

# **A Look Into the Future: SVG and related XML standards**

*Jon Ferraiolo  
Jon Ferraiolo Consulting  
15 July 2002*

# SVG's Virtues

- **Open standard**
  - Not just an openly published specification
- **Richness**
  - Graphics, interactivity, animation, etc.
  - Wide applicability
    - Graphics design, engineering, mapping, user interface, ...
- **Compatibility**
  - XML, DOM, XSL, SMIL, XHTML, etc.
  - Allows consolidation with other XML langs

# What We've Learned from SVG Version 1

- **Some things could be easier:**
  - GUIs/Forms
  - Text flows
  - Progressive rendering (animations, slideshows)
  - Mapping (path data, zooming, etc.)
  - Print workflows (CMYK/spot)
  - Symbols (limitations with <use>)
  - Dynamic layout (boxes just big enough to fit text)
- **How good is your JavaScript?**

# Help is on the way

- **From the W3C (1.2 or 2.0):**
  - Text flows
  - SVG+XForms
  - SVG+SMIL
  - Improved streaming
  - SVG for print
  - Mapping
  - More powerful <use>
- **From implementers?**
  - Java integration would be killer in the enterprise
  - J2ME is installed on nearly every handheld

# **True cross-media interoperability is crucial**

- **Cross-media is a key to the future**
  - Back to the Future
    - Olden times: many flavors of NN and IE
    - Today: nearly everyone runs Win/IE
    - Future: many flavors of handhelds and desktops
  - Huge rewards for those who solve the cross-media problem

# The New Document

- **The medium is the message**
  - Legacy notion of “document” comes from limitations of paper as a medium
    - Paper documents represent a single snapshot of a set of information presented in a single visual representation
  - But in the new world, many constraints have gone away
    - Two-way communication, not just one way
    - Multiple views of the information, not just a single presentation
    - Connected to the entire world (e.g., can grab more info as needed)
    - Immersive and live documents (e.g., animations)

# Examples of the New Document

- **Example: Maps**

- Paper maps

- Pick a single scale factor and a single level of detail
    - Usually includes a single snapshot of supplemental data: street index, landmarks, etc.

- Electronic maps

- User chooses scale factor and which features to see
    - Additional geographical info can be downloaded as needed
    - Directions !! (Clearly not possible with paper maps)
    - User can update the database

# Examples of the New Document

- **Example: Maintenance manuals**

- Paper:

- Just a single snapshot of information at a particular moment in time

- Electronic:

- Document created on-the-fly with current information and gets re-updated via server push
    - User can update the database

# **The New Document must be cross-media**

- **Today, businesses need to present information and interact with users across multiple types of media:**
  - Paper
  - Desktop PCs
  - PDAs
  - Cell phones
  - Telephone

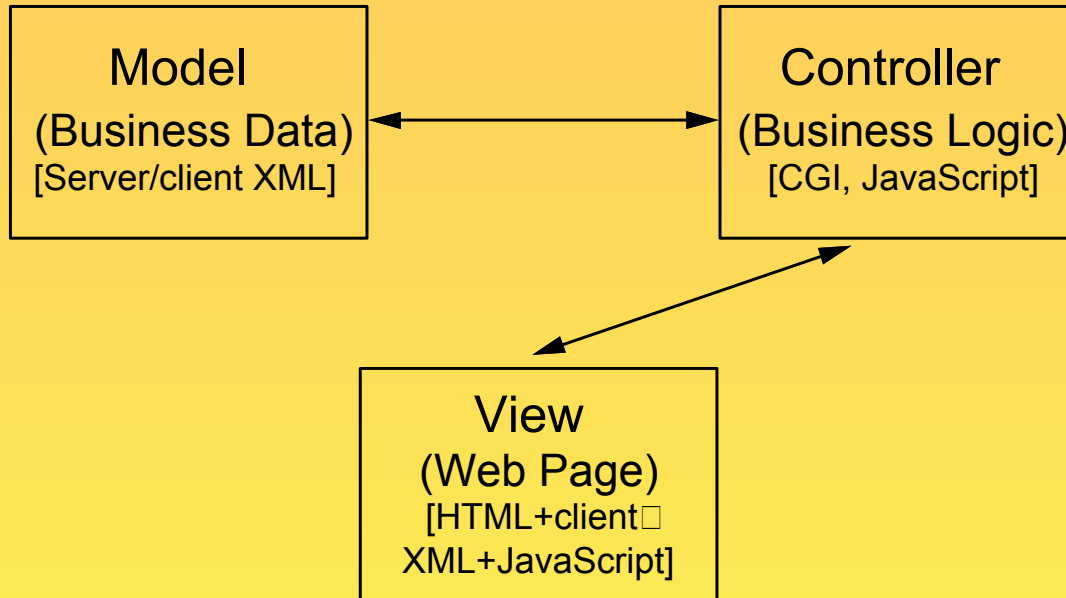
# **Approaching the New Document and Cross-Media**

