying clear learning objectives.</lcObjective>
 <lcObjective>Adding good test items to assess knowledge gained.</lcObjective>
</lcObjectivesGroup>
```

lcObjectivesGroup/@class default value

- topic/ul learningBase/lcObjectivesGroup

lcObjective/@class default value:

- topic/li learningBase/lcObjective

This will be processed by a tool that does not have special processing for the learning and training specialization as:

```
<ul>
 <li>Creating a good learning overview topic.</li>
 <li>Identifying clear learning objectives.</li>
 <li>Adding good test items to assess knowledge gained.</li>
</ul>
```

Relax NG

Relax NG

Defines the structure of a document in terms of patterns

Patterns all the way down

- The schema itself is a pattern
- No element or attribute “declarations”
- All definitions define patterns, thus all references are references to patterns

Understanding Relax NG validation

A regular expression pattern matching a string

Input: abcabc

Pattern: (a,b,c)+

(a,b,c)+	+	abcabc	→	(b, c), (a,b,c)*
(b, c), (a,b,c)*	+	abcabc	→	c, (a,b,c)*
c, (a,b,c)*	+	abcabc	→	(a,b,c)*
(a,b,c)*	+	abcabc	→	(b, c), (a,b,c)*
(b, c), (a,b,c)*	+	abcabc	→	c, (a,b,c)*
c, (a,b,c)*	+	abcabc	→	(a,b,c)*

(a,b,c)* is emptiable so the pattern (a, b, c)+ accepts the abcabc input!

Understanding Relax NG validation

The same ideas apply to Relax NG, the schema is the initial pattern and the input is the XML document split into tokens like below:

- Start element tag <element attribute="value">text</element>
- Attribute name <element attribute="value">text</element>
