NoPHP: A Conference Website in XForms

Steven Pemberton, CWI Amsterdam

Version: 2019-07-24

Contents

Introduction

Consider this conference program overview:

Conference Programme

Preconference Tutorials

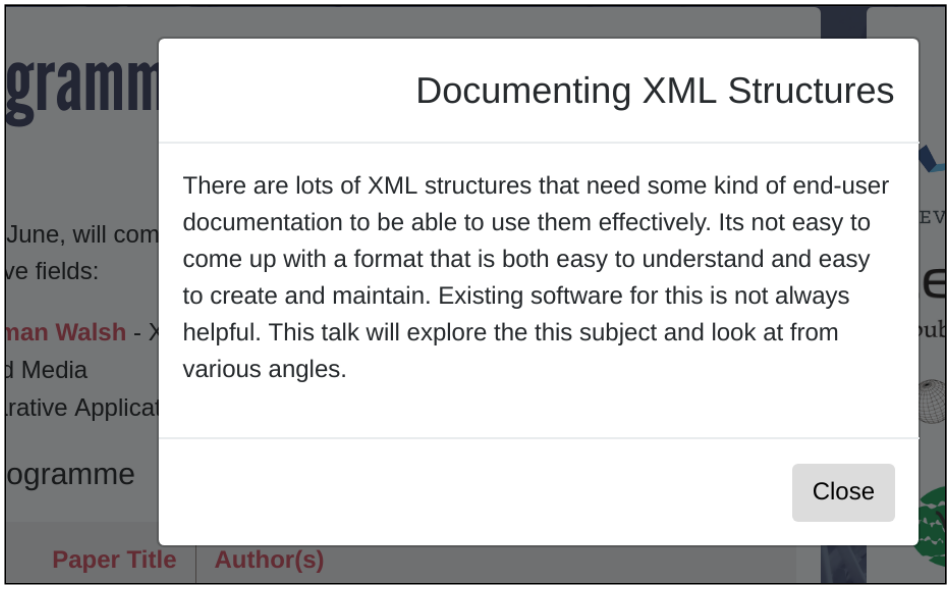
The preconference day, Friday 7 June, will comprise of three tutorial sessions by some of the world's foremost experts in their respective fields:

- Achim Berndzen and Norman Walsh - XProc 3.0
- Tony Graham - CSS Paged Media
- Steven Pemberton - Declarative Applications with XForms

Conference Preliminary Programme

Paper Title	Author(s)
<Angle-brackets/> on the Branch Line - (model) railway engineering with XML technologies Abstract	John Lumley (jwLResearch)*
Documenting XML Structures Abstract	Erik Siegel (Xatapult)*
An improved diff3 format for changes and conflicts in tree structures	Robin La Fontaine (DeltaXML Ltd)*; Nigel A Whitaker (DeltaXML Ltd)

If you look at the code, you can see that the page contains all the content, including abstracts for the talks, with the use of several scripts to handle pop ups and so on. The abstracts are initially hidden, but clicking on an "Abstract" button pops up the abstract for a talk:



Let's do this with XForms.

The Data

Firstly, we don't want the data about the talks in the page. The talks will be stored in two files, the tutorials:

```
<tutorials when="Friday 7 June">
   <talk>
      <title>XProc 3.0</title>
      <author>Achim Berndzen</author>
      <author>Norman Walsh</author>
   </talk>
   <talk>
      <title>CSS Paged Media</title>
      <author>Tony Graham</author>
   </talk>
   <talk>
      <title>Declarative Applications with XForms</title>
      <author>Steven Pemberton</author>
   </talk>
</tutorials>
```

and similar for the talks, but with abstracts:

```
<talks>
   <talk>
      <title>&lt;Angle-brackets/&gt; on the Branch Line
             - (model) railway engineering with XML technologies</title>
      <abstract>
         <p>[Elided]</p>
      </abstract>
      <author>John Lumley (jwLResearch)*</author>
   </talk>
   <talk>
      <title>Documenting XML Structures</title>
      <abstract>
         <p>[Elided]</p>
      </abstract>
      <author>Erik Siegel (Xatapult)*</author>
   </talk>
   <talk>
      etc.
```

