

# Schema-aware editing

George Bina  
george@oxygenxml.com  
Syncro Soft  
<oXygen/> XML Editor  
<http://www.oxygenxml.com>



# Summary

Introduction

Finding the schema

Schema information

Validation

Content completion

No tags – visual editing

Formatting (pretty print)

Conclusions

# Summary

Introduction

Finding the schema

Schema information

Validation

Content completion

No tags – visual editing

Formatting (pretty print)

Conclusions

# Summary

Introduction

Finding the schema

Schema information

**Validation**

**Content completion**

**No tags – visual editing**

**Formatting (pretty print)**

Conclusions

# Summary

Introduction

Finding the schema

Schema information

Validation

Content completion

No tags – visual editing

Formatting (pretty print)

**Conclusions**

# Summary

## Introduction

Finding the schema

Schema information

Validation

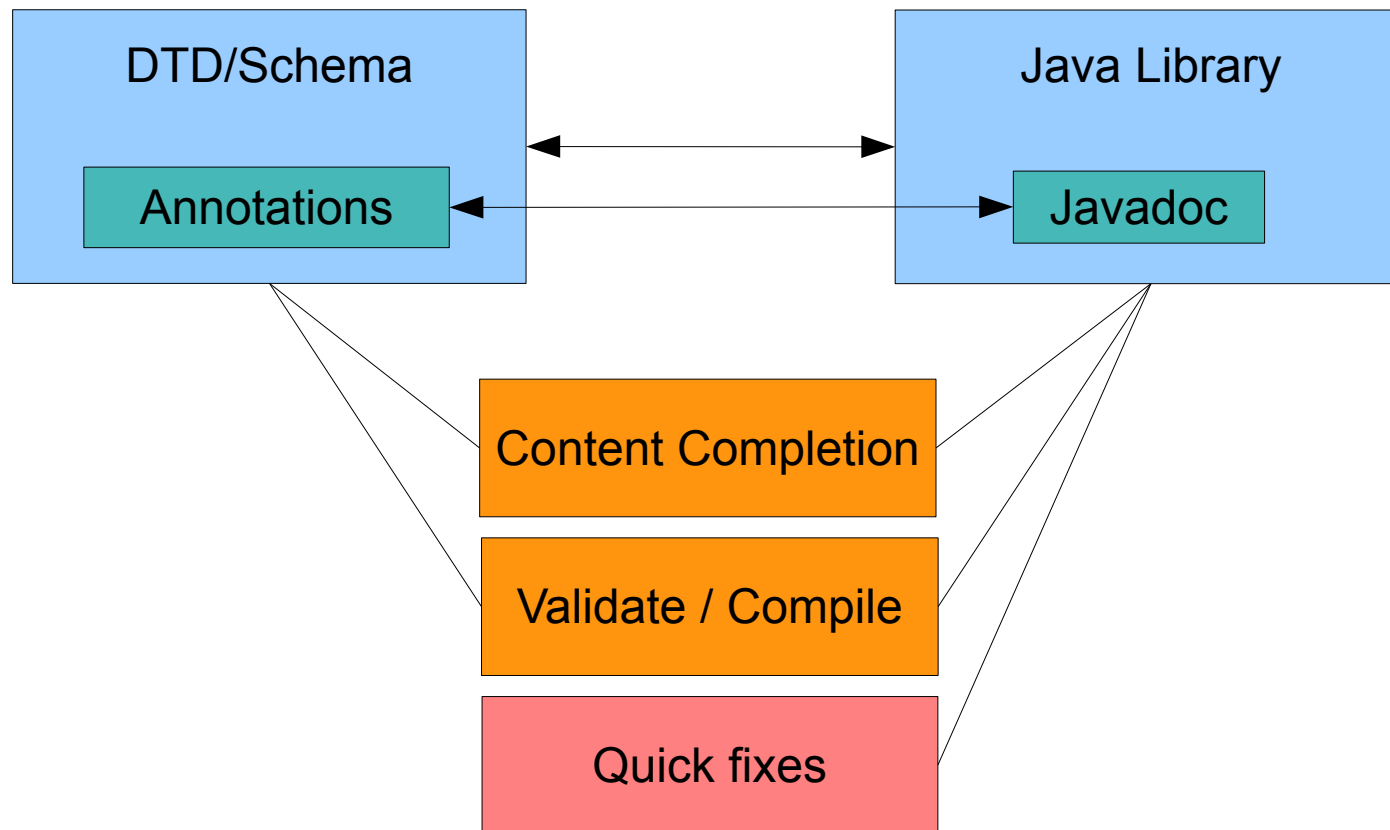
Content completion

No tags – visual editing

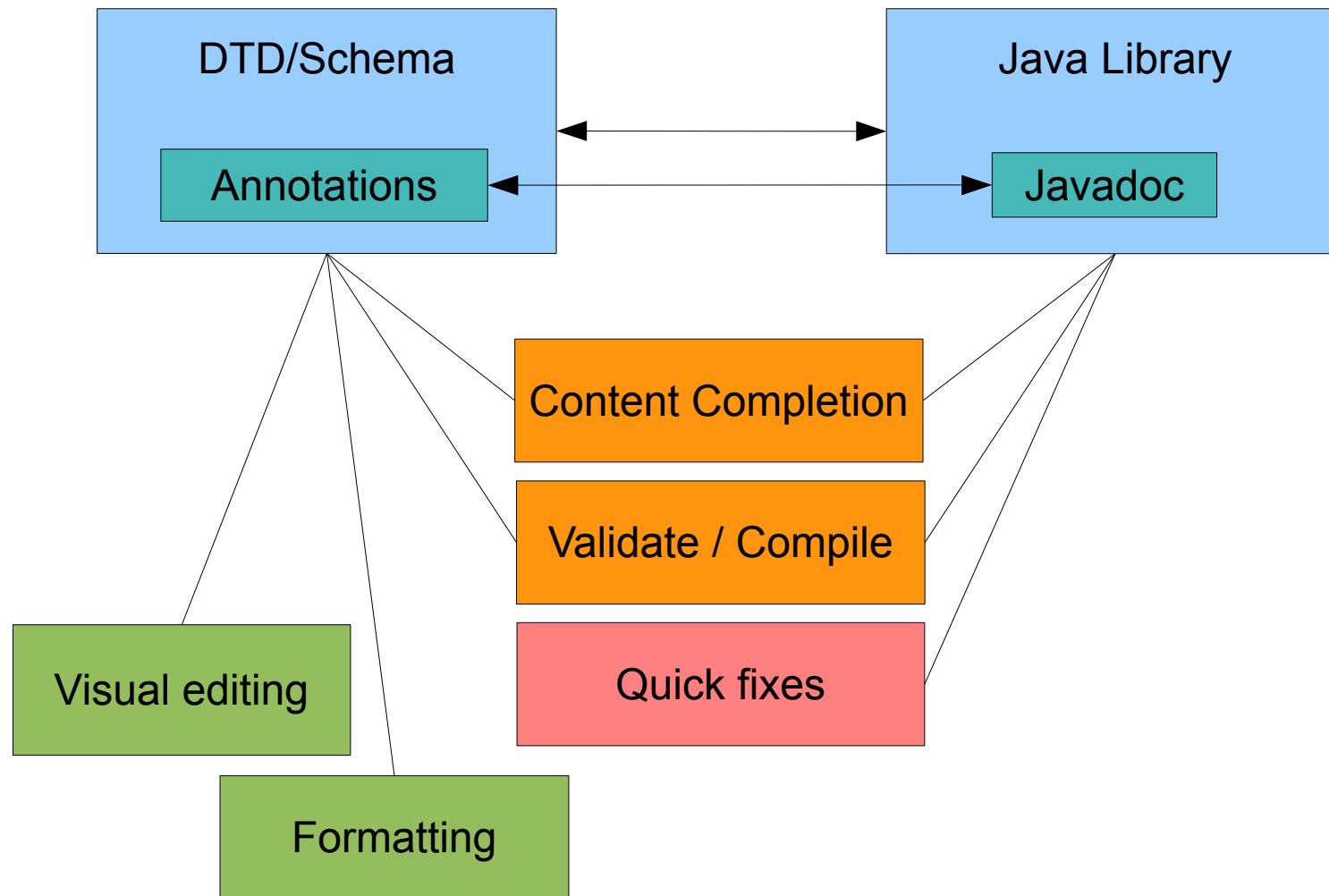
Formatting (pretty print)

Conclusions

# XML editing & Java development



# XML editing & Java development





# Why use schema information?

## Flexibility

- Support new XML languages (any language)
- React automatically to schema changes

# Why use schema information?

## Flexibility

- Support new XML languages (any language)
- React automatically to schema changes
  - DocBook customizations
  - DITA specializations
  - TEI profiles

# Summary

Introduction

**Finding the schema**

Schema information

Validation

Content completion

No tags – visual editing

Formatting (pretty print)

Conclusions

# Schema association

- DTDs
  - DOCTYPE declaration
- XML Schema
  - xsi:schemaLocation
  - xsi:noNamespaceSchemaLocation
- Other schema languages
  - Not specified!

