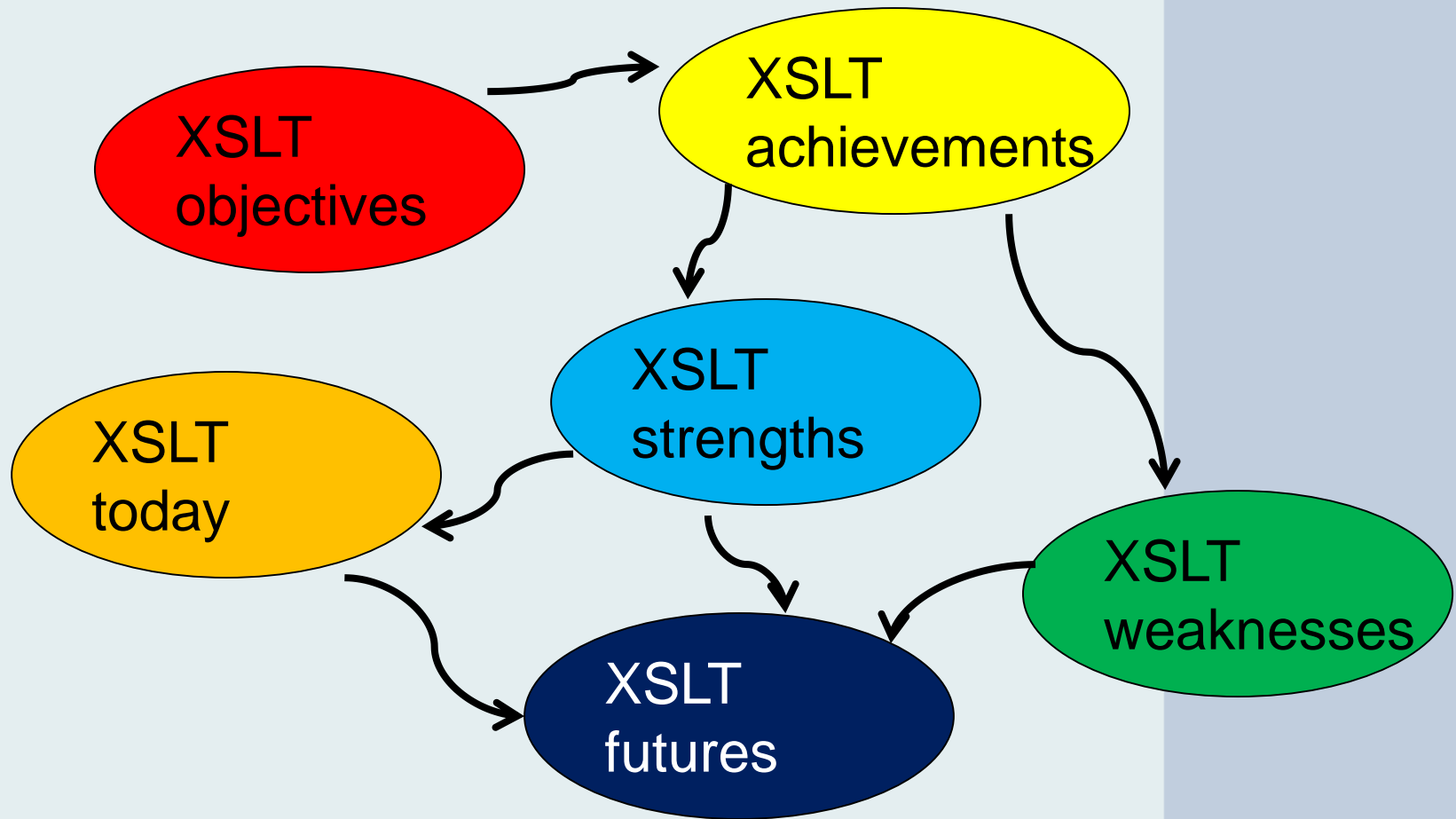


# The Future of XSLT

Michael H. Kay

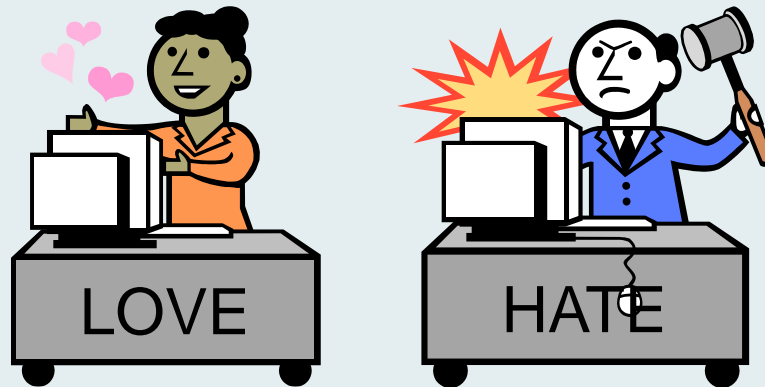
# Road map



# Where did we come from?

- SGML, Publishing, Typesetting, Device independence, DSSSL.
- XML, Data+Documents, Transformation
- XML on the browser
- XML server-side
  - in the publishing workflow
  - for content delivery

