An introduction to XProc

George Bina

@georgebina
georgebina@oxygenxml.com

What is XProc?

W3C standard since May 2010

An XML Pipeline Language

Why XProc

Orchestrate complex XML transformations

- chain multiple transformation
- transform and validate results

Platform and language neutral

Tuned for XML processing

avoid serialization and parsing between steps → fast

Simple

XProc concepts

Pipelines

- Steps
  - Atomic
  - Compound
  - Multi-container
- Ports
  - Input
  - Output
- Connections/bindings

Steps

- Atomic - single operations
  - filter
  - wrap
  - XSLT
  - etc.
- Compound - contains a sub-pipeline
  - group
  - for each
  - etc.
- Multi-container
  - try/catch
  - choose

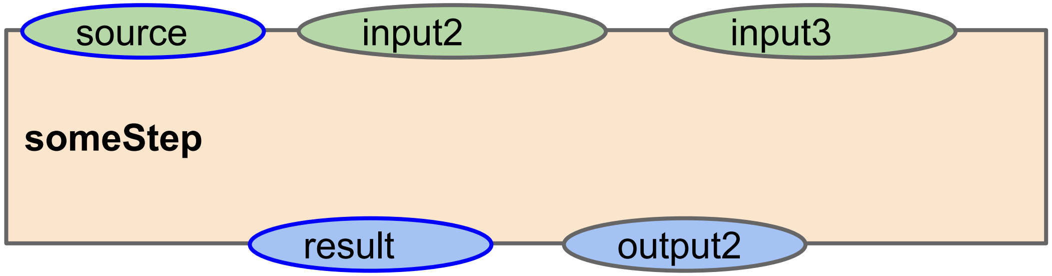
Atomic steps

p:add-attribute
p:add-xml-base
p:compare
p:count
p:delete
p:directory-list
p:error
p:escape-markup
p:filter
p:http-request
p:identity
p:insert
p:label-elements
p:load
p:make-absolute-uris
p:namespace-rename
p:pack
p:parameters
p:rename
p:replace
p:set-attributes
p:sink
p:split-sequence
p:store
p:string-replace
p:unescape-markup
p:unwrap
p:wrap
p:wrap-sequence
p:xinclude
p:xslt
p:exec
p:hash
p:uuid
p:validate-with-relax-ng
p:validate-with-schematron
p:validate-with-xml-schema
p:www-form-urldecode
p:www-form-urlencode
p:xquery

Ports

Input

Output



One primary input and one primary output port!