- **Need to break down the New Document into logical pieces**
- **Model/View/Controller**

# Model/View/Controller

- **Model:**
  - Data in raw form (expressed in XML, of course)
  - Some combination of server-side and client-side
- **View:**
  - Controls presentation and interaction with user
  - Different presentation depending on device
- **Controller**
  - Business logic
  - Some combination of server-side and client-side
  - SQL, Perl, JavaScript, Java, C#

# Model/View/Controller: HTML

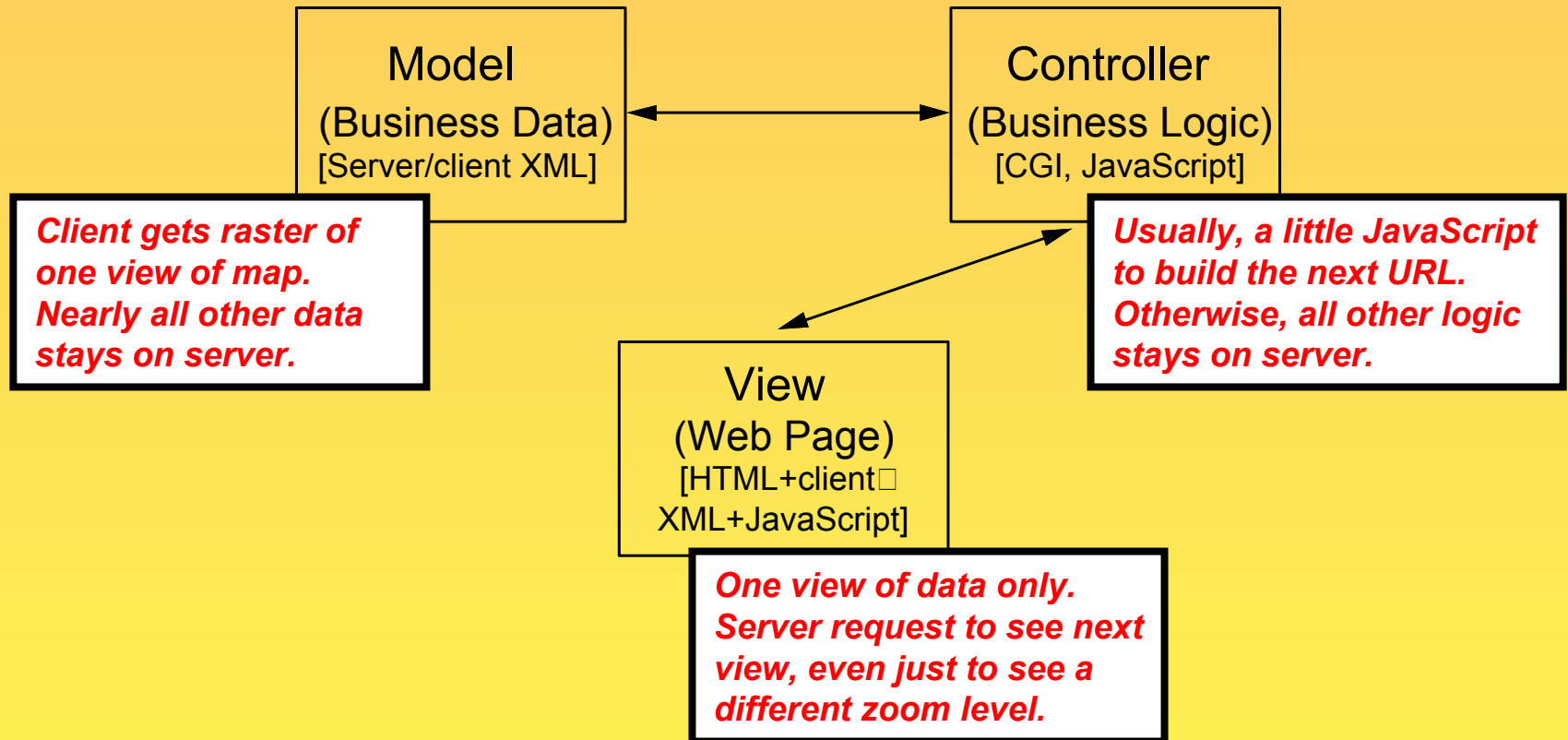


**HTML Limitations: Presentation, timeline, interactivity**

# For text applications, HTML goes a long way

- **Positives:**
