



## **Easy Page Composition with XML Style Tags: *Work in Progress***

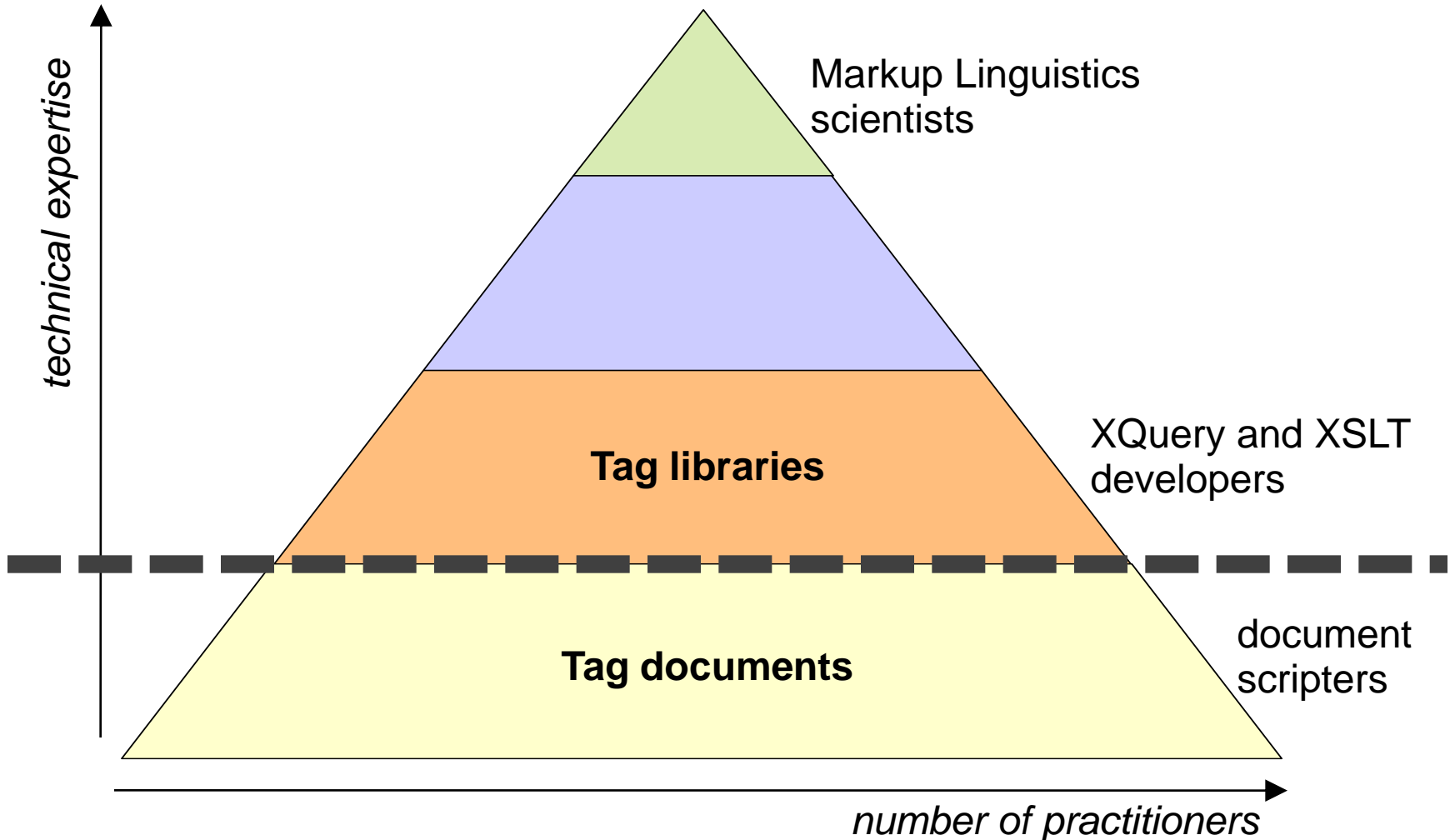
Erik Hennum, Senior Application Services Engineer  
Vyacheslav Zholudev, Jacobs University Bremen  
2011-4-25

## Extensible page parts through function references

```
declare function asc:sidebar() as element(div)? {  
  let $view := $config:CONTEXT/*:view  
  return  
    if ($view = ("help", "contact", "terms"))  
    then ()  
    else  
      <div class="sidebar" arcsize="5 5 0 0">  
        <div class="..." id="...">...</div>  
        <div class="sidebar-shadow"><div>...</div></div>  
        { if ($view eq "detail")  
          then xdmp:apply($config:remove-facet)  
          else if ($view = ("search", "intro"))  
          then xdmp:apply($config:facets)  
          else ()}  
      </div>};
```