- Attribute value <element attribute="value">text</element>
- End of start tag <element attribute="value">text</element>
- Element value/text <element attribute="value">text</element>
- End element tag <element attribute="value">text</element>

Understanding Relax NG validation

- The initial schema pattern is successively derived against the input document tokens to obtain the pattern that needs to match the rest of the document
- A document is valid if the resulting pattern after the whole document was processed is optional (there is nothing required after the document was consumed)

Why DITA/Relax NG did not work

- DocBook 5 uses Relax NG
- TEI P5 uses Relax NG

- ...there should be some reasons why a Relax NG implementation of DITA was not available before...

Why DITA/Relax NG did not work

The DITA class attribute values are specified in the DTD/Schema as default values

(it would not be practical to request each author to enter those values for each element)

- Relax NG does not specify way to associate a document with a schema
- Relax NG itself provides no infoset contributions, that means no default attribute values

Schema association solutions

- application specific processing instructions
- application level user preferences
- W3C xml-model processing instruction

```
<?xml-model
        href="urn:oXygenxml:dita:rng:concept.rng"
        schematypens="http://relaxng.org/ns/structure/1.0"?>
```

Default values

There is a Relax NG DTD compatibility specification that defines an a:defaultValue annotation to specify default values:

```
<attribute name="test" a:defaultValue="value"/>
```

But there was no implementation available

Make DITA work with Relax NG

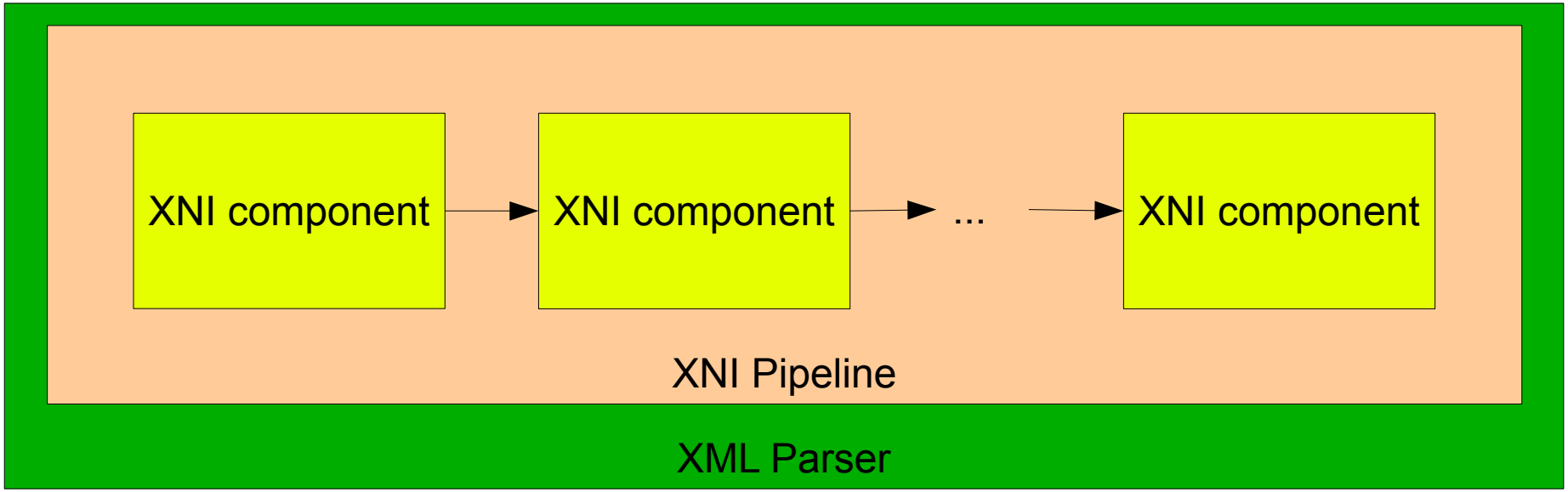
DITA-NG needed to:

- Provide support for default attribute values
- Update the processing workflow (the XML parser) to use the support for default values based on Relax NG
- Write the Relax NG schemas and specify the default values using a:defaultValue annotations

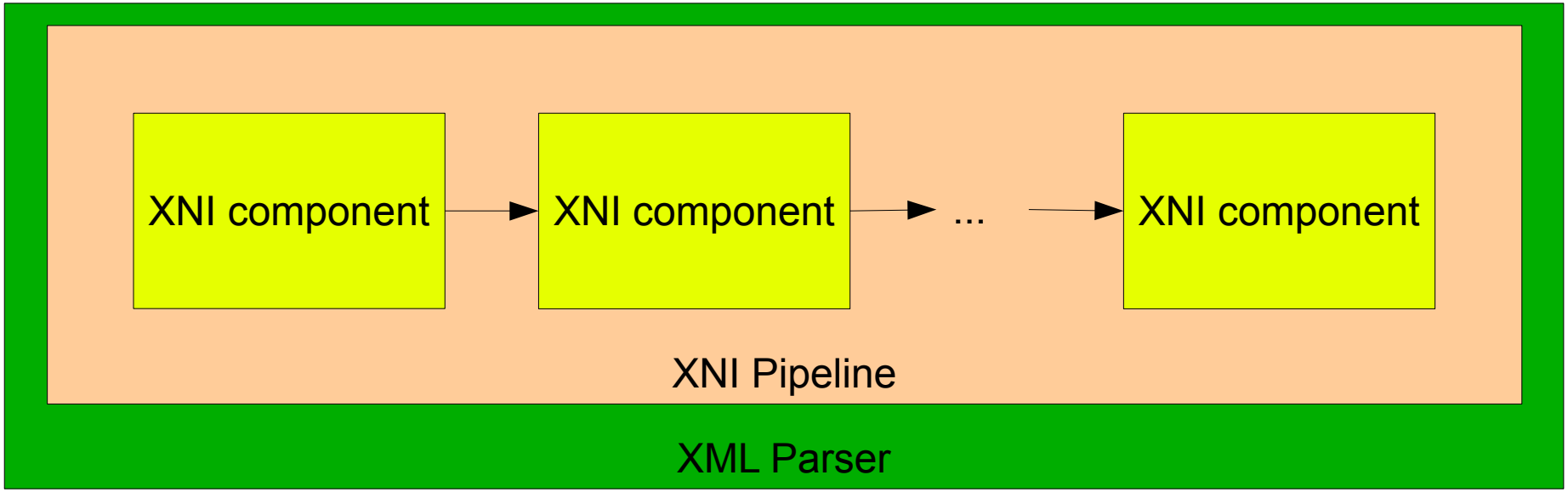
Make DITA work with Relax NG

- Change Jing (the main Relax NG processor) to store attribute default values annotations
- Implement a component that parses a schema and creates a map between attributes and their default values
- Update the XML parser (Xerces) to add default attribute values based on Relax NG schemas

Xerces parser configurations



Xerces parser configurations

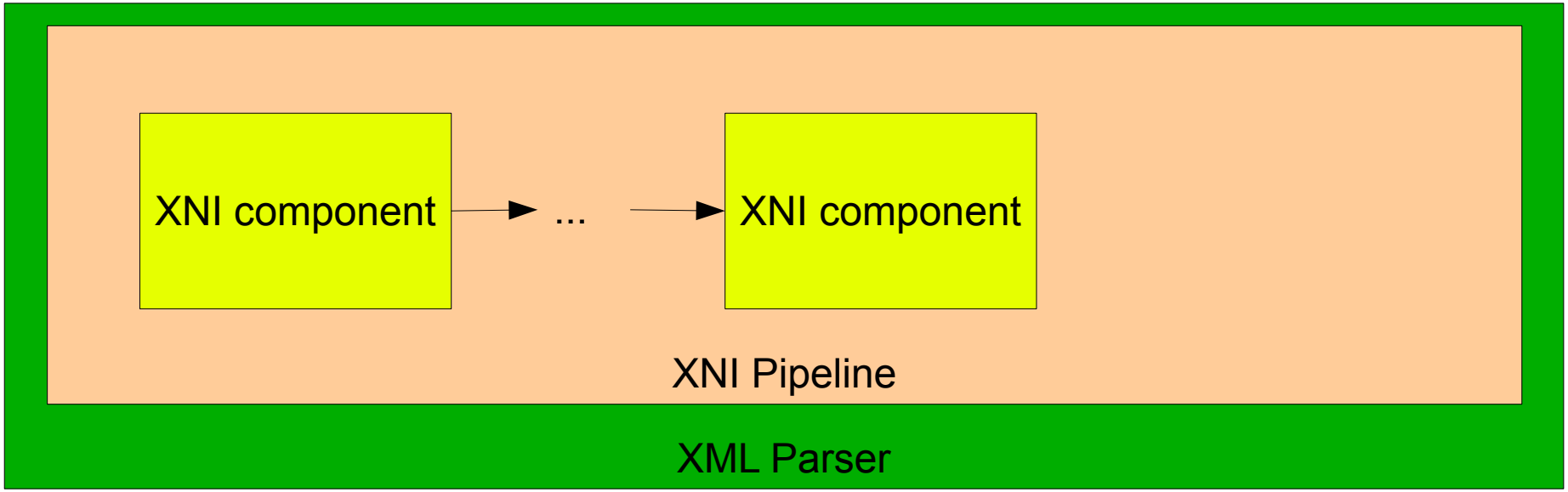


- Sample XNI components
  - Scanner
  - Validator
  - XInclude handler
- SAX-like events but with more information

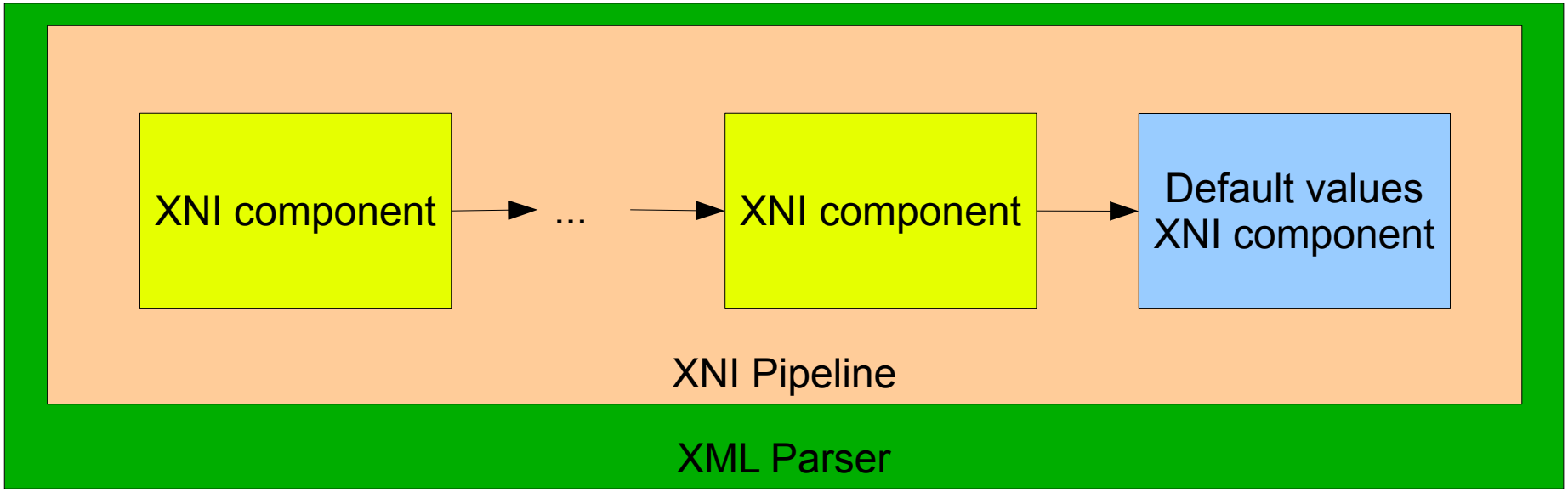
Default values XNI component

- Two roles / operation modes
  - Detect the associated schema and create the attributes to default values map
    - The current implementation looks for W3C xml-model schema association PIs.
  - On startElement events add default values when an attribute is not specified but it has a default value

Add the component to the pipeline



Add the component to the pipeline