We create two instances to store that data:

```
<instance id="tutorials" src="tutorials.xml"/>
<instance id="talks"     src="talks.xml"/>
```

Since this is to be generic, so that it can also be used in future years, we've added the date to the top element of the tutorials data. Now we can start displaying. Starting with the tutorials, we'll begin with the prelude. The `group` element sets the context for the enclosed bit so that all enclosed references will by default be to data in the tutorials instance. Since we don't know how many tutorials there will be each year we let XForms do the work, by outputting a count of the talks.

```
<group ref="instance('tutorials')">
   <label class="h2">Preconference Tutorials</label>
   The preconference day, <output ref="@when"/>,
   will comprise of <output value="count(talk)"/>
   tutorial sessions by some of the world's foremost experts in their respective fields:
```

Now we can output the list of tutorials. Note that it deals with more than one author:

```
<repeat ref="talk">
   <output class="title" ref="title"/>
   <repeat ref="author">
      <output class="author" ref="."/>
   </repeat>
</repeat>
```

which, with suitable CSS, looks like this:

Conference Programme

Preconference Tutorials

The preconference day, Friday 7 June, will comprise of 3 tutorial sessions by some of the world's foremost experts in their respective fields:

XProc 3.0
Achim Berndzen
Norman Walsh

CSS Paged Media
Tony Graham

Declarative Applications with XForms
Steven Pemberton

Source

The Talks

These are treated in the same way, except we now want to add a button to display the abstract. It starts the same:

```
<group ref="instance('talks')">
  <repeat ref="talk">
     <output class="title" ref="title"/>
     <repeat ref="author">
        <output class="author" ref="."/>
     </repeat>
```

Then comes the button to show the abstract:

```
   <trigger ref="abstract">
      <label>Abstract</label>
      <action ev:event="DOMActivate">
         <toggle case="show"/>
      </action>
   </trigger>
```

Attaching the trigger like this to the abstract (`ref="abstract"`) means that if there is no abstract, there'll be no button (we've removed the abstract from one of the talks below to show this working).

When the trigger is clicked on, or otherwise activated, a `DOMActivate` event is sent. This causes the `toggle` element to be processed.

The `toggle` element causes a case of a `switch` element to be activated. Switch has a number of cases, only one of which is visible at a time. Initially by default the first is visible, and a different one can be activated with a toggle. In this case, we have a switch with two cases, one which shows nothing, and the other that shows the abstract:

```
   <switch>
      <case id="hide"/>
      <case id="show">
         ... this is where we display the abstract ...
      </case>
   </switch>
```

To display the abstract, we repeat over the paragraphs of the data. This is preceded by a trigger to hide the abstract again by activating the other case:

```
<group ref="abstract" class="abstract">
   <trigger>
      <label>x</label>
      <action ev:event="DOMActivate">
         <toggle case="hide"/>
      </action>
   </trigger>
   <repeat ref="p">
      <output class="p" ref="."/>
   </repeat>
</group>
```

This looks like this (try it):

Conference Preliminary Programme

<Angle-brackets/> on the Branch Line - (model) railway engineering with XML technologies
*John Lumley (jwLResearch)**
Abstract

Documenting XML Structures
*Erik Siegel (Xatapult)**
Abstract

An improved diff3 format for changes and conflicts in tree structures
*Robin La Fontaine (DeltaXML Ltd)**
Nigel A Whitaker (DeltaXML Ltd)
Abstract

Scrap the App, Keep the Data
*Barnabas Davoti (Ovitas AS)**

Software we have lost — the mortar that held the bricks together
*Peter Flynn (Silmaril Consultants)**
Abstract

Source

This doesn't pop up the abstract, but just makes it visible. Should popping up be essential to the user experience, then we just change the CSS for that. You really don't need Javascript for these purposes:

Conference Preliminary Programme

<Angle-brackets/> on the Branch Line - (model) railway engineering with XML technologies
*John Lumley (jwLResearch)**
Abstract

Documenting XML Structures
*Erik Siegel (Xatapult)**
Abstract

An improved diff3 format for changes and conflicts in tree structures
*Robin La Fontaine (DeltaXML Ltd)**
Nigel A Whitaker (DeltaXML Ltd)
Abstract

Scrap the App, Keep the Data
*Barnabas Davoti (Ovitas AS)**

Software we have lost — the mortar that held the bricks together
*Peter Flynn (Silmaril Consultants)**
Abstract

Source

If your complaint is that it is formatted differently to the original, well, we just change the CSS for that as well (this isn't a CSS tutorial, but if you need a hint, we've used `display: inline-block`).

