EXPath

A packaging system for XML libraries and a portable web application framework

XML Prague, March 13th, 2010

Florent Georges

H2O Consulting

H2O Consulting

EXPath

- **Introduction** ←
- Simple examples
- The packaging system
- Web applications
- A comprehensive example
- Conclusion

H2O Consulting

Introduction - History

- EXSLT for XSLT 1.0
- XSLT 2.0 and needs for new extensions
- EXSLT 2.0, EXQuery & EXProc
- XML Prague 2009 – EXPath
- First modules – HTTP Client & ZIP Facility
- Balisage 2009 – the Packaging System
- XML Prague 2010 – the Webapp module

H2O Consulting

Introduction - Scope

- Quite vague *(that is a feature, not a bug)*
- Extensions for XPath of course
- But also: XSLT, XQuery, XProc (and XForms, XML Schema, DSDL, etc.; that is, XML technologies)
- Mainly extension functions
- But also: servlet container, packaging system...

Introduction - Goals

- *Collaboratively defining open standards for portable XPath extensions*
- The main means is extension functions
- The main goal is defining portable specifications...
- ...and convincing vendors to endorse them
- But also providing support to open-source implementations

H2O Consulting

Introduction - Processes

- More or less formal, more or less informal *(that is a feature, not a bug)*
- The definitive goal is writing specifications
- The main tool is the mailing list
- Each module has one main maintainer, responsible of editing & achieving consensus
- Other tools include Subversion repositories, public Wiki, etc.
- More infos about processes on the wiki

EXPath

- Introduction
- **Simple examples** ←
- The packaging system
- Web applications
- A comprehensive example
- Conclusion

H2O Consulting

Examples – HTTP Client

- A single function to send HTTP requests and handle corresponding responses
- Get the description of the request as an XML element (with verb, headers, content...)
- Return a description of the response as an XML element (with code, headers, content...)
- Actually the content is handled differently (as separate items, either text, XML, HTML or binary) to avoid the SOAP envelope syndrome

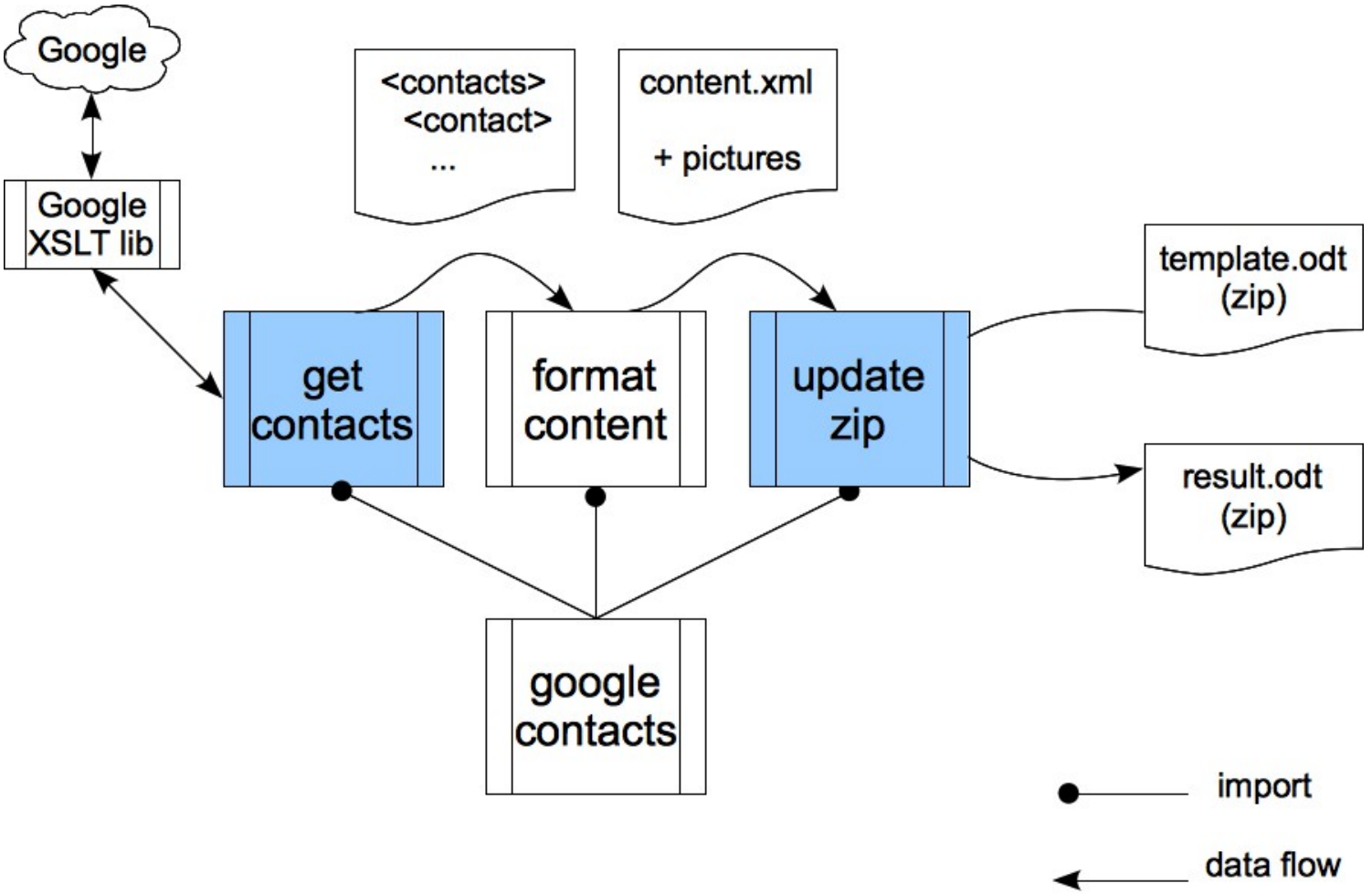
Examples – HTTP Client

```
http:send-request($request as element(http:request)) as item()+


<http:request href="http://www.example.com/..." method="post">
   <http:header name="X-Header" value="some value"/>
   <http:body content-type="application/xml">
      <hello>World!</hello>
   </http:body>
</http:request>


<http:response status="200" message="Ok">
   <http:header name="..." value="..."/>
   ...
   <http:body content-type="application/xml"/>
</http:response>
```