# Alternative schema association

- Inside-document association
  - Application specific PIs

```
<?oxygen RNGSchema="http://www.docbook.org/xml/5.0/rng/docbook.rng"  
type="xml"?>
```

# Alternative schema association

- Inside-document association

- Application specific PIs

- ```
<?oxygen RNGSchema="http://www.docbook.org/xml/5.0/rng/docbook.rng" type="xml"?>
```

- The xml-model PI from W3C

- ```
<?xml-model href="http://www.docbook.org/xml/5.0/rng/docbook.rng"?>
```

# Alternative schema association

- Inside-document association

- Application specific PIs

```
<?oxygen RNGSchema="http://www.docbook.org/xml/5.0/rng/docbook.rng" type="xml"?>
```

- The xml-model PI from W3C

```
<?xml-model href="http://www.docbook.org/xml/5.0/rng/docbook.rng"?>
```

- Outside-document association

- Application settings
- Configuration files

# Alternative schema association

- Inside-document association

- Application specific PIs

```
<?oxygen RNGSchema="http://www.docbook.org/xml/5.0/rng/docbook.rng" type="xml"?>
```

- The xml-model PI from W3C

```
<?xml-model href="http://www.docbook.org/xml/5.0/rng/docbook.rng"?>
```

- Outside-document association

- Application settings
- Configuration files

Both inside and outside associations have their use cases



# Summary

Introduction

Finding the schema

**Schema information**

Validation

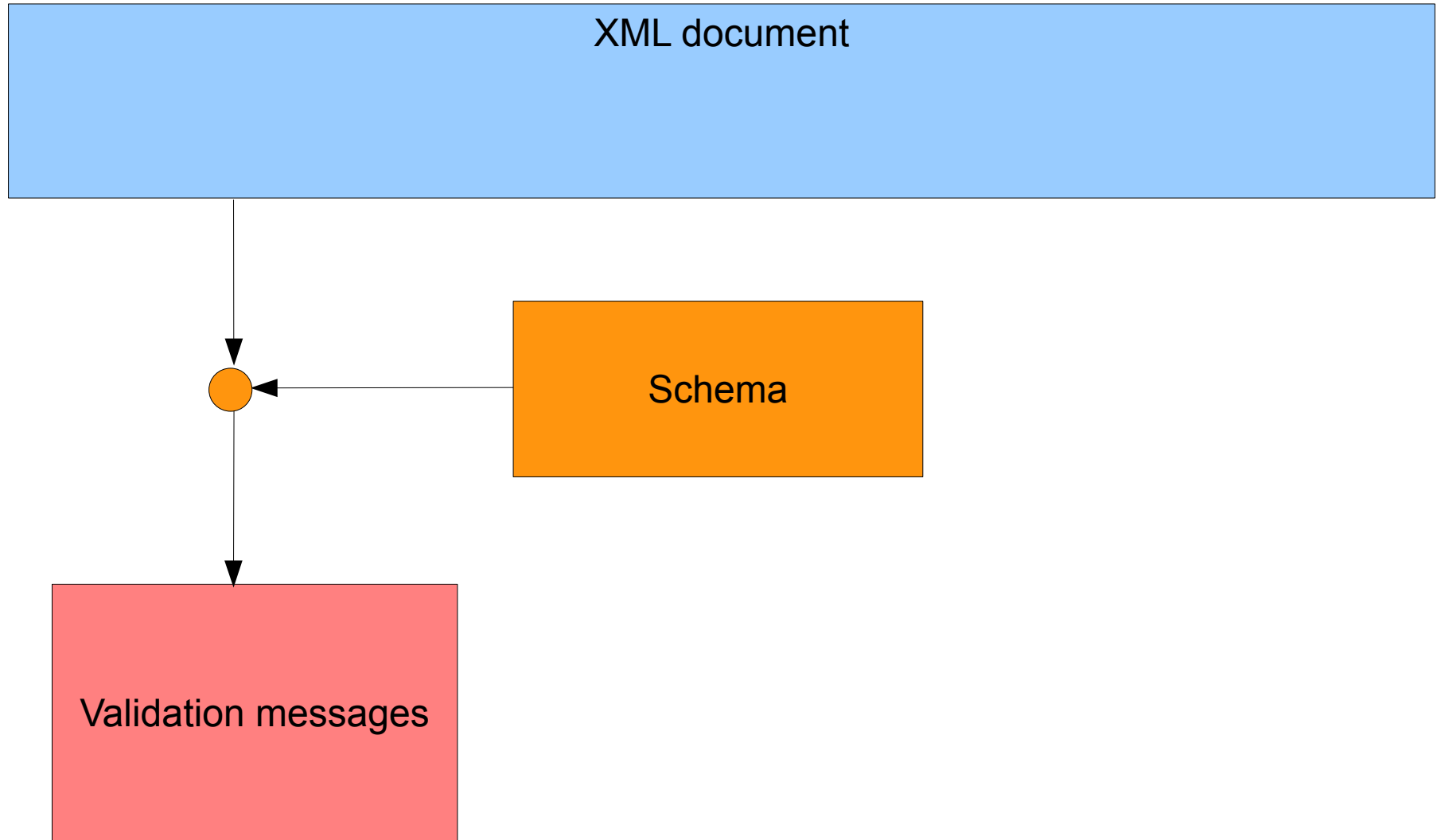
Content completion

No tags – visual editing

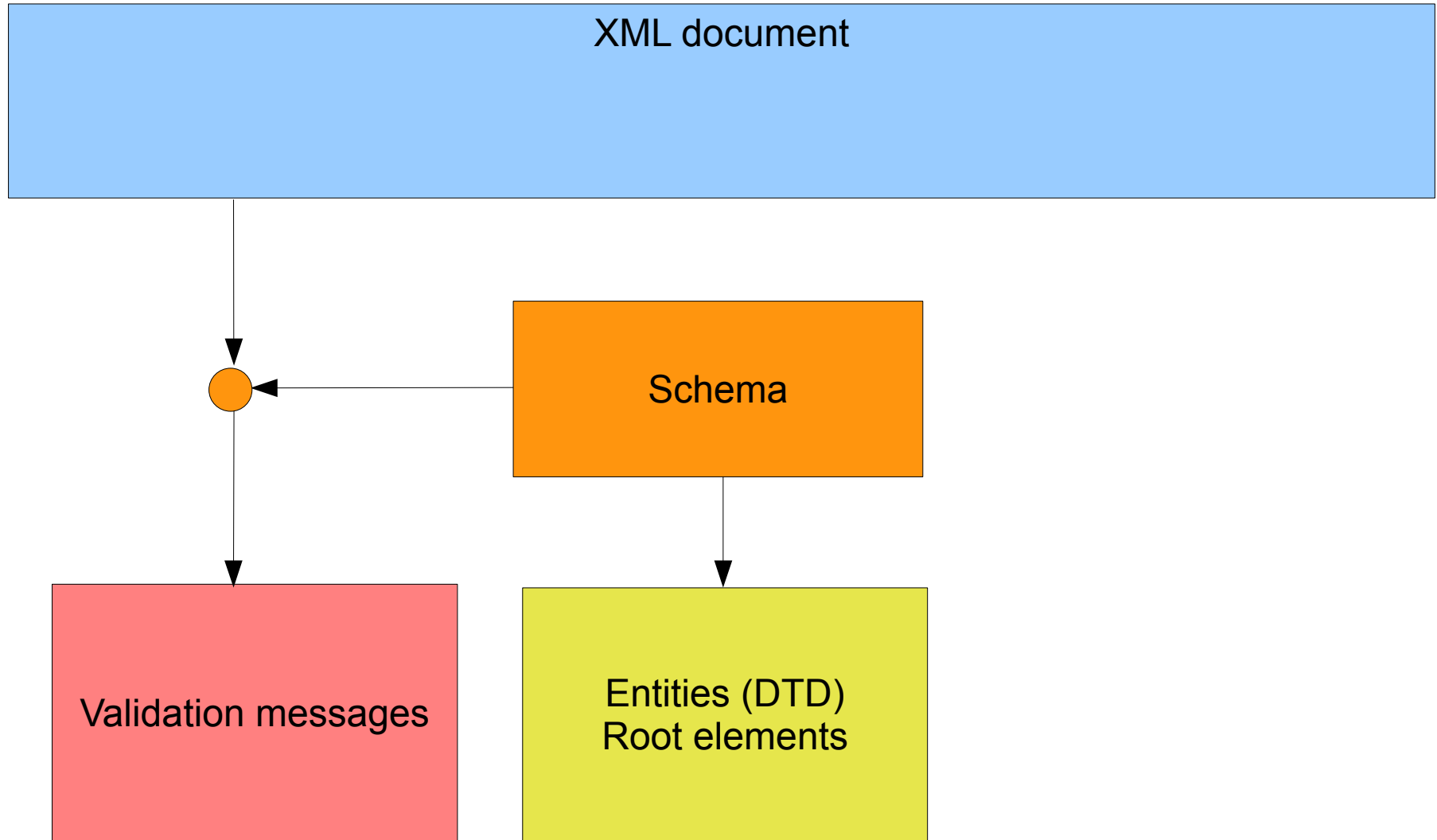
Formatting (pretty print)