Extension  
point

# Let scripters exploit XML technology for dynamic documents



# Tag libraries in 10 seconds

## Tag documents assemble tag libraries for dynamic documents

- Inspired by Java Tag Libraries and XSLT Simplified Stylesheets
- Tag library = binds data producers or page components to tag vocabulary
- Tag document = document vocabulary (HTML, XSL-FO) + tag vocabulary + expression language
- Preprocessed to standard XQuery or XSLT for execution on any processor

## Goals

- Separation of concerns: layout vs data; assembly vs component
- Reuse: page fragments, page production logic
- Collaboration: between page scripters and library developers

Hello, World

---

Insert a dynamic greeting with the current time

## Hello, World – tag library

```
<xst:taglib xmlns:fn="..." xmlns:xs="..."  
  xmlns:xst = "..."  
  xmlns:hello = "..."  
  version="0.2">
```

```
<xst:tag name="hello:welcome" as="xs:string">  
  <xst:param name="place" as="xs:string"/>
```

```
  <xst:content select="fn:concat(  
    'Welcome to ', $place,  
    ', where the time is ', fn:current-time()  
  )"/>
```

```
</xst:tag>
```

```
</xst:taglib>
```

Tag definition:  
**hello** vocabulary has  
**welcome** tag with  
**place** parameter

# Hello, World – tag document

```
<html xmlns      = "..."  
      xmlns:xst  = "..."  
      xmlns:hello = "..."  
      xst:version = "0.2">  
<xst:import href=".../hello.xstag"/>  
<head>  
  <meta http-equiv="Content-Type" content="..."/>  
  <title>Hello, World</title>  
</head>  
<body>  
  <p style="...">  
    <hello:welcome place="Tag Libraries"/>  
  </p>  
</body>  
</html>
```

Tag call

## Hello, World – generated dynamic document

```
<html xmlns="...">
```

```
<head>
```

```
  <meta http-equiv="Content-Type" content="..."/>
```

```
  <title>Hello, World</title>
```

```
</head>
```

```
<body>
```

```
  <p style="...">
```

```
    Welcome to Tag Libraries, where the time is 07:25:15-07:00
```

```
  </p>
```

```
</body>
```

```
</html>
```

**Welcome to Tag Libraries, where the time is  
07:25:15-07:00**



Generate a list of links for the articles in a feed

## Feeds – tag library

```
<xst:tag name="feed:list" as="node()*">
  <xst:param name="feeds" as="document-node()*"/>
  <xst:param name="onItem" as="handler(...) as node()*"/>
  ...
  <xst:for-each name="item" select="$feeds/rss/channel/item">
    <xst:sort select="$item/fn:string(title)"/>
    <xst:call handler="$onItem">
      <xst:with-param name="number" .../>
      <xst:with-param name="uri" .../>
      <xst:with-param name="date" .../>
      <xst:with-param name="title" .../>
      <xst:with-param name="description" .../>
    </xst:call>
  </xst:for-each>
</xst:tag>
```

Handler call:  
fragment  
parameterized  
and output  
for each feed

## Feeds – tag document (1 of 3) – for basic HTML

```
<body>
  <xst:variable name="feeds">
    <feed:get>
      http://www.reddit.com/.rss
      http://feeds.feedburner.com/TechCrunch
    </feed:get>
  </xst:variable>
  ...
  <ul>
    <feed:list feeds="{ $feeds}" xst:handles="...">
      <li><a href="{ $suri}">{ $title}</a></li>
    </feed:list>
  </ul>
</body>
```

Tag result  
captured in  
variable

Handler for  
each feed

## Feeds – generated dynamic document (1 of 3) – for basic HTML

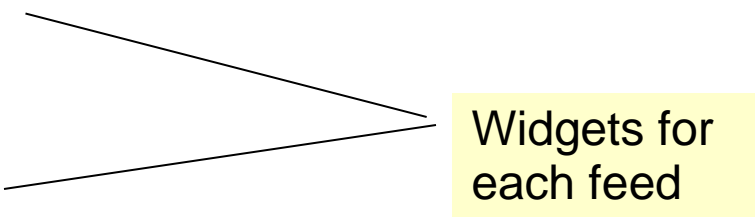
```
<h1 style="...">Feeds</h1>
<ul>
<li><a href="...">15 Years Of Photo Sharing Exits ...</a></li>
<li><a href="...">AdSense For Images Pixazza...</a></li>
<li><a href="...">Amazing Video Of Google All-Hands ...</a></li>
<li><a href="...">Another animated film I made ...</a></li>
<li><a href="...">Ant vs. Spider...just wait for it.</a></li>
...
</ul>
```