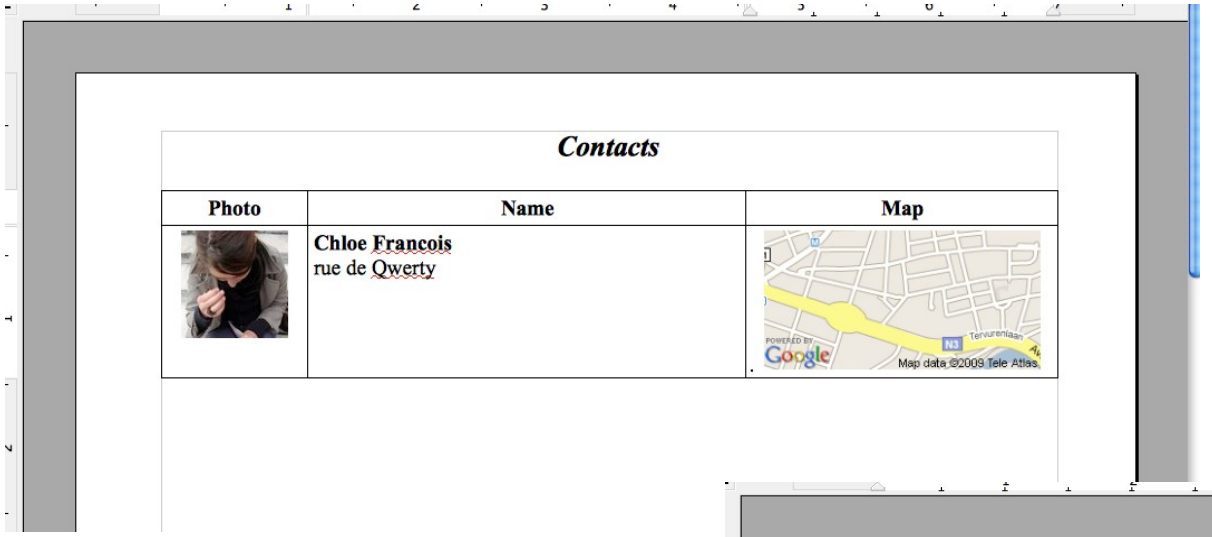
Examples – ZIP Facility

- List entries:
  - zip:entries($href) as element(zip:file)
- Extract entries:
  - zip:xml-entry($href, $path) as document-node()
  - zip:html-entry($href, $path) as document-node()
  - zip:text-entry($href, $path) as xs:string
  - zip:binary-entry($href, $path) as xs:base64Binary
- Create new ZIP files:
  - zip:zip-file($zip) as empty()
  - zip:update-entries($zip, $output) as empty()

