Excellent XProc 3.0
What's up?

• XProc is an XML based programming language for complex data processing - pipelining
• V1 (2010) turned out to be
  • hard to use and understand
  • verbose
• People that use it, use it extensively
• V1 became outdated with respect to underlying standards
What happened?

• V2 Initiative (non-XML) – not enough support
• V3 Initiative (2016) - W3C Community Group
  • Stay close to existing syntax
  • Make language more usable, understandable and concise
  • Update underlying standards
  • Allow other document types to flow through
  • Clean up loose ends

• Editors
  • Norman Walsh, Achim Berndzen, Gerrit Imsieke, Erik Siegel

• Meetings, proposals, discussions, ...
Helicopter status

- Final call core spec
- Working on the steps
- End before end 2019

In the making:
- A specification (http://spec.xproc.org)
- Two processor implementations (XML Calabash, MorganaXProc)
- A programmer's reference book

And its name is... Kanava

Logo!

Thanks to Bethan Tovey
Crash Course XProc fundamentals
Pipelines, steps (1.0 & 3.0)

- Document(s) as input
- Process the data flowing through using steps
- Produce output(s)
Pipelines, steps (1.0 & 3.0)
Documents flowing through (1.0)

<doc>
...
</doc>

1.0: Only XML documents
Documents flowing through (3.0)

Native document types:
- XML
- HTML
- JSON
- Text
- Other

All documents carry an adaptable properties map

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>content-type</td>
<td>application/xml</td>
</tr>
<tr>
<td>base-uri</td>
<td>file:///C:/data/doc.xml</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Steps, ports, options (1.0 & 3.0)

- **document**
  - source
  - stylesheet
  - p:xslt
    - parameters
    - options
    - result
    - secondary
  - result
  - Transformed document

- **stylesheet**
  - options
  - parameters

- **p:xslt**
  - source
  - secondary
  - result
  - Additional documents

- **Base document**
  - source
  - insertion
  - options
  - match
  - position
  - Document to insert

- **p:insert**
  - source
  - result
  - Combined document
Primary ports (1.0 & 3.0)

The port names **source** and **result** for the primary ports are convention.
Primary ports, implicit connections (1.0 & 3.0)

Primary ports implicitly connect

Click!

```xml
<p:insert>
  ...
</p:insert>

<p:xslt>
  ...
</p:xslt>
```
Implicit connection of the first step to the primary input port.

Implicit connection of the last step to the primary output port.

Implicit connection of steps.

<code><p:declare-step xmlns:p="http://www.w3.org/ns/xproc">
    <p:input port="source" primary="true"/>
    <p:output port="result" primary="true"/>
    <p:xslt>
        <p:with-input port="stylesheet" href="myxslt1.xsl"/>
    </p:xslt>
    <p:xslt>
        <p:with-input port="stylesheet" href="myxslt2.xsl"/>
    </p:xslt>
</p:declare-step></code>
Explicitly connecting ports (3.0)

```
<p:xslt>
  <p:with-input port="stylesheet">
    ...
  </p:with-input>
</p:xslt>
```

Rules:
- Input ports *must* be connected (or have a default)
- Output ports can be left dangling (*also primary output ports*)
Explicitly connecting ports – with an external document (1.0 & 3.0)

1.0: You *had* to use `<p:document>` and `@href` had to be a literal

```xml
<p:xslt>
  <p:with-input port="stylesheet">
    <p:document href="/some/stylesheets.xsl"/>
  </p:with-input>
</p:xslt>
```
Explicitly connecting ports (3.0)

```
<p:variable name="path" select="'/some'"/>

<p:xslt>
    <p:with-input port="stylesheet">
        <p:document href="{$path}/stylesheet.xsl"/>
    </p:with-input>
</p:xslt>
```

NEW!

Attribute-Value Templates
Explicitly connecting ports (3.0)

```xml
<p:xslt>
  <p:with-input port="stylesheet"
               href="{$path}/stylesheet.xsl"/>
</p:xslt>
```

<p:document> can be shortened to a @href (just one example of lots of syntactic sugar shortcuts)
Explicitly connecting ports – to an inline document (1.0 & 3.0)

1.0: You *had* to use `<p:inline>`
Explicitly connecting ports – to an inline document (3.0)

Inline documents can contain Text- and Attribute-Value Templates

<p:xslt>
  <p:with-input port="stylesheet">
    <xsl:stylesheet ...
    ...
    </xsl:stylesheet>
  </p:with-input>
</p:xslt>
Explicitly connecting ports – to an output port of another step (1.0 & 3.0)

1.0: You had to use `<p:pipe>`
Explicitly connecting ports – to an output port of another step (3.0)

```xml
<p:xslt>
  <p:with-input port="stylesheet"
     pipe="port-name@other-step"/>
</p:xslt>
```

<p:pipe> can be shortened to @pipe

NEW!
Setting step parameters (1.0)

1.0: You had to use `<p:with-param>` or parameter ports
Setting step parameters

```xml
<p:xslt>
  ...
  <p:with-option name="parameters" select="
    map{ 'title': 'bla bla bla',
         'heading-level': 2
    }
  ">
</p:xslt>
```

Pass parameters in a map

NEW! XPath 3.1 support
Variables (1.0)

```xml
<p:variable name="path" select="//*/@path"/>

1.0: String only

1.0: Only at the top of a sub-pipeline

```
Variables (3.0)

```
<p:variable name="path" as="xs:string" select="/*/@path"/>
<p:variable name="debug" as="xs:boolean" select="true()"/>
<p:variable name="mymap" as="map(*)" select="map{ ... }"/>
```

XDM support and strong typing

- Static variables
  - @use-when expressions
  - Constants for setting up the pipeline
  - Constants in libraries

Variables anywhere
Structure (1.0 & 3.0)

- `<p:for-each>`
  - `<p:iteration-source>`
- `<p:choose>` / `<p:when>` / `<p:otherwise>`
- `<p:if>` ✨NEW!
- `<p:viewport>`
- `<p:group>`
- `<p:try>` / `<p:catch>` ✨NEW!
Crash Course XProc fundamentals - end
MorganaXProc meets Star Wars using JSON

https://swapi.co/api

XProc 3 pipeline

Runs on MorganaXProc 0.8.5-alpha
Wrap up

• XProc 3.0 is a more concise and easier-to-use version than 1.0
  • Updated underlying standards
  • Support for non-XML documents
  • Lots of syntactic sugar
  • Value templating {...}

• You can help us:
  • By joining the discussions
  • By reviewing the specification

• Processors and learning materials are on their way
XProc 3.0

- XProc on github: [https://github.com/xproc/](https://github.com/xproc/)
- XProc on W3C: [https://www.w3.org/community/xproc-next/](https://www.w3.org/community/xproc-next/)
- Use the issues list on github
- Contact: xproc-dev@w3.org