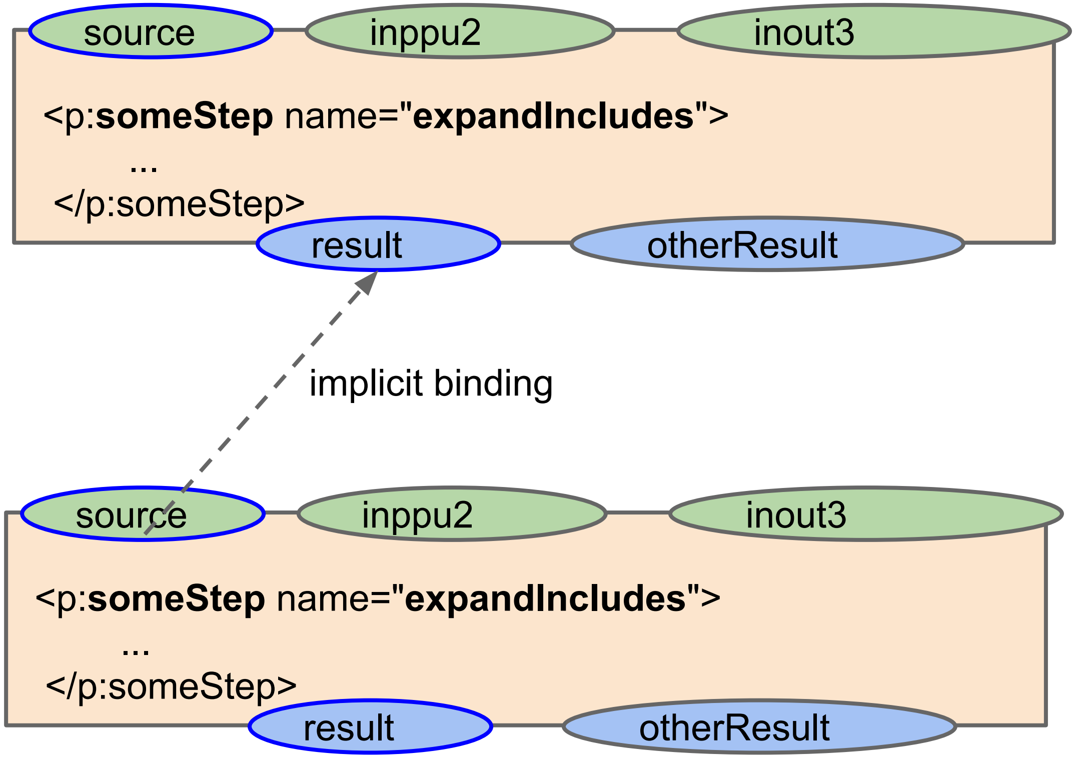
Though a port we can have:

- One XML document
- A sequence of XML documents

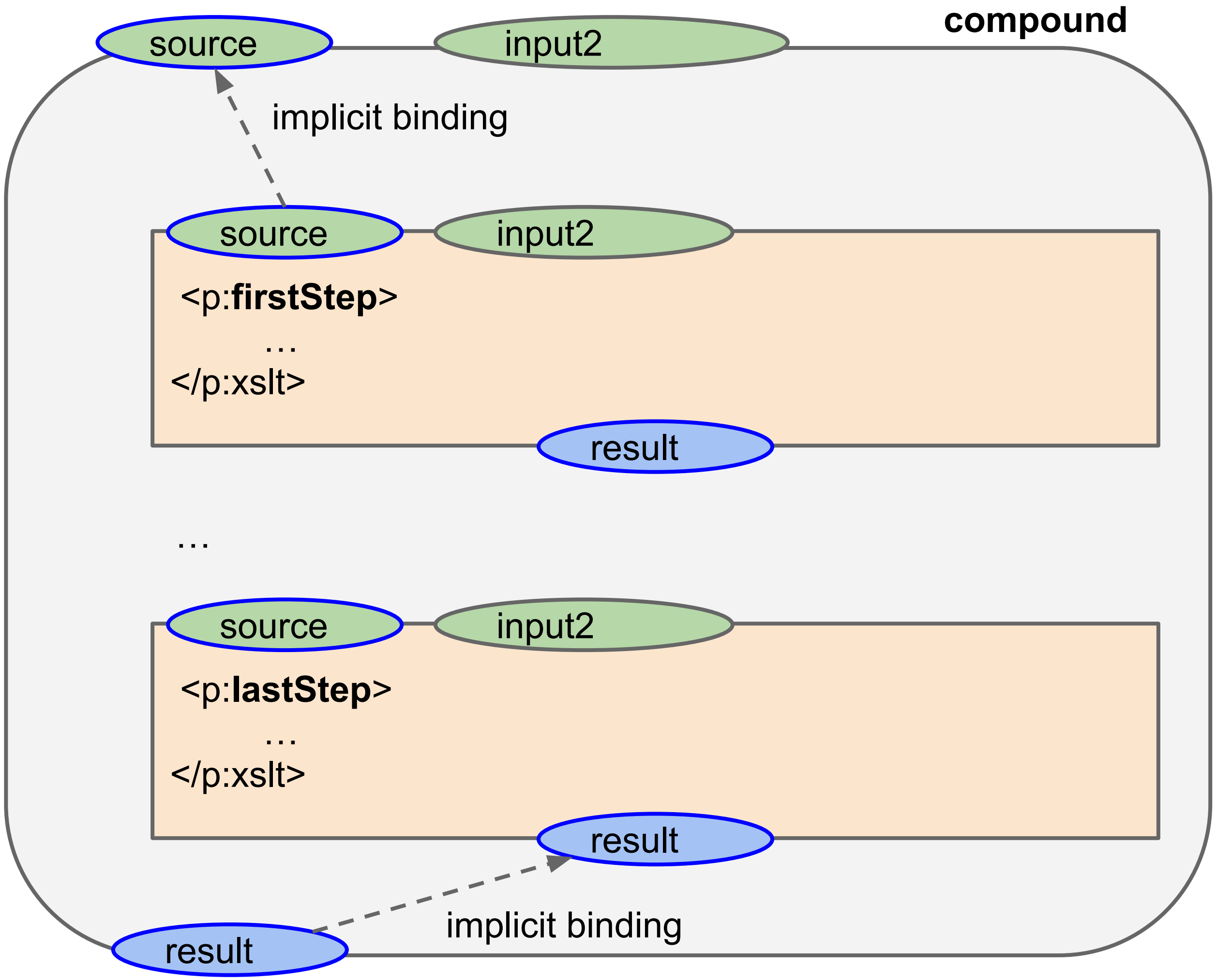
Connections

- Are always specified on the receiving port!
- There are implicit connections between primary input and output ports in a sequence of steps
- Connections can be made to:
  - A port that provides data
  - An external document
  - An inline document
  - An empty document