Examples – Google Contacts

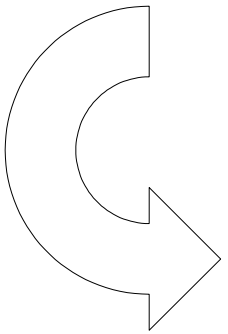


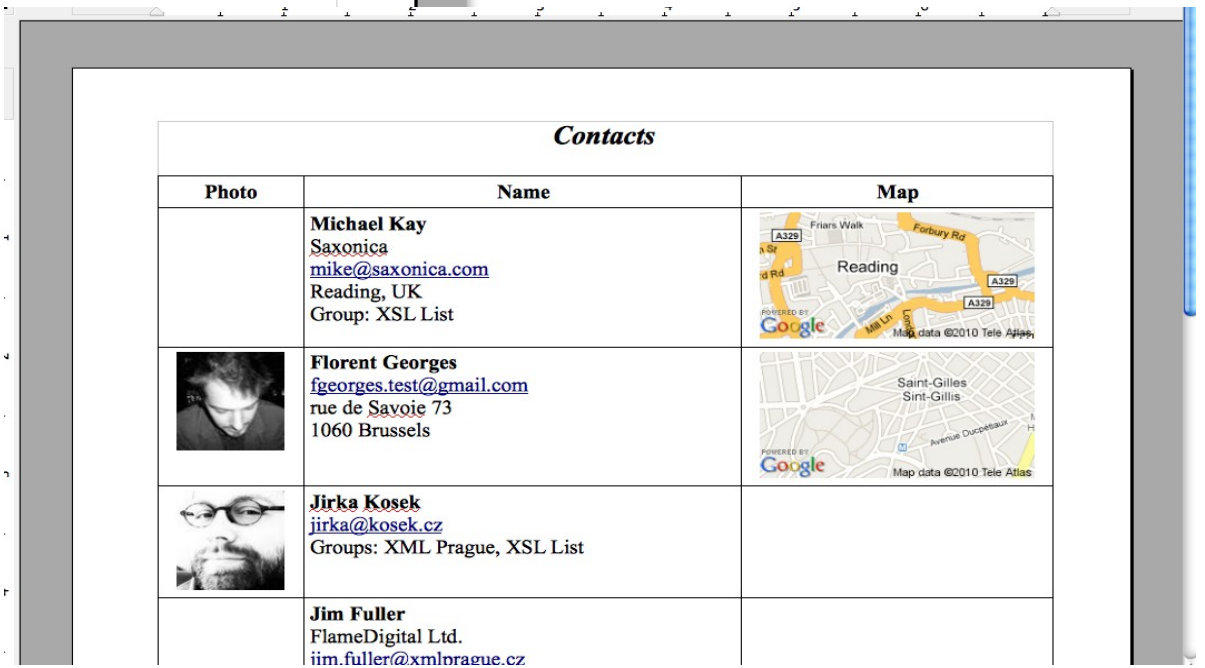
Examples – Google Contacts



Contacts

Photo	Name	Map
	Chloe Francois rue de Qwerty	





Contacts

Photo	Name	Map
	Michael Kay Saxonica mike@saxonica.com Reading, UK Group: XSL List	
	Florent Georges fgeorges.test@gmail.com rue de Savoie 73 1060 Brussels	
	Jirka Kosek jirka@kosek.cz Groups: XML Prague, XSL List	
	Jim Fuller FlameDigital Ltd. jim.fuller@xmlprague.cz	

H2O Consulting

EXPath

- Introduction
- Simple examples
- **The packaging system** ←
- Web applications
- A comprehensive example
- Conclusion

H2O Consulting

Packaging – The problem

- How to install a library of UBL format helpers?
- Depends on the processor
- Easy: just copy the files *somewhere*
- To use them, adapt the XSLT import URI or the XQuery import *at hint* in your using stylesheets or queries
  - `<xsl:import href="file:///c:/xquery/ubl-helpers/mod-1.xsl"/>`
  - `import module namespace ubl = "http://fgeorges.org/ubl/mod-1" at "dadb://modules/ubl-helpers/mod-1.xql";`

Packaging – Houston, we've...