### Feeds

- [15 Years Of Photo Sharing Exits Vs. YouTube](#)
- [AdSense For Images Pixazza Surpasses 100M Unique Monthly Visitors; Hires CFO](#)
- [Amazing Video Of Google All-Hands Meeting Circa 1999: There's Silly String Everywhere](#)
- [Another animated film I made, Sebastian's Voodoo. This one got over 100 awards in festivals and got me the Disney job](#)
- [Ant vs. Spider...just wait for it.](#)

Combine the feeds with widgets or produce a PDF

```
<widget:accordion-list>
  <feed:list feeds="{ $feeds}" xst:handles="...">
    <widget:accordion-summary>
      { $title }
    </widget:accordion-summary>
    <widget:accordion-detail>
      <p>Source:
        <a href="{ $suri}">{ $suri}</a><br/>
        <feed:trim>{ $description}</feed:trim>
      </p>
    </widget:accordion-detail>
  </feed:list>
</widget:accordion-list>
```



Widgets for  
each feed

## Feeds – generated dynamic document (2 of 3) – for widget HTML

```
<div id="accordion">
<h3> <a href="#">15 Years Of Photo Sharing ...</a> </h3>
<div>...</div>
<h3> <a href="#">AdSense For Images Pixazza ...</a> </h3>
<div>...</div>
<h3> <a href="#">Amazing Video Of Google ...</a> </h3>
```

```
<div> <p>Source: <a href="#">
sWhat an amazing blas
today — burdened by c
and countless legal bat
more humble beginning
```

▶ 15 Years Of Photo Sharing Exits Vs. YouTube

▶ AdSense For Images Pixazza Surpasses 100M Unique Monthly Visitors; Hires CFO

▶ Amazing Video Of Google All-Hands Meeting Circa 1999: There's Silly String Everywhere

Source: <http://feedproxy.google.com/~r/Techcrunch/~3/hE5hs7CxXtU/>

sWhat an amazing blast to the past. Google is a massive company today — burdened by corporate politics, layers of management, and countless legal battles — and it's sometimes easy to forget its more humble beginnings just over a decade ...

```
<block ...>Table of Contents</block> ...  
<feed:list feeds="{ $feeds}" xst:handles="...">  
  <basic-link internal-destination="{ $number}" ...>  
    { $title}</basic-link>  
</feed:list>
```

```
<feed:list feeds="{ $feeds}" xst:handles="...">  
  <block id="{ $number}" ...>{ $title}</block>  
  <basic-link external-destination="{ $uri}" ...>  
    { $uri}</basic-link>  
  <block ...>{ $description}</block>  
</feed:list>
```

Summary in  
TOC and  
detail in pages



# Feeds – generated dynamic document (3 of 3) – for XSL-FO

## Table of Contents

—

[15 Years Of Photo Sharing Exits Vs. YouTube.....](#) 5

[AdSense For Images Pixazza Surpasses 100M U Hires CFO.....](#)

[Amazing Video Of Google All-Hands Meeting Circa 1999: There's Silly String Everywhere.....](#)

[Another animated film I made, Sebastian's Voodoo awards in festivals and got me the Disney job....](#)

[Ant vs. Spider...just wait for it.....](#)