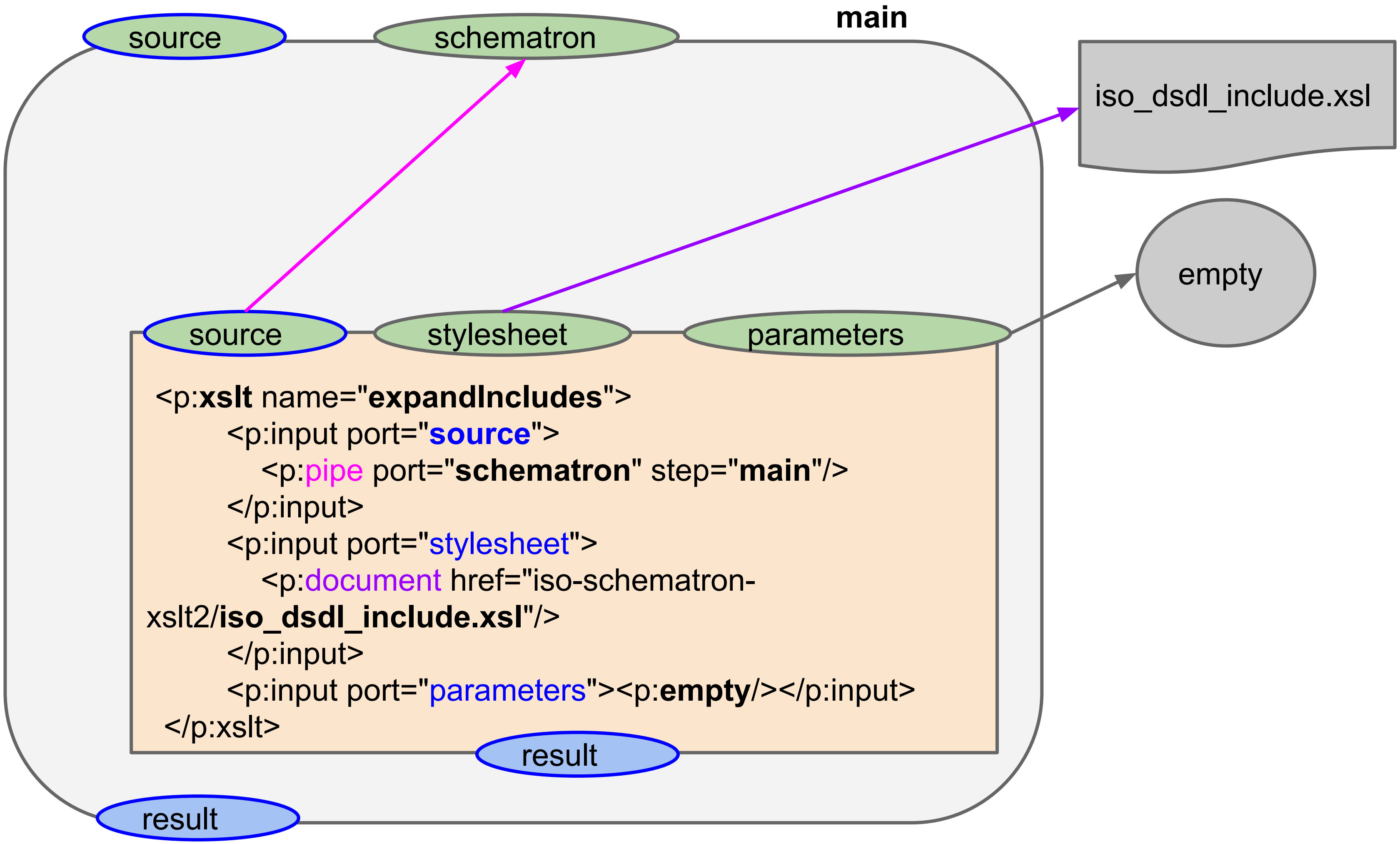
Connections between primary ports



Connections between primary ports



Explicit connections



```
<p:xslt name="expandIncludes">
        <p:input port="source">
            <p:pipe port="schematron" step="main"/>
        </p:input>
        <p:input port="stylesheet">
            <p:document href="iso-schematron-
xslt2/iso_dsdl_include.xsl"/>
        </p:input>
        <p:input port="parameters"><p:empty/></p:input>
 </p:xslt>
```

XProc implementations

- Calabash / Java
  - Norm Walsh, MarkLogic
- Calumet / Java
  - Vojtech Toman, EMC
- xprocxq / XQuery
  - James Fuler, MarkLogic
- QuiXProc / Java
  - Mohamed Zergaoui, Innovimax
- Componize / Java
  - Frank Shipley, Componize

XProc support in oXygen - 1/3

Plugable XProc engines

- default implementation for Calabash
- Calumet provides an oXygen connector

XProc script validation

- using Relax NG schema
- using the XProc processor, if that supports validation

XProc support in oXygen - 2/3

Run XProc scripts / transformations

- configure XProc transformation scenarios
- invoke individual and batch transformations