Conclusions

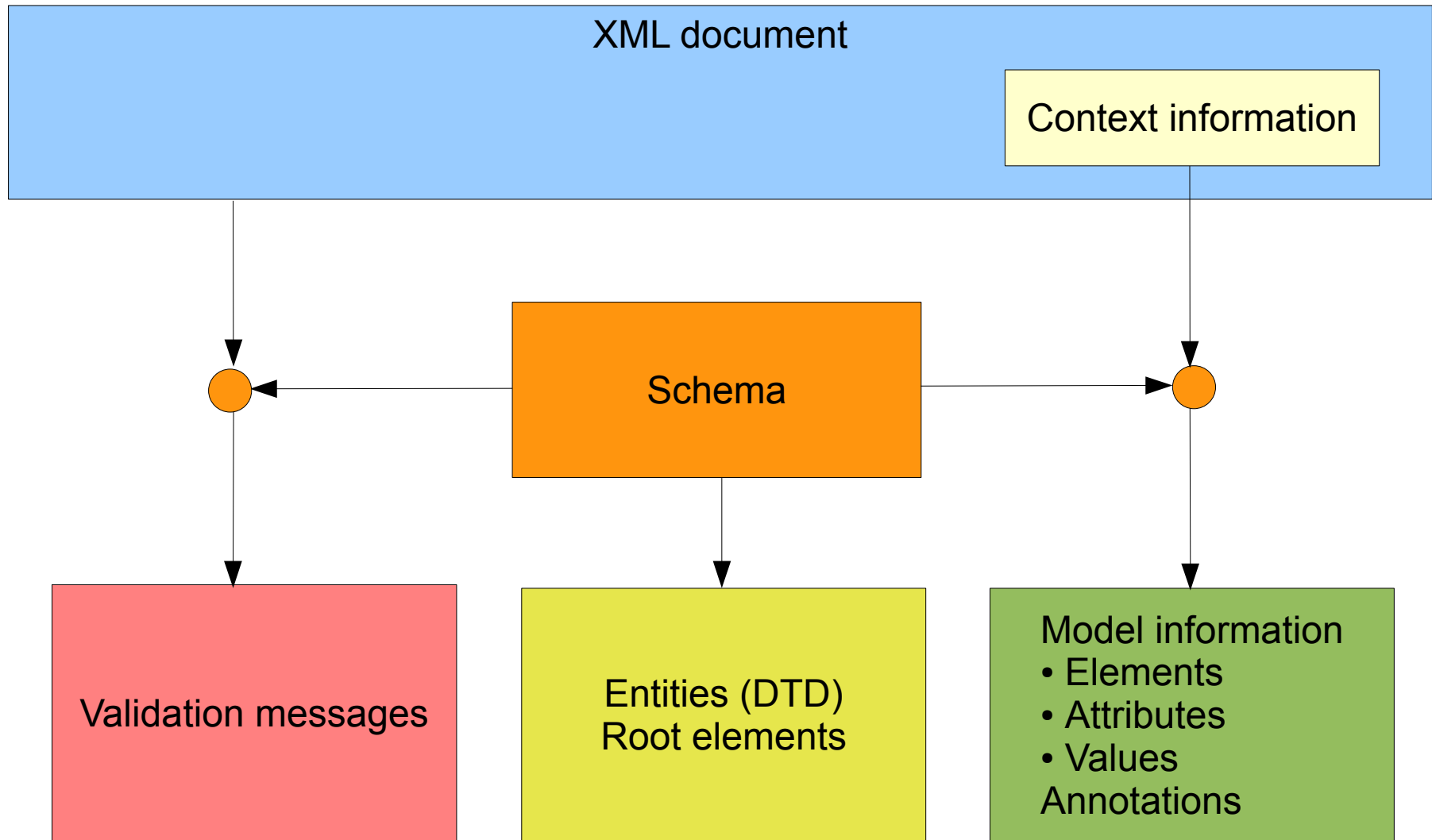
# Schema information



# Schema information



# Schema information



# Schema languages

## W3C

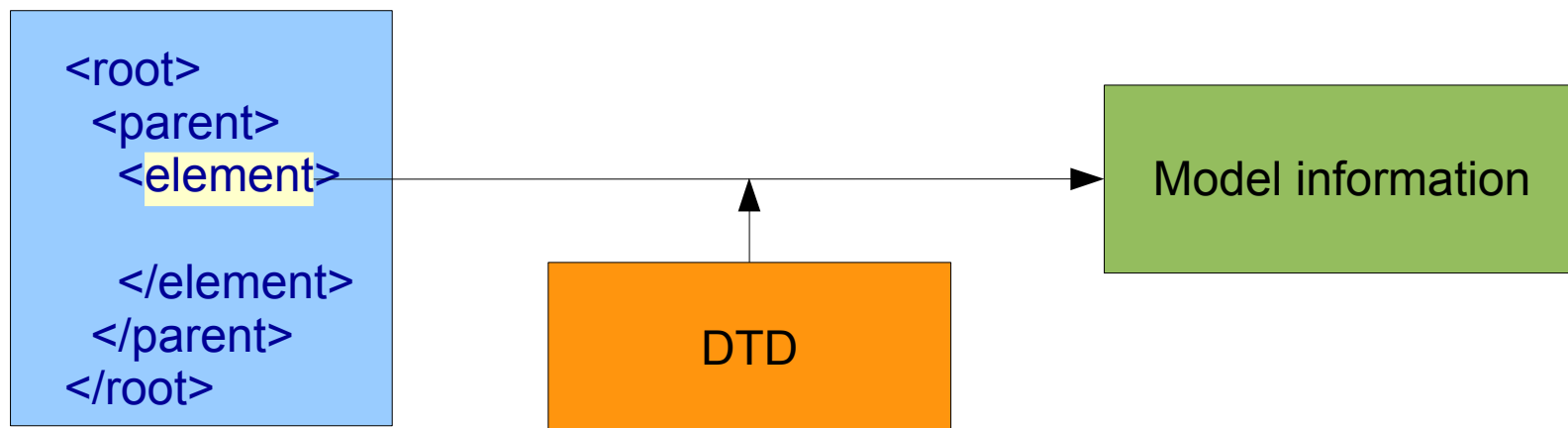
- DTD
- XML Schema

## ISO

- Relax NG
- Schematron
- NVDL

# DTD

- Most of the XML tools support DTDs
- Easy to support
  - Only global elements
  - No namespace support



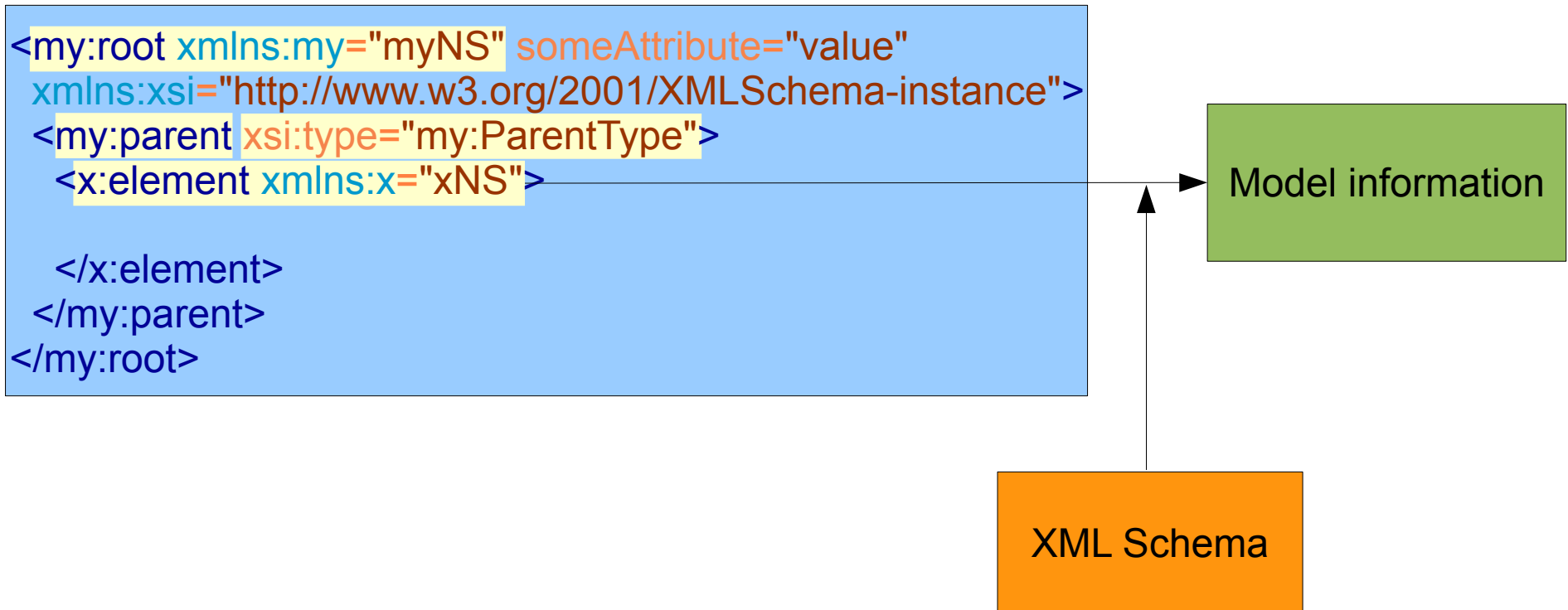
# XML Schema

- Allows **global** and **local** elements and attributes
- **xsi:type** attributes in the instance

```
<my:root xmlns:my="myNS" someAttribute="value"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <my:parent xsi:type="my:ParentType">
    <x:element xmlns:x="xNS">
      </x:element>
    </my:parent>
  </my:root>
```