# User reaction to XSLT



# The good, the bad, and the ugly

XML syntax

Cross-platform

Declarative

no side effects

XML data  
model

rule-based

# XSLT on the browser

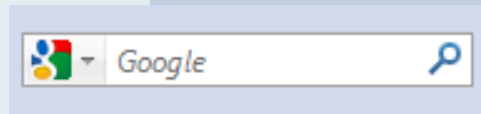
- Early false start by Microsoft (WD-xsl)
- Soon corrected with MSXML 3
- Other browser vendors slow to follow
- Only just becoming viable 10 years later
- No XSLT 2.0 implementation
- No support for “web 2.0” or “ajax”
- Little enthusiasm

# Why did this happen?

- It's all about
  - power
  - money
  - market share
  - glitz

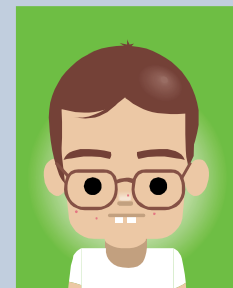


- Browser vendors control the game
  - collectively, not individually
- They want more eyeballs
  - 94% of Mozilla's revenue comes from



# What about the web developer?

- What does client-side XSLT offer?
  - portability hassles (until recently)
  - CSS can handle most of the rendering
  - Need to escape to Javascript for
    - anything interactive
    - forms
    - AJAX
  - one-way traffic (needs XForms for the return trip)
  - no technical advances since 2001





So, let's stick to our comfort zone...

XSLT on the server

# Server-side XSLT

- Unlike the browser, this is a free economy
- But investment is low because prices are low
- Vendor strategies
  - Do it for fun
  - Give it away and hope to make money on something else (Altova)
  - Bundle it as part of something expensive (IBM, Intel, MarkLogic)
  - Do it cheaply, sell it cheaply (Saxonica)

# XSLT 2.0 Processors

Product	Status
Saxon	first and still dominant. Java and .NET
Altova	widely deployed as a development tool
Gestalt	hobbyist project, now abandoned
Intel	ships as part of ExpressWay
IBM	ships as part of Websphere
MarkLogic	XQuery offshoot, part of database
XQSharp	XQuery offshoot, freestanding (.NET)

Nothing for the browser

Nothing for the open-source LAMP stack

## STRENGTHS

- Effective and efficient XML processing language
- Wide recognition
- Client+Server
- Multi-vendor, good interop

## WEAKNESSES

- Difficult to learn
- Some hostility
- Some limitations
- No longer the latest fashion
- Implementation on the browser has stalled
- Availability of 2.0 on the server still patchy

## OPPORTUNITIES

- Top end: scaleability
- Parallel processing
- On the browser: an alternative to Javascript
- Distributed Applications

## THREATS

- XML itself faces competition
- Browser vendors dropping support
- Languages with better XML support (Linq, Scala)
- XQuery more fashionable in academia

# What does XSLT 3.0 offer?

- Streaming of large documents
- Separate compilation of large stylesheets
- Extra data types for managing complex data
- In short:
  - very useful stuff for high-end XSLT users who are stretching the limits
  - not much to attract people away from Java / JavaScript / PHP / RoR etc
- But who will implement it?

# On the browser

The browser vendors control everything; no one else can make things happen

Javascript is now so powerful and fast that it can be used as a VM for other languages

# A vision

- User interface programming becomes more rule-based and declarative
- The browser becomes part of the cloud: applications run where they choose, not where the developer puts them
- The whole application shares a common type system
- Developers don't have to worry about all the low-level details of web programming (history, cookies, cross-site scripting, injection attacks)



# A step towards that goal...

- XSLT 2.0 on the browser
  - not just XML-to-HTML conversion
  - full user-interface support
  - events handled by templates
  - seamless AJAX support
  - XForms integration
  - no Javascript needed
- Can be achieved by cross-compiling Java processors using GWT

# Conclusion

- XSLT has critical mass
- But it's losing momentum
  - it needs a mid-life kicker
- In particular, it hasn't moved forward on the browser in 10 years
  - it has remained a Web 1.0 technology
  - but there are great opportunities