XProc editing

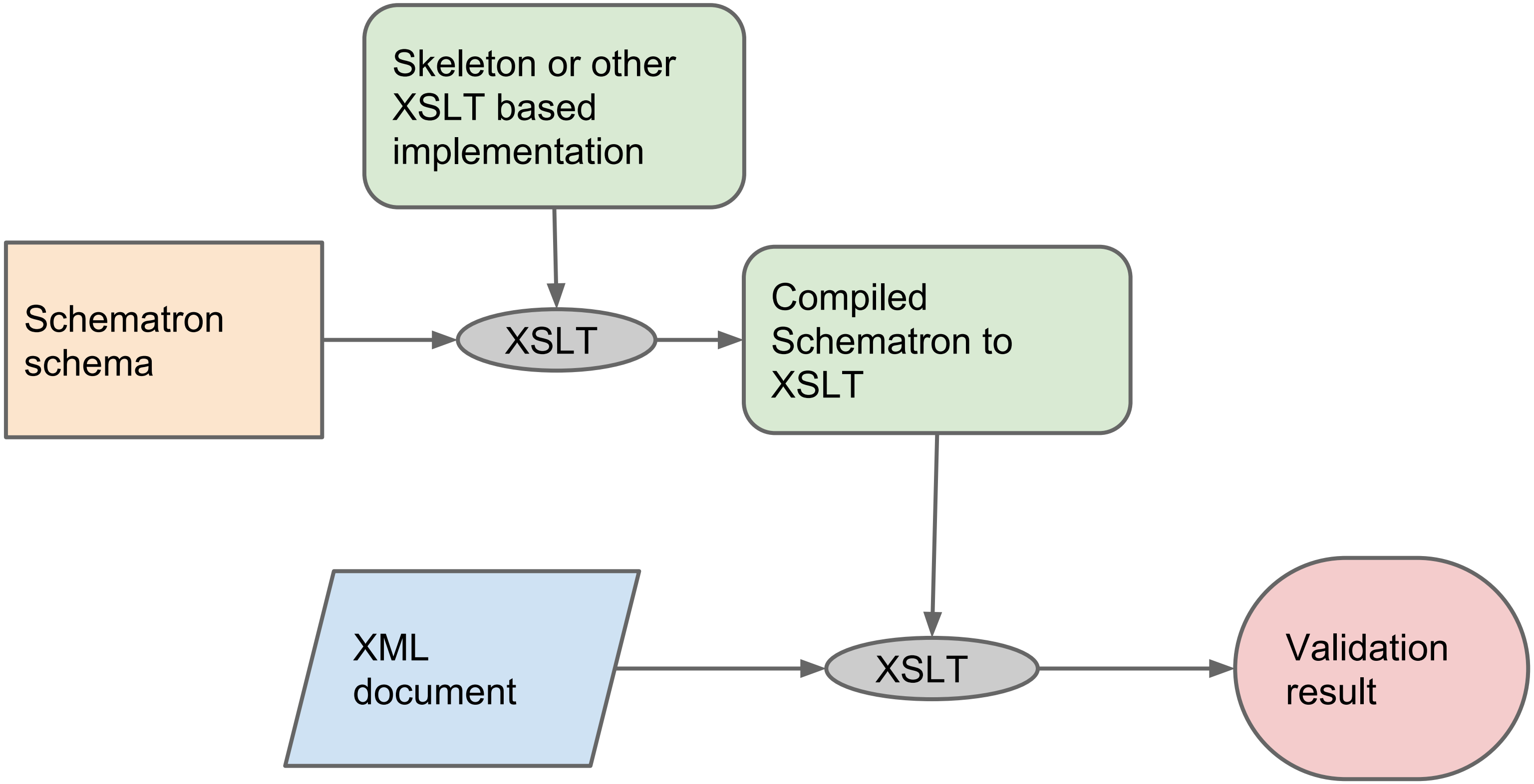
- driven by the XProc Relax NG schema
- support for hosted languages, like XSLT, XML Schema, etc.
- dynamic values, like defined port names