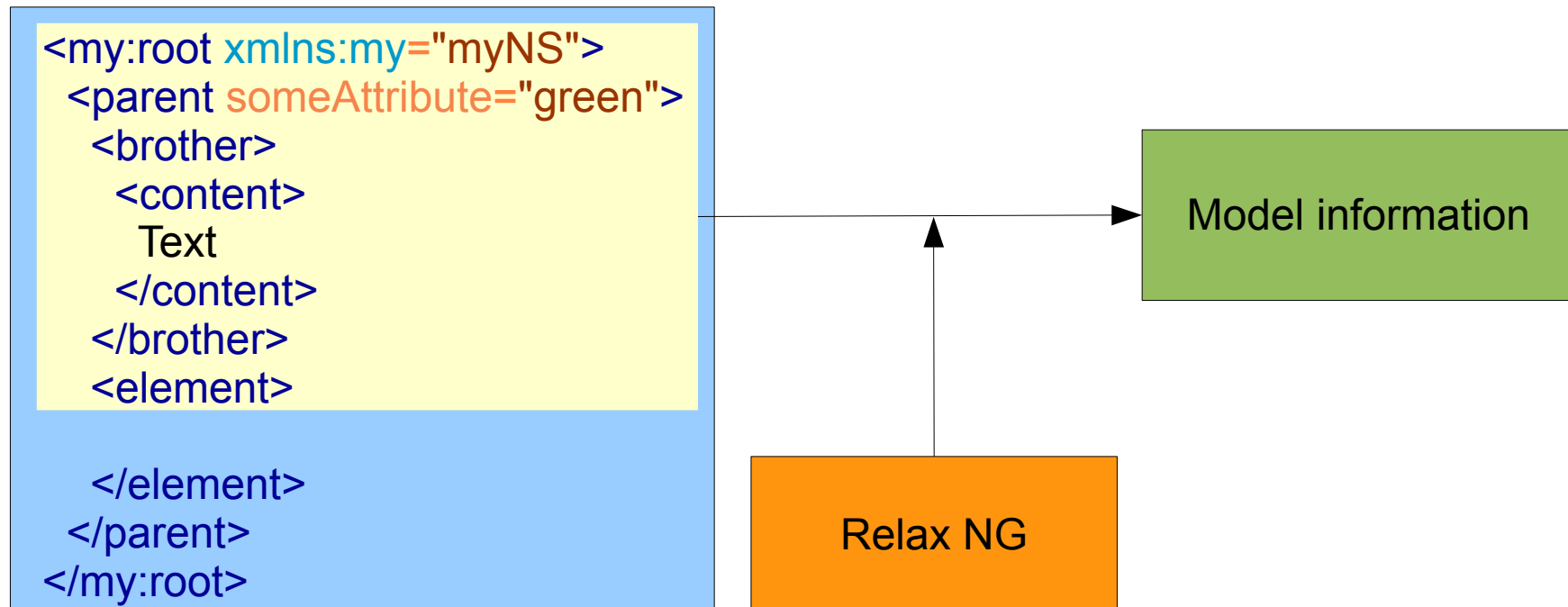
Model information

XML Schema



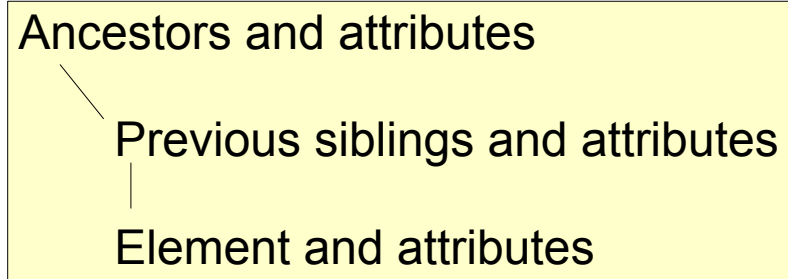
# Relax NG

- More flexible but more difficult to support
- Pattern based





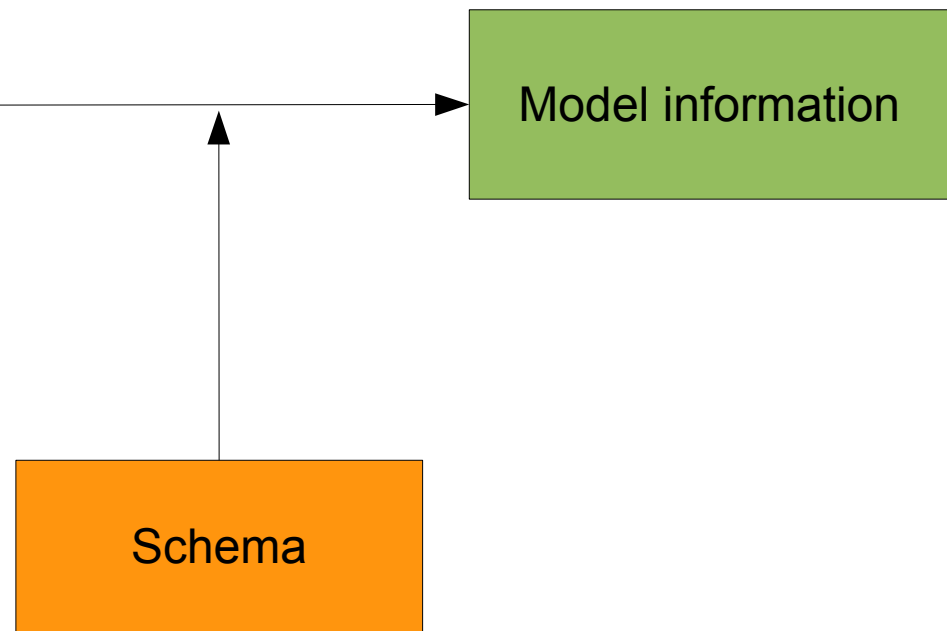
# Common context



```

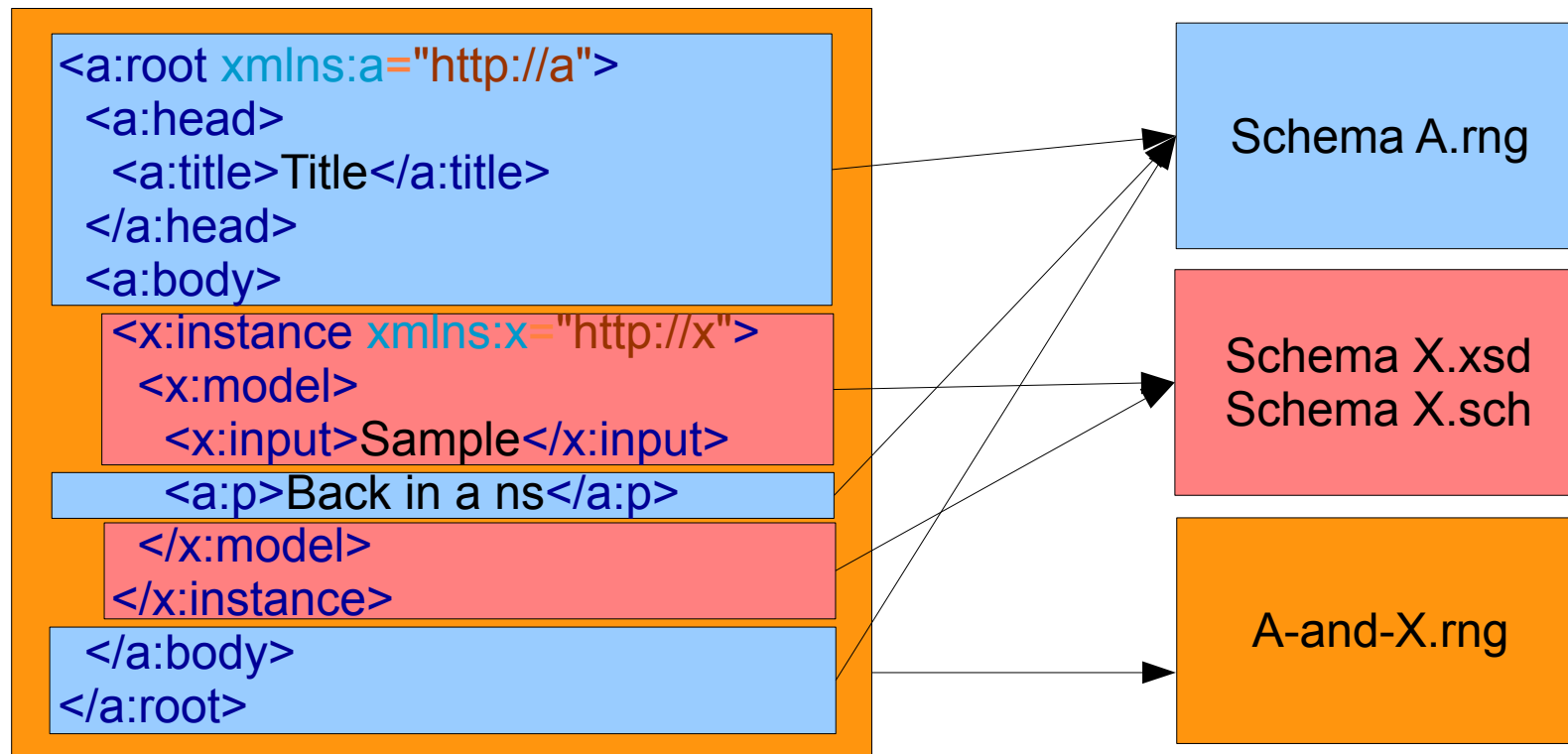
<my:root xmlns:my="myNS">
  <parentBrother>
  ...
  <parentBrother>
  <parent someAttribute="green">
    <brother att="test">
      <content>
        Text
      </content>
    </brother>
    <element>

    </element>
  </parent>
</my:root>
    
```



# NVDL

- Document partitions → validation candidates
- Multiple schemas
- Acts on context and delegates



# Schematron

- Limited support (validation)
- Rule based
- User defined error messages

## Example Schematron rules:

```
<rule context="person">  
  <assert test="name">A "person" element should contain "name".</assert>  
</rule>
```

```
<rule context="test">  
  <assert test="count(*)=@children">The number of elements inside the  
  "test" element should match the children attribute value.</assert>  
</rule>
```

# User defined error messages

```
<section>  
</section>
```

## DTD error

The content of element type "section" is incomplete, it must match "((title,subtitle?,titleabbrev?),(toc|lot|index|glossary|bibliography)\*,(((calloutlist|glosslist|bibliolist|itemizedlist|orderedlist|segmentedlist|simplelist|variablelist|caution|important|note|tip|warning|literallayout|programlisting|programlistingco|screen|screenco|screenshot|synopsis|cmdsynopsis|functsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|formalpara|para|simpara|address|blockquote|graphic|graphicco|mediaobject|mediaobjectco|informalequation|informalexample|informalfigure|informaltable|equation|example|figure|table|msgset|procedure|sidebar|qandaset|task|anchor|bridgehead|remark|highlights|abstract|authorblurb|epigraph|indexterm|beginpage)+,(refentry\*|section\*|simplesect\*))|refentry+|section+|simplesect+),(toc|lot|index|glossary|bibliography)\*)".