Conference Preliminary Programme

<Angle-brackets/> on the Branch Line - (model) railway engineering with XML technologies Abstract	*John Lumley (jwLResearch)**
Documenting XML Structures Abstract	*Erik Siegel (Xatapult)**
An improved diff3 format for changes and conflicts in tree structures Abstract	*Robin La Fontaine (DeltaXML Ltd)** *Nigel A Whitaker (DeltaXML Ltd)*
Scrap the App, Keep the Data	*Barnabas Davoti (Ovitas AS)**
Software we have lost — the mortar that held the bricks together Abstract	*Peter Flynn (Silmaril Consultants)**

Source

Finally, if a popping up animation is your thing, you can add that with a bit of CSS as well. (Look up CSS animations to discover how this was done.)

Conference Preliminary Programme

<Angle-brackets/> on the Branch Line - (model) railway engineering with XML technologies
*John Lumley (jwLResearch)**
Abstract

Documenting XML Structures
*Erik Siegel (Xatapult)**
Abstract

An improved diff3 format for changes and conflicts in tree structures
*Robin La Fontaine (DeltaXML Ltd)**
Nigel A Whitaker (DeltaXML Ltd)
Abstract

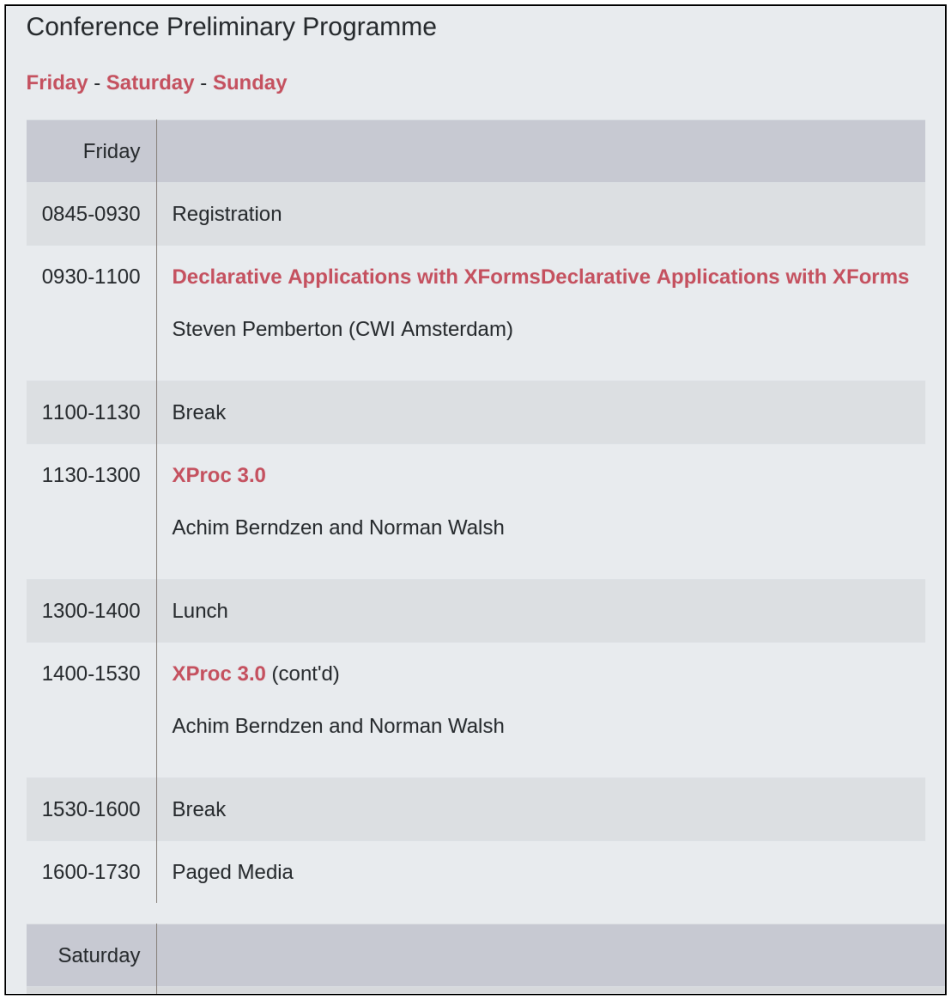
Scrap the App, Keep the Data
*Barnabas Davoti (Ovitas AS)**