XProc support in oXygen - 3/3

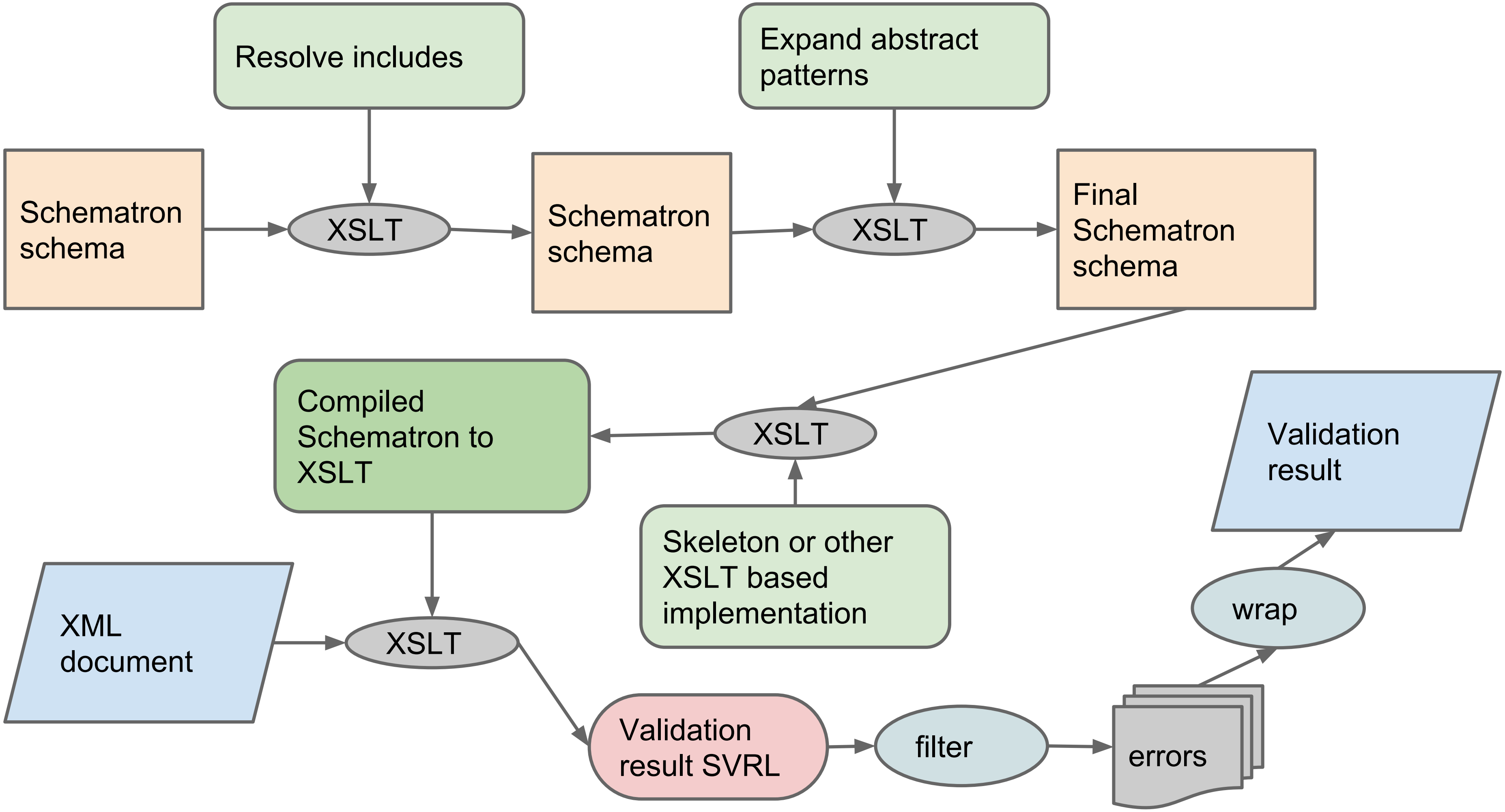
Author mode for XProc

- renders the XProc though CSS

Example: Schematron validation


Skeleton or other XSLT based implementation
Schematron schema
XSLT
Compiled Schematron to XSLT
XML document
XSLT
Validation result

The actual XProc processing flow



Usecases

XProc NVDL implementation as part of oNVDL open source project

Conversion of Relax NG schemas to one big DTD

XProc implementation of DITA transformations (Componize)

Learn more

Roger Costello’s xfront.com website:

http://xfront.com/xproc/index.html

What is new for XProc?

Public Working Draft of XProc V2.0 Requirements

http://www.w3.org/XML/XProc/staging/WD-xproc-v2-req-20131105/

Thank you!

Questions?

Now, or later at:

@georgebina

george@oxygenxml.com