## Schematron

Your section does not have a title, please add “title” at the beginning of the section!

# Summary

Introduction

Finding the schema

Schema information

## Validation

Content completion

No tags – visual editing

Formatting (pretty print)

Conclusions

# Validation

## Continuous validation

- Applied automatically
- Inline error markers

## On-demand validation

- Applied on user request
- List of errors + inline error markers

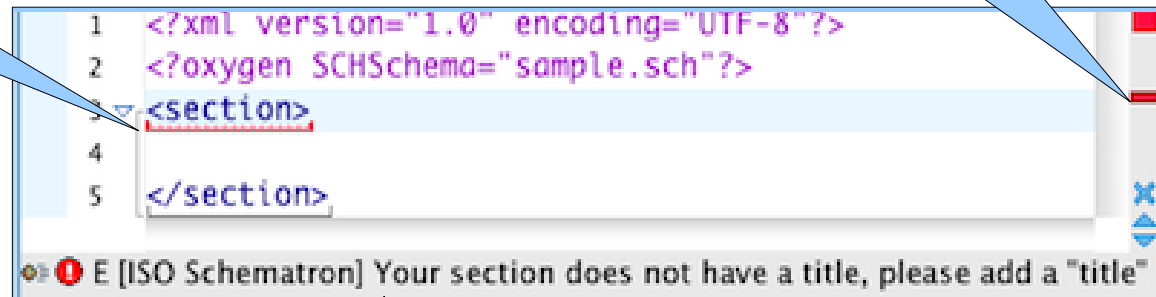
## Sample errors:

- Signal incomplete content  
element "article" incomplete; missing required element "title"
- Mark invalid content  
element "x" not allowed anywhere; expected element "title"

# Error markers

In document  
error markers


Errors marks in  
the overview ruler



The screenshot shows an XML editor window with the following content:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?oxygen SCHSchema="sample.sch"?>
3 <section>
4
5 </section>
```

The overview ruler on the right side of the editor shows a red vertical bar at the end of line 3, indicating an error. The status area at the bottom of the editor displays the following error message:

 E [ISO Schematron] Your section does not have a title, please add a "title"

Error message in the status area

# Quick fixes

- Automatic fix for specific errors
- Not yet implemented for XML (AFAIK)
- Example errors and possible fixes
  - element "article" incomplete; missing required element "title"  
→ Add "title" in "article"
  - element "x" not allowed anywhere; expected element "title"  
→ Remove "x" from the current element  
→ Replace "x" with "title"



# Summary

Introduction

Finding the schema

Schema information

Validation

**Content completion**

No tags – visual editing

Formatting (pretty print)

Conclusions

# Elements and attributes

- Required content
  - Required attributes
  - Required elements
- Handle undeclared namespaces

```
<?oxygen NVDLSchema="xhtml-xforms.nvdl"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Sample</title>
  </head>
  <body>
    <model#http://www.w3.org/2002/xforms
  </body>
</html>
```



# Values

Possible values are extracted from

- Enumerations
- Default and fixed constraints
- Types
  - boolean  $\rightarrow$  (true, false)
- The defines IDs in the document

# Entities

- DTD specific
- Schema for validation + DTD for entities

# Annotations as documentation

```
<element name="remark">  
  <a:documentation>A remark (or comment) intended for  
  presentation in a draft manuscript</a:documentation>  
  <ref name="db.remark.attlist"/>  
  <ref name="db._text"/>  
</element>
```

Selected  
element

Split sect2

- remark
- sidebar
- para
- tip
- address
- anchor
- annotation
- remark

A remark (or comment) intended for presentation in a draft manuscript

chosen the editor can insert not only the  
insert required attributes or required elements  
or request.

The annotation  
from the schema

# Comments for DTDs

## Defined

- Above
- Below

## Show

- All comments
- Matching a specific format

`<!--doc:`A remark (or comment) intended for presentation in a draft manuscript. The Remark element is designed to hold remarks, for example, editorial comments, that are useful while the document is in the draft stage, but are not intended for final publication. Remarks are available almost anywhere and have a particularly broad content model. Your processing system may or may not support either the use of comments everywhere they are allowed or the full generality of the Remark content model. Prior to version 4.0 of DocBook, this element was named Comment.-->

```
<!ELEMENT remark %ho; (%para.char.mix;)*  
          %remark.exclusion;>
```

# Summary

Introduction

Finding the schema

Schema information

Validation

Content completion

**No tags – visual editing**

Formatting (pretty print)

Conclusions

# No tags – visual editing

Targets also non-developer users

Use schema information to:

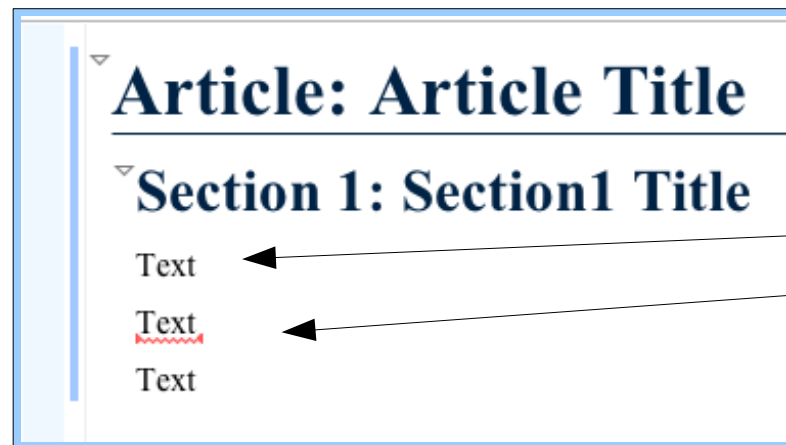
- Guide the user
- Restrict actions
- Apply recovery strategies



# Text in element only content

This is the most encountered error!

Not easy to spot if there are no tags