Software we have lost — the mortar that held the bricks together
*Peter Flynn (Silmaril Consultants)**
Abstract

Source

The Conference Program

Later, the conference changed the page to include the timings. Let's adapt.



The Program

Our options here are to reorder and add times to the talks document we created, or to create a separate program document.

Let's do that.

The program consists of a number of days, and each day consists of a number of slots. A slot has a start and end time, and a title:

```
<program>
   <day which="Friday">
      <slot start="08:45" end="09:30">Registration</slot>
      <slot start="09:30" end="11:00">Declarative Applications with XForms</slot>
      <slot start="11:00" end="11:30">Break</slot>
      <slot start="11:30" end="13:00">XProc 3.0</slot>
      <slot start="13:00" end="14:00">Lunch</slot>
      <slot start="14:00" end="15:30">XProc 3.0 (cont'd)</slot>
      <slot start="15:30" end="16:00">Break</slot>
      <slot start="16:00" end="17:30">CSS Paged Media</slot>
   </day>
   <day which="Saturday">
      <slot start="08:45" end="09:30">Registration</slot>
      <slot start="09:30" end="10:15">Everyone Knows What a Dragon Looks Like</slot>
```

etc.

To display this we repeat over the days, and for each day, repeat over the slots:

```
<repeat ref="instance('program')/day">
   <output class="h2" value="@which"/>
   <repeat ref="slot">
      <output class="times" value="concat(@start, '–', @end)"/>
      <output class="entry" ref="."/>
   </repeat>
</repeat>
```

Here's what this looks like:

Friday

08:45–09:30	Registration
09:30–11:00	Declarative Applications with XForms
11:00–11:30	Break
11:30–13:00	XProc 3.0
13:00–14:00	Lunch
14:00–15:30	XProc 3.0 (cont'd)
15:30–16:00	Break
16:00–17:30	CSS Paged Media

Saturday

08:45–09:30	Registration
09:30–10:15	Everyone Knows What a Dragon Looks Like
10:15–11:00	Beyond the brick, for the past in the future, you find the archive!
11:00–11:30	Break
11:30–12:00	Software we have lost — the mortar that held the bricks together

Source

This of course is still missing the authors and the abstract. We get those from the talks instance by finding the talk whose title is the same as the entry in the slot, and then output the authors and abstract (in the same way as earlier):

```
<group class="entry">
   <output class="title" ref="."/>
   <group ref="instance('talks')/talk[title=context()]">
      <repeat ref="author">
         <output class="author" ref="."/>
      </repeat>
      <trigger ref="abstract">
      ... etc ...
```

If no talk matches the condition, no talk will be selected, and so neither author nor abstract will be output. (By the way, we've merged the tutorials instance, and the talks instance for simplicity in this version, and used a different transition in the CSS for the abstracts).

Here's what it looks like:

Friday

08:45-09:30	**Registration**
09:30-11:00	**Declarative Applications with XForms** *Steven Pemberton*
11:00-11:30	**Break**
11:30-13:00	**XProc 3.0** *Achim Berndzen* *Norman Walsh*
13:00-14:00	**Lunch**
14:00-15:30	**XProc 3.0 (cont'd)**
15:30-16:00	**Break**
16:00-17:30	**CSS Paged Media** *Tony Graham*

Saturday

08:45-09:30	**Registration**
09:30-10:15	**Everyone Knows What a Dragon Looks Like** *Tommie Usdin (Mulberry Technologies, Inc.)**

Source

Dates and Times

Since the data contains times, an extra feature that we can add is to show which items are in the past, which in the future, and which is happening now.

For instance on Saturday at 11:30, the display could look like this:

To achieve this, we only have to compare the time for the slot with the time now: if the slot end time is less than the current time, the slot is in the past, if the start time is less than the current time, but the end time isn't, then the slot is now active, and otherwise the slot is in the future.