- But UBL Helpers actually use FunctX
- So you have to change the import statements throughout this third-party library
- Wait a minute, no problem, really, UBL Helpers have to include FunctX
- But your application uses another library, which in turn also uses FunctX
- ...

Packaging – Intermezzo

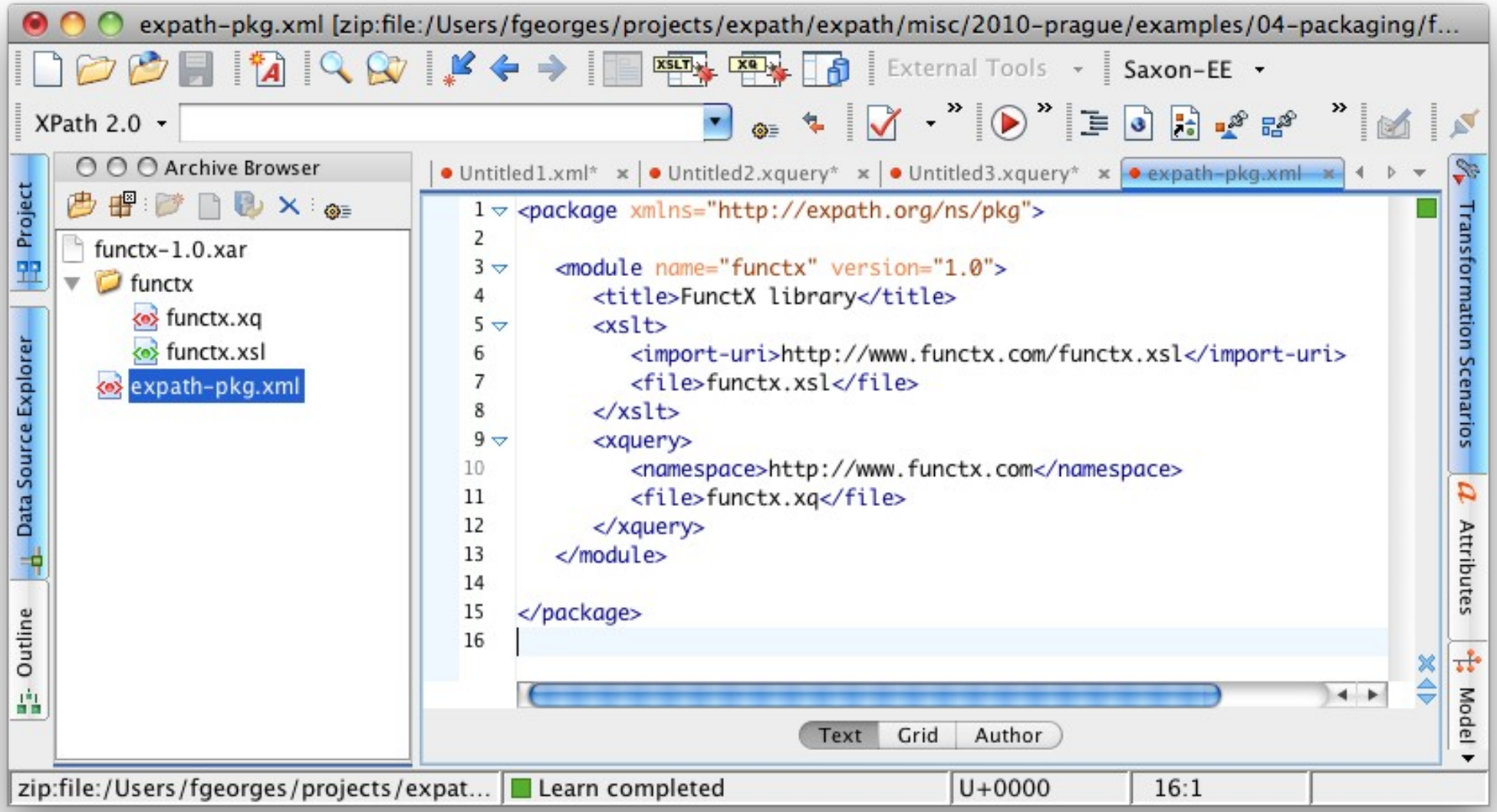
- OASIS XML Catalogs provide a bit of fresh air
- They help avoiding the requirement of changing import statements...
- ...as long as everyone agree on URIs to use in import statements
- They must then be supported by all products
- One has to install configure catalogs for its system by hand, again and again