```
<article>
  <title>Article Title</title>
  <sect1>
    <title>Section1 Title</title>
    <para>Text</para>
    Text
    <para>Text</para>
  </sect1>
</article>
```

# Validation (post-editing)

The automatic validation signals the error

Unexpected character data "Text". The content of the parent element type is element only.

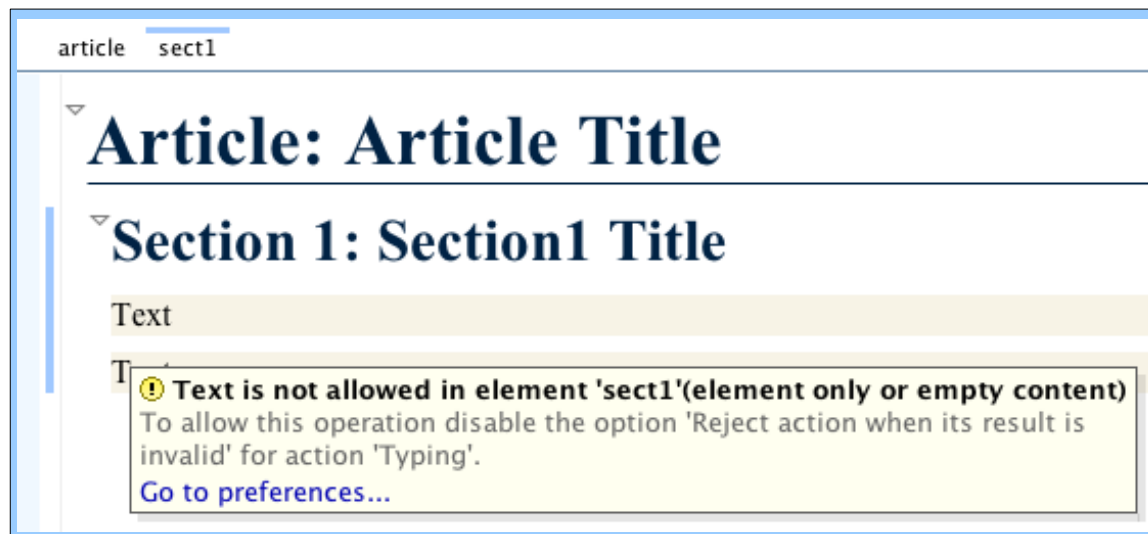
but:

- The user is exposed to the invalid situation
- Effort is required to correct the problem

A quick-fix may help

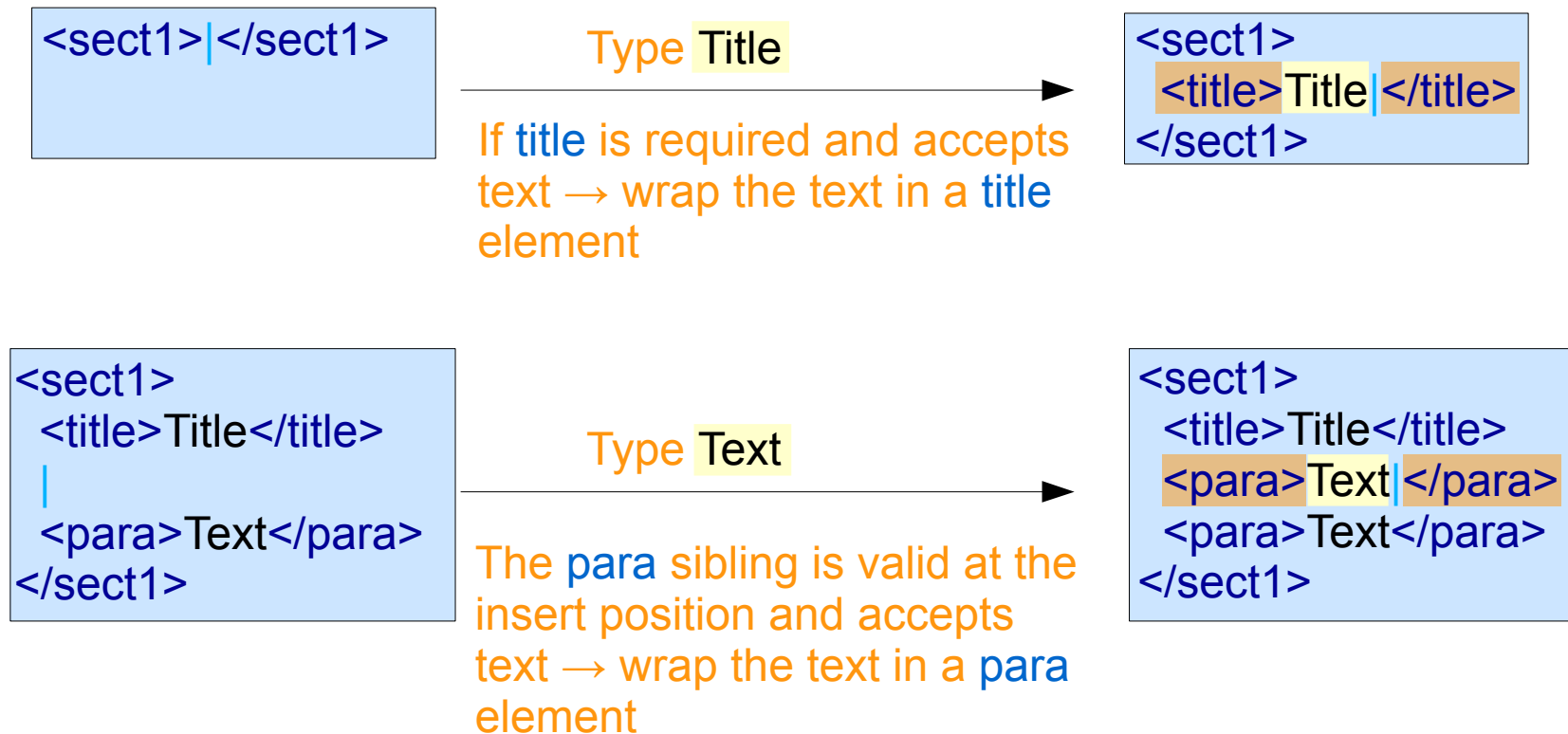
# Pre-validation

- Blocks the action if text is not allowed
- Informs the user
- No invalid state → no correction effort



# Automatic recovery

## Combine heuristics and schema information



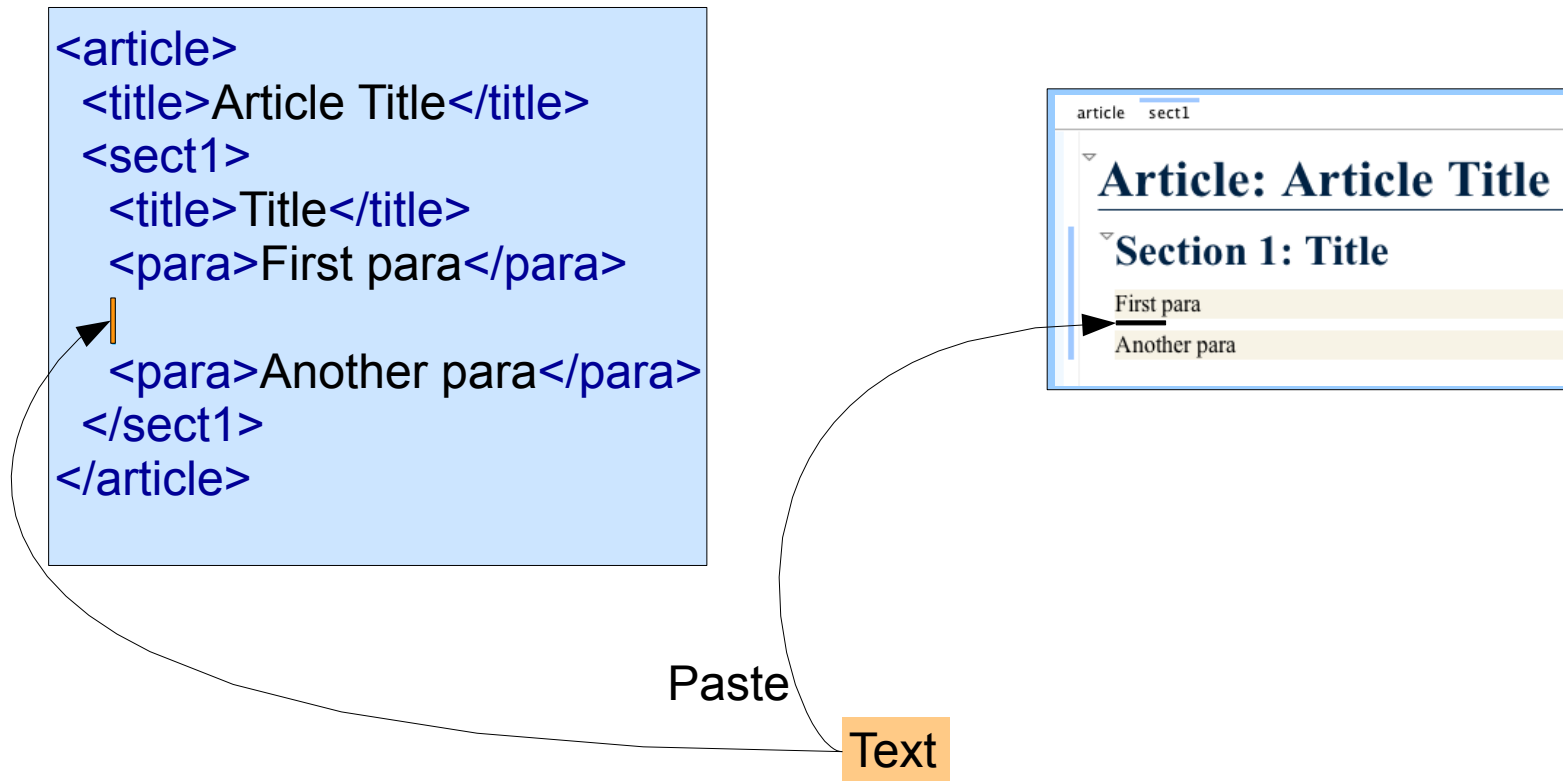
# Recovering strategies on paste

Triggered by pasting in a position where the pasted content is invalid

Sample recovery strategies:

- **Wrap** the content **in a sibling** node
- **Move** the content to a **near-by** location
- **Split** the node at the insert location **and insert** the content in an ancestor
- **Insert** only **the text** from the content
- **Wrap** the content **in a parent from** the content's **original location** (when that information is available)

# Paste text between two “para”



# Paste recovery proposals