We extend the Xerces parser configuration adding the default values component in the pipeline

Relax NG schemas for DITA 1.2

Conversion from DTD

Trang does a good job, but:

- the folder structure is not preserved
- multiple versions of the same file are obtained with slightly different content (due to overwritten entities)

so a manual merge is needed

Use Relax NG features

- Combine patterns
- Redefine patterns

- Allow domain contributions to stay inside the domain schema, thus simplifying the schema structure

Automated schema checks

- Use Schematron to check that the domains default value matches the included domains
- Additional Schematron checks (todo)
  - Class values follow the specified format

Conversion to RNC

- Trang can convert each schema but we hit again the Trang limitations wrt preserving folder structure
- Solution:

  an ant script that invokes Trang and performs the adjusting of the folder structure and fixes schema references

XML Catalogs

- Automatically generated from the Relax NG schemas with XSLT

Wrap-up

- You can use Relax NG and DITA with DITA-OT right now, it just takes 3-5 minutes to setup
- There are many advantages on using Relax NG for DITA
- The DITA-NG open source project contains
  - the code to enable default values based on Relax NG schemas
  - the DITA 1.2 schemas in Relax NG
  - the integration to DITA OT
  - conversion script to obtain DTDs from RelaxNG

Thank you!

Questions?

DITA-NG http://code.google.com/p/dita-ng/

<oXygen/> XML Editor
http://www.oxygenxml.com
george@oxygenxml.com
@georgebina