## Amazing Video Of Google All-Hands Meeting Circa 1999: There's Silly String Everywhere

*Date:* Wed, 27 Apr 2011 06:33:53 +0000

---

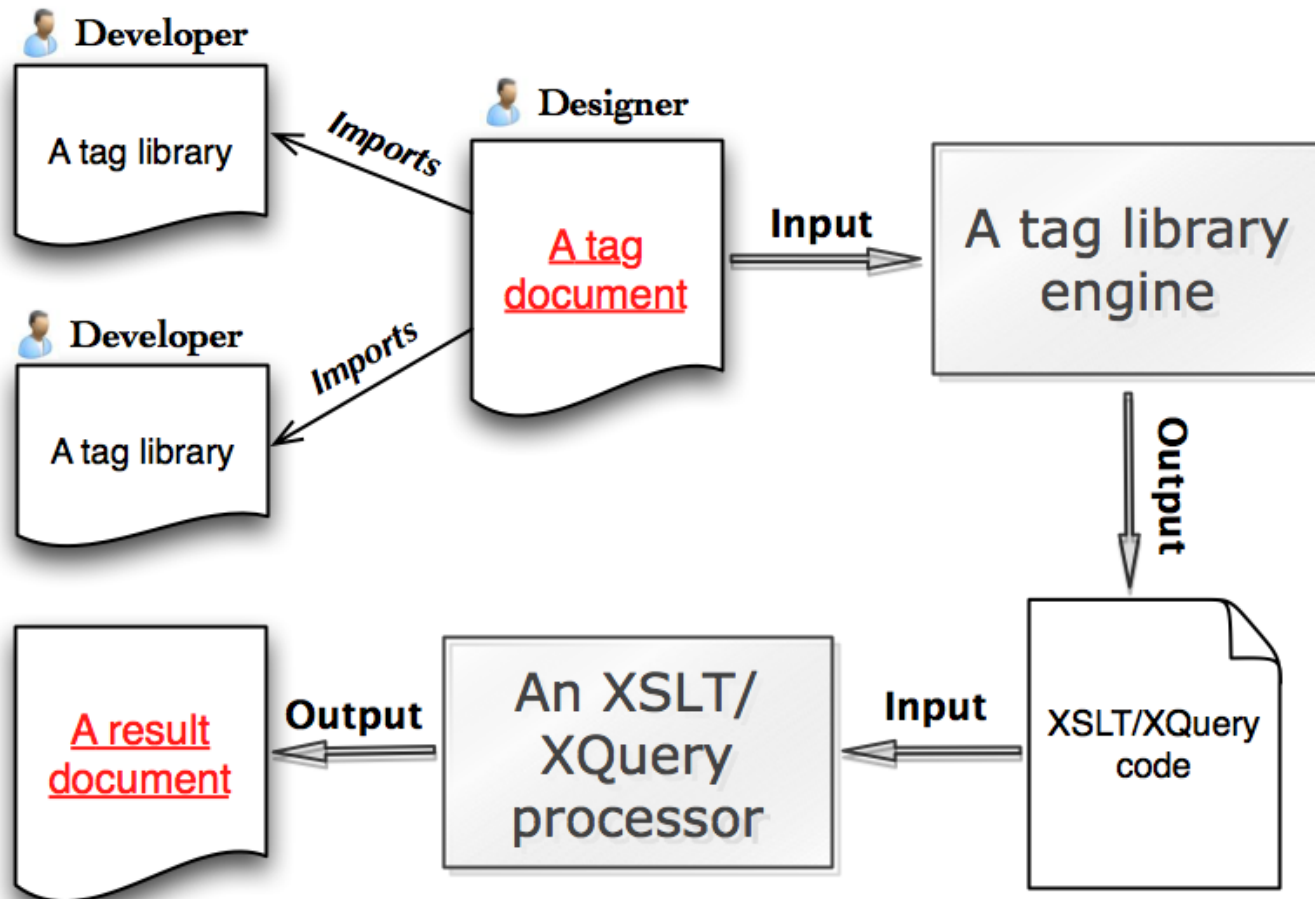
*Source:* <http://feedproxy.google.com/~r/Techcrunch/~3/hE5hs7CxXtU/>

sWhat an amazing blast to the past. Google is a massive company today — burdened by corporate politics, layers of management, and countless legal battles — and it's sometimes easy to forget its more humble beginnings just over a decade ago, when it wasn't all that different from the startups we write about every day on TechCrunch. But in the

# Summary

---

# Preprocessing and executing tag libraries



## For another day

- Calls from one tag library to another
- Calls to function libraries implemented in XQuery, XSLT, or both
- Tags defined in XQuery or XSLT
- Speculations about executing tag calls in JavaScript

## Refactoring the initial XQuery library – a plausible tag document

```
<oscar:facets>
```

```
  <oscar:onCategory xst:handles="...">
```

```
    <div class="category ... ">
```

```
      <h4 title="Collapse {$cat-title} category">{$cat-title}</h4>
```

```
      <ul>{$show-subjects} ...</ul>
```

```
    ...
```

```
  <oscar:onSubject xst:handles="...">
```

```
    <li>
```

```
      <a href="{ $href }">... {$title} ...</a>
```

```
      <i>({ $count })</i>
```

```
    ...
```

```
  <oscar:onChiclet xst:handles="...">
```

```
    <div class="facet" title="Remove {$title}">
```

```
      <a href="{ $href }" class="close">...</a>
```

```
    ...
```

### Tag documents assemble tag libraries for dynamic documents

- Separation of concerns: layout vs data; assembly vs component
- Reuse: page fragments, page production logic
- Collaboration: between page scripters and library developers

### Open Source, work in progress

- Finalizing fine points on the syntax
    - Not all tests passing yet in both XQuery and XSLT
    - No downloads yet
- <http://code.google.com/p/xstag/>

erik.hennum@marklogic.com