```
<article>
  <title>Article Title</title>
  <sect1>
    <title>Title</title>
    <para>First para ③ </para>
    | ①
    <para>② Another para</para>
  </sect1>
</article>
```

wrap content in para ①

move content right in para ②

move content left in para ③

The screenshot shows a web editor interface with a document structure: "article" containing "sect1". The "sect1" element is expanded to show "Article: Article Title" and "Section 1: Title". Below the section title, there are two paragraphs: "First para" and "Another para". A "Validation Problem" dialog box is open, displaying the message: "The inserted content is not valid at the current position." Below this message, it asks "Insert the content:" and provides four options:

- inside a new **para** element
- inside the closest **para** element, to the right of the current position
- inside the closest **para** element, to the left of the current position
- at current position, even if invalid

The dialog box also includes "Cancel", "Options", and "OK" buttons.

# Paste text between para results

article sect1 para

Article: Article Title

Section 1: Title

First para

Text

Another para

This diagram shows a document structure with three levels: 'article', 'sect1', and 'para'. The 'para' level is expanded to show 'Article: Article Title', 'Section 1: Title', 'First para', 'Text', and 'Another para'. The 'Text' element is highlighted in yellow.

wrap content in para ①

article sect1 para

Article: Article Title

Section 1: Title

First para

TextAnother para

This diagram shows the same document structure as diagram 1, but the 'Text' element is now concatenated with 'Another para' to form 'TextAnother para'.

move content right in para ②

article sect1 para

Article: Article Title

Section 1: Title

First paraText

Another para

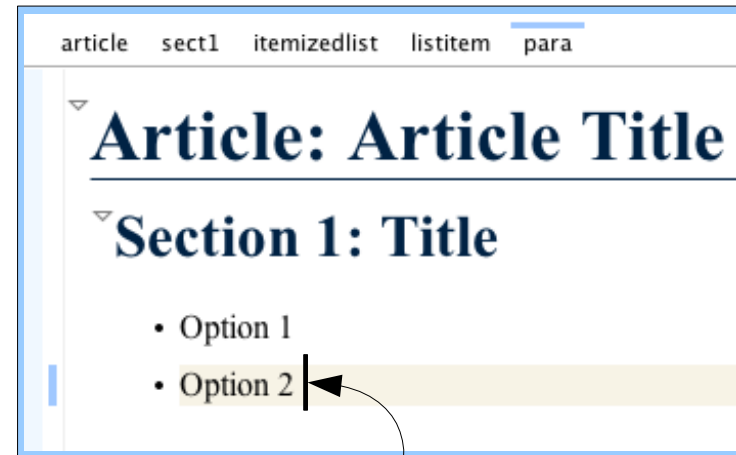
This diagram shows the same document structure as diagram 1, but the 'Text' element is now concatenated with 'First para' to form 'First paraText'.

move content left in para ③



# Paste a listitem in a listitem/para

```
<article>
  <title>Article Title</title>
  <sect1>
    <title>Title</title>
    <itemizedlist>
      <listitem>
        <para>Option 1</para>
      </listitem>
      <listitem>
        <para>Option 2</para>
      </listitem>
    </itemizedlist>
  </sect1>
</article>
```



Paste

```
<itemizedlist>
  <listitem>
    <para>Option 3</para>
  </listitem>
</itemizedlist>
```

# Paste recovery proposals

```

<article>
  <title>Article Title</title>
  <sect1>
    <title>Title</title>
    <itemizedlist>
      <listitem>
        <para>Option 1</para>
      </listitem>
      <listitem>
        <para>Option 2|2,3</para>
      </listitem> ①
    </itemizedlist>
  </sect1>
</article>

```

The screenshot shows a 'Validation Problem' dialog box with the following text:

**Section 1: Title**

- Option 1
- Option 2

The dialog box contains the following text:

The inserted content is not valid at the current position.

Insert the content:

- ① inside the closest **itemizedlist** element, to the right of the current position
- ② inside a new **itemizedlist** element
- ③ T at current position, as plain text
- ⚠ at current position, even if invalid

Buttons: Cancel, Options, OK

- ① move content right in **itemizedlist**
- ② wrap the content in the original **itemizedlist**
- ③ insert the text content, removing the markup

# Paste listitem in listitem/para results

article sect1 itemizedlist

▼ **Article: Article Title**

▼ **Section 1: Title**

- Option 1
- Option 2
- Option 3

1 move content right in itemizedlist

article sect1 itemizedlist listitem para itemizedlist

▼ **Article: Article Title**

▼ **Section 1: Title**

- Option 1
- Option 2
- Option 3

2 wrap the content in the original itemizedlist

article sect1 itemizedlist listitem para

▼ **Article: Article Title**

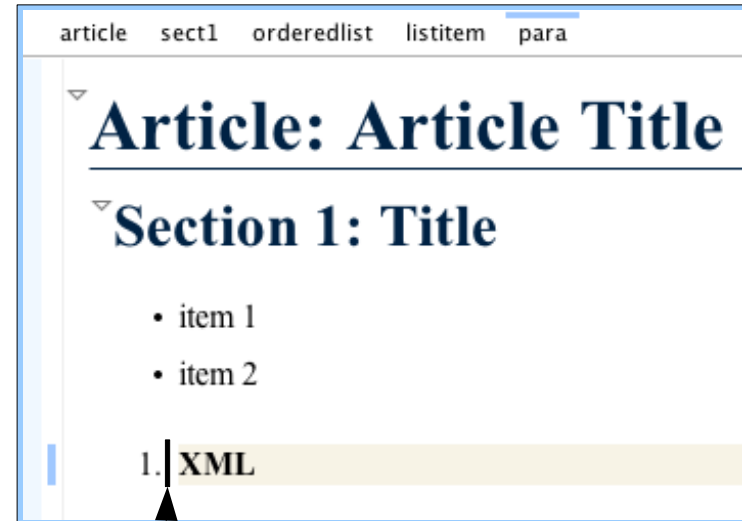
▼ **Section 1: Title**

- Option 1
- Option 2Option 3

3 insert the text content, removing the markup

# Recovery on delete

```
<article>
  <title>Article Title</title>
  <sect1>
    <title>Title</title>
    <itemizedlist>
      <listitem>
        <para>item 1</para>
      </listitem>
      <listitem>
        <para>item 2</para>
      </listitem>
    </itemizedlist>
    <orderedlist>
      <listitem>
        <para>|<abbrev>XML</abbrev></para>
      </listitem>
    </orderedlist>
  </sect1>
</article>
```




Press backspace

# Delete action

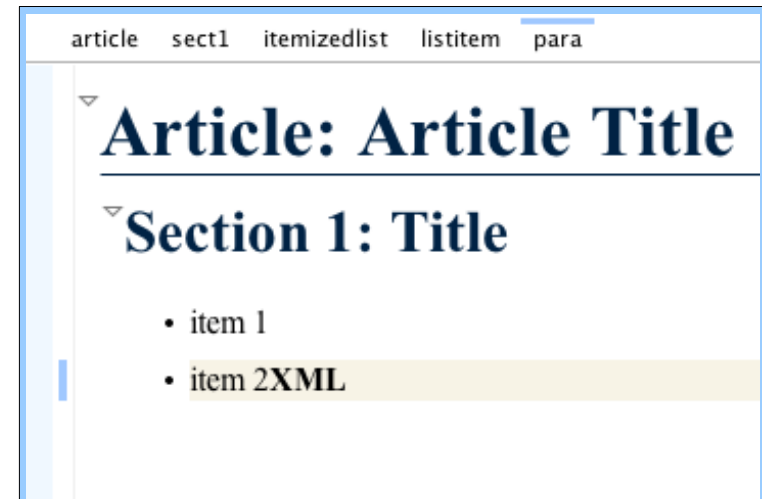
```
<article>
  <title>Article Title</title>
  <sect1>
    <title>Title</title>
    <itemizedlist>
      <listitem>
        <para>item 1</para>
      </listitem>
      <listitem>
        <para>item 2<img alt="green dot" data-bbox="255 515 275 535"/></para><img alt="red dot" data-bbox="355 515 375 535"/>
      </listitem><img alt="red dot" data-bbox="210 555 230 575"/>
    </itemizedlist><img alt="red dot" data-bbox="240 585 260 605"/>
    <orderedlist><img alt="red dot" data-bbox="225 620 245 640"/>
      <listitem><img alt="red dot" data-bbox="200 655 220 675"/>
        <para><img alt="orange highlight" data-bbox="190 680 430 715"/><abbrev>XML</abbrev></para>
      </listitem>
    </orderedlist>
  </sect1>
</article>
```

 Invalid location

 Valid location

# Recovery on delete result