In essence, we set the class of the slot by calculating which of three it is, past, future, or now:

```
<repeat ref="slot">
   <group class="{if(@end   &lt; instance('admin')/now, 'past',
                 if(@start &lt; instance('admin')/now, 'now',
                                                       'future'))}">
```

That is the essence. Now the detail.

To do the comparisons properly, we need both the time *and* the date. And because the computer displaying the page may be set to a different timezone, we need the timezone information as well.

One approach would be to set all the `start` and `end` attributes in the program data file to include full date and time information:

```
<slot start="2019-06-07T08:45:00+01:00" end="2019-06-07T09:30:00+01:00">Registration</slot>
```

and then display the times accordingly:

```
<output class="times" value="concat(substring(@start, 12, 5), '–', substring(@end, 12, 5))"/>
```

Then the comparisons would be simple. We have the times in the timezone of the conference, but we need to convert them to the timezone of the computer displaying the page:

```
class="{if(adjust-dateTime-to-timezone(@end)   &lt; instance('admin')/now, 'past',
        if(adjust-dateTime-to-timezone(@start) &lt; instance('admin')/now, 'now',
                                                                           'future'))}">
```

All that we then need to do is update the `now` value every minute. At startup we dispatch an event we shall call `tick`:

```
<action ev:event="xforms-ready">
   <dispatch name="tick" targetid="m"/>
</action>
```

and we catch it, update the `now` value, and dispatch another `tick`, with a delay of a minute (in milliseconds):

```
<action ev:event="tick">
   <setvalue ref="instance('admin')/now" value="local-dateTime()"/>
   <dispatch name="tick" delay="60000" targetid="m"/>
</action>
```

That solution, though simple, does require a lot of fiddly work for the author of the data. The other option is to include the parts of the date and time in the data at suitable points:

```
<program>
  <day which="Friday" date="2019-06-07" timezone="+01:00">
    <slot start="08:45" end="09:30">Registration</slot>
```

(The timezone needs to be included per day, for the case that the conference straddles a change to or from daylight savings time.)

In this case, the display of the times remain the same as the old method, but the comparisons have to be done differently. For each slot we have to construct the time to compare:

```
concat(../@date, 'T', @start, ':00', ../@timezone)
```

and again convert to the right timezone::

```
adjust-dateTime-to-timezone(concat(../@date, 'T', @start, ':00', ../@timezone))
```

Here it is working. I have faked the dates, so that the conference appears to start today, and the times are in your timezone, so that you can see it working.

Friday

08:45–09:30	**Registration**
09:30–11:00	**Declarative Applications with XForms** *Steven Pemberton*
11:00–11:30	**Break**
11:30–13:00	**XProc 3.0** *Achim Berndzen* *Norman Walsh*
13:00–14:00	**Lunch**
14:00–15:30	**XProc 3.0 (cont'd)**
15:30–16:00	**Break**
16:00–17:30	**CSS Paged Media** *Tony Graham*

Saturday

08:45–09:30	**Registration**
09:30–10:15	**Everyone Knows What a Dragon Looks Like** *Tommie Usdin (Mulberry Technologies, Inc.)**

Source

Error checking

Suppose that there is a talk in the talks document that was forgotten in the program, or added twice to the program by mistake. To warn for that we repeat over the talk titles:

```
<repeat ref="instance('talks')/talk/title">
```

and output those that don't occur exactly once in the program:

```
   <output class="warning" ref=".[...condition here...]"/>
```

How do we find out how many times a title occurs in a program? We locate all slots in the program with that title:

```
instance('program')/day/slot[. = context()]
```

and then count them:

```
count(instance('program')/day/slot[. = context()])
```

That count should be 1.

In full:

```
<repeat ref="instance('talks')/talk/title">
   <output class="missing" ref=".[count(instance('program')/day/slot[. = context()]) != 1]"/>
</repeat>
```

If there are no missing or duplicated talks, nothing will show up.