H2O Consulting

Packaging – Standard format

- The solution lays in a standard format to describe X* components
  - Must describe what is needed but is not in the X* specifications *(well-defined import URIs)*
  - Must be understood by most processors *(suitable for standard libraries)*
  - Must package the components and additional informations in a single file *(suitable as a delivery format)*
  - Must be eXtensible *(that's **X**ML, isn't it?)*
- Installation process can then be automated

Packaging – Overview



H2O Consulting

Packaging – Tools

- Standard on-disk repository layout
- Command-line repository manager
- Implementation for Saxon (XQuery, XSLT)
- Implementation for Calabash (RELAX NG RNG + RNC, Schematron, XProc, XQuery, XML Schema and XSLT)
- Implementations provide a Java API as well as command-line scripts

Packaging – Further

- Repository manager can install directly from the web
  - xrepo install http://cxan.org/xxx.xar
- That leads to the idea of CXAN (like CPAN for Perl or CTAN for TeX/LaTeX): a global, online repository of existing package
  - cxan install functx
- A key point is user and vendor adoption, and support in existing tools like IDEs

H2O Consulting

Packaging – Projects

- The package format is low-level and strict
  - e.g. require the target namespace for an XQuery module in the package descriptor
- Using simple conventions, it is possible to create consistent project structure
- Other info can be added in the component source files
- Structure and annotations captures some packaging information, used by a packager

Packaging – Projects

- Low-level specifications are technical and strict
- They are as less as possible
- Conventions are flexible
- They can be used to ease developer's day-to-day life, by automating repetitive tasks
- They can be used to define different kinds of projects
  - Standard X* library, extension, webapp, tests...

H2O Consulting

EXPath

- Introduction
- Simple examples
- The packaging system
- **Web applications** ←
- A comprehensive example
- Conclusion

H2O Consulting

Webapps – Intro

- Using X* technologies end-to-end for web applications
- Most existing XML databases provide proprietary framework for that (eXist, MarkLogic, Sausalito, etc.)
- Then again, we are stuck with processor-locked applications
- A standard would allow to write portable web applications, libraries and frameworks

Webapps – Principles

- Not define all aspects of web applications
- A component can be an XQuery function or module, an XSLT function, template or stylesheet, or an XProc pipeline
- The only technical missing piece is a way to map HTTP requests to components
- That means dispatching and providing infos

Webapps – Mapping

- URLs are matched by regular expressions, and associated to a component
- A component is identified by its name (if applicable) and the import URI of it module
- This is configured in a simple descriptor
- ...and packaged using the Packaging System
- The Packaging System already provides setting of the component's public URIs
- The web descriptor simply builds on top of that

H2O Consulting

Webapps – Requests

```
<web:request servlet="name" path="/path" method="get">
  <web:uri>http://example.org/my-app/path</web:uri>
  <web:authority>http://example.org</web:authority>
  <web:context-root>/my-app</web:context-root>
  <web:path>
    <web:part>path</web:part>
  </web:path>
  <web:header name="connection" value="keep-alive"/>
  ...
</web:request>
```

```
<web:response status="200" message="Ok">
  <web:header name="..." value="..."/>
  ...
  <web:body content-type="text/html" method="xhtml"/>
</http:response>
```

H2O Consulting

Webapps – Servlex

- Servlex is an early implementation
- Server technology is Java Servlet
- XSLT and XQuery are provided by Saxon
- XProc is (going to be) provided by Calabash
- Webapp manager to (un)deploy applications

EXPath

- Introduction
- Simple examples
- The packaging system
- Web applications
- **A comprehensive example** ←
- Conclusion

H2O Consulting

Freedom

- A web application to download a backup of your data in some Google services
- Use EXPath ZIP Facility
- Use EXPath HTTP Client
- Use Google API XSLT libraries
- Use OAuth-like authentication

H2O Consulting

That's all Folks!

- Join the mailing list and browse the website:

http://expath.org/

H2O Consulting