```
<article>
  <title>Article Title</title>
  <sect1>
    <title>Title</title>
    <itemizedlist>
      <listitem>
        <para>item 1</para>
      </listitem>
      <listitem>
        <para>item 2<abbrev>XML</abbrev></para>
      </listitem>
    </itemizedlist>
  </sect1>
</article>
```



# Restrict editing actions

- Split element
  - if a next occurrence of that element is not allowed
- Join
  - if the content of the second element is not accepted in the first element
- Unwrap
  - if the content of the element is not accepted by the parent

# Summary

Introduction

Finding the schema

Schema information

Validation

Content completion

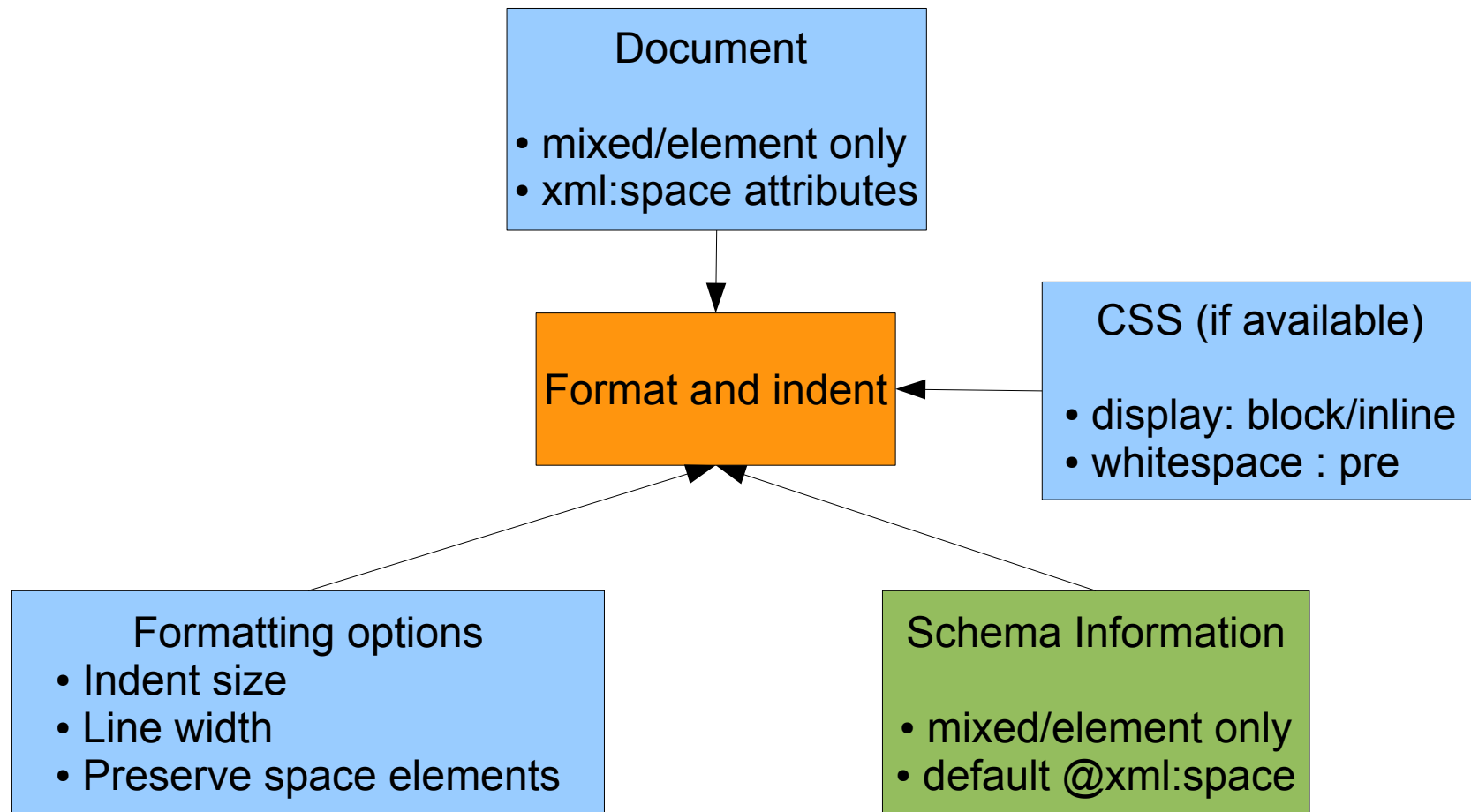
No tags – visual editing

**Formatting (pretty print)**

Conclusions



# Formatting action



# Some formatting examples

```
<data><x>1</x><y>3</y></data>
```

```
<data>  
  <x>1</x>  
  <y>3</y>  
</data>
```

XML document  
Element only content

XML document  
Mixed content

```
<p><b>G</b><i>eorge</i> is my first name.</p>
```

```
<p><b>G</b><i>eorge</i>  
is my first  
name.</p>
```

# Use content type information

```
<p><b>G</b><i>eorge</i></p>
```

XML document  
Element only content

Schema  
Mixed content

```
<p>  
  <b>G</b>  
  <i>eorge</i>  
</p>
```

```
<p><b>G</b><i>eorge</i></p>
```

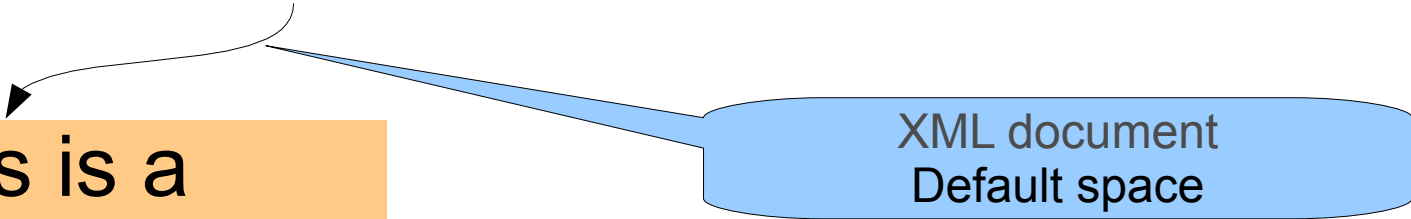
" **G** eorge " instead of "**George**"

If the schema specifies `p` as **mixed** the formatting will not add whitespace between `</b>` and `<i>`

# Other formatting examples

```
<data>This is a sample text</data>
```


```
<data>This is a  
sample text</data>
```



XML document  
Default space

```
<ex xml:space="preserve">This is a sample text</ex>
```

```
<ex xml:space="preserve">This is a sample text</ex>
```



XML document  
Preserve space

# Use xml:space information

```
<data>This is a sample text</data>
```

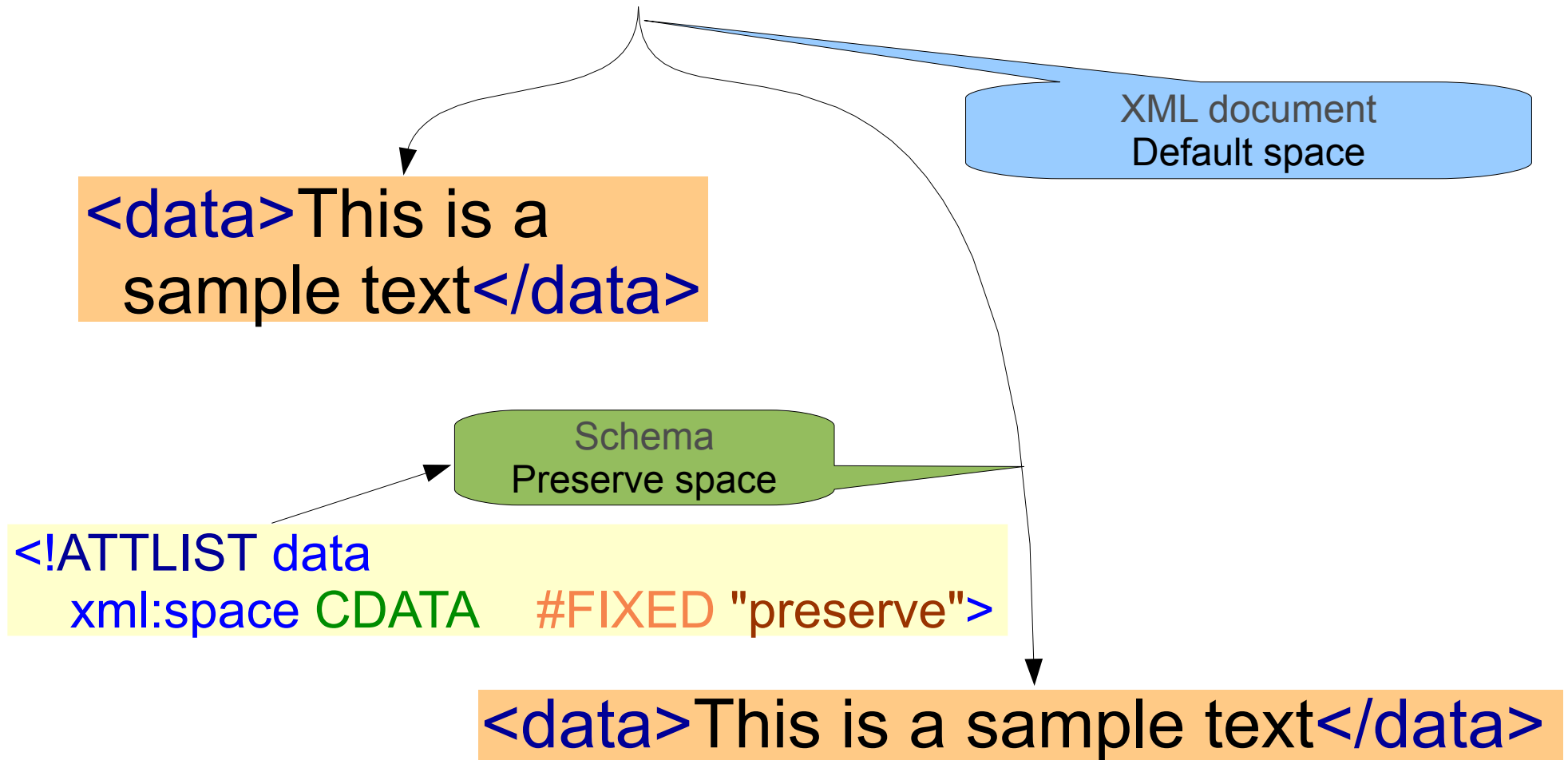
```
<data>This is a  
sample text</data>
```

Schema  
Preserve space

```
<!ATTLIST data  
xml:space CDATA #FIXED "preserve">
```

XML document  
Default space

```
<data>This is a sample text</data>
```



# Formatting information

- Default or fixed `xml:space` attribute
  - preserve
  - default
- The content type of an element
  - Element only
  - Mixed / text / data

These are practical issues (→ DocBook list)

# Summary

Introduction

Finding the schema

Schema information

Validation

Content completion

No tags – visual editing

Formatting (pretty print)

**Conclusions**

# Conclusions

- **Generic support**
- **Critical information** (in some cases - formatting)
- Clearly **improves the editing experience** (especially in the visual editing case)
- **Useful in many operations** (not only validation and content completion)
- Things **to do** (implement quick fixes, find new use cases)



# Thank you!

## Questions ?

George Bina  
george@oxygenxml.com  
Syncro Soft  
<oXygen/> XML Editor  
<http://www.oxygenxml.com>