One other thing we can check for is that there are no unaccounted gaps between slots, by reporting all the slots whose start time doesn't match the preceding slot's end time:

```
<repeat ref="instance('program')/day/slot[
             position() != 1 and @start != preceding-sibling::slot[1]/@end]">
   <output class="gap"
           value="concat('On ', ../@which, ' there is a gap between ',
                         preceding-sibling::slot[1]/@end, ' and ', @start)"/>
</repeat>
```

Error check

In talks file but not in program: Using XForms to locate lost conference papers

On Saturday there is a gap between 2019-06-08T17:30:00+01:00 and 2019-06-08T18:00:00+01:00

Source

But if we can check that they abut, why not just assume they do? Each slot then only needs a start time, and unless there is an end time given, we can assume it ends at the start time of the next slot:

```
<program>
  <day which="Friday" date="2019-06-19" timezone="+01:00">
    <slot start="08:45">Registration</slot>
    <slot start="09:30">Declarative Applications with XForms</slot>
    <slot start="11:00">Break</slot>
    <slot start="11:30">XProc 3.0</slot>
    <slot start="13:00">Lunch</slot>
    <slot start="14:00">XProc 3.0 (cont'd)</slot>
    <slot start="15:30">Break</slot>
    <slot start="16:00" end="17:30">CSS Paged Media</slot>
  </day>
  <day which="Saturday" date="2019-06-20" timezone="+01:00">
      ...
```

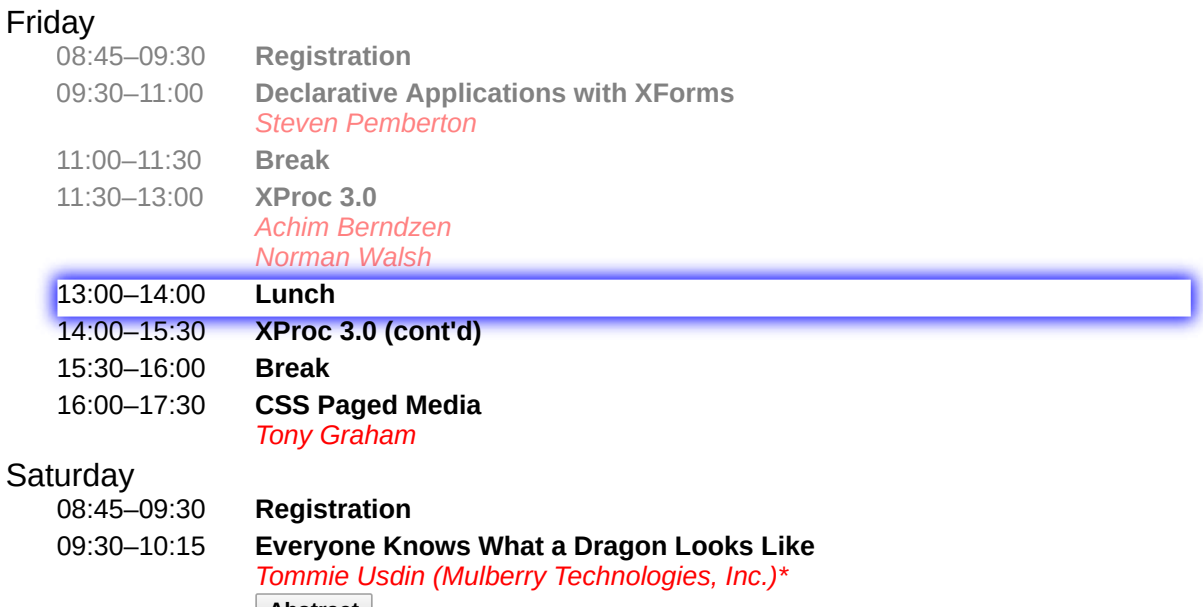
In this version, instead of displaying the times with

```
<output class="times" value="concat(@start, '–', @end)"/>
```

we display the end time if it is given, and otherwise the start time of the next slot:

```
<output class="times"
        value="concat(@start, '–', if(@end, @end, following-sibling::slot/@start))"/>
```

So here is the final version, again with faked dates and timezone, so that you can see it working, and a different CSS style for the current talk:



Source

It comprises 111 lines: 10 for the template, 26 lines of CSS, and 75 lines of XForms, of which 10 are there to support the CSS transition effects, and 2 for faking the dates.