  - Highly scalable
  - Handhelds can just show the data
  - User interaction possible via HTML Forms
  - Often, you can get high-quality presentation
- **Negatives:**
  - No graphics
    - Just upright text and rasters
    - Perfect answer if that's all you need
  - No timeline
  - Limited interactivity

# HTML Example: Mapping



**HTML Limitations: Presentation, timeline, interactivity**

# Cross-media graphics: Flash wants to be the solution

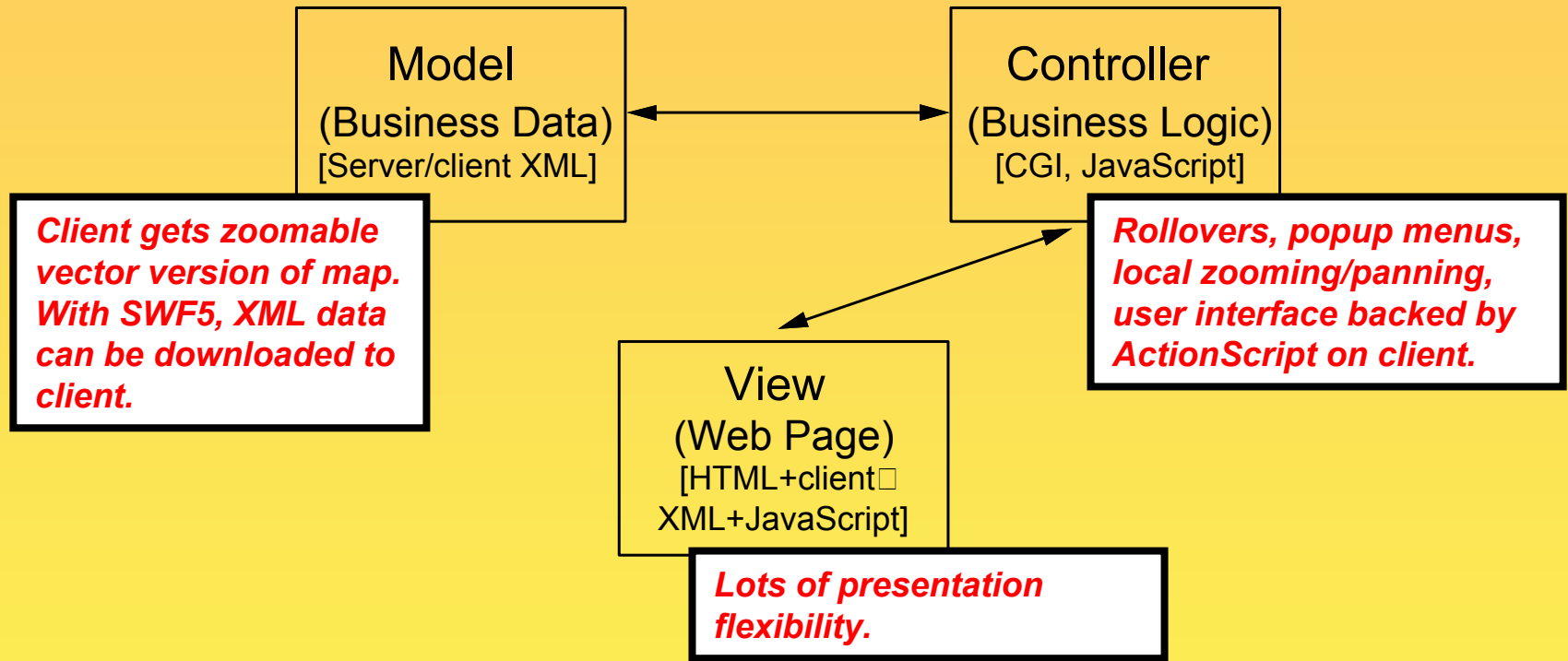
- **Positives:**

- Rich graphics
- Timeline
- Forms (with Flash/MX)
- Embedded XML
- Highly scalable

- **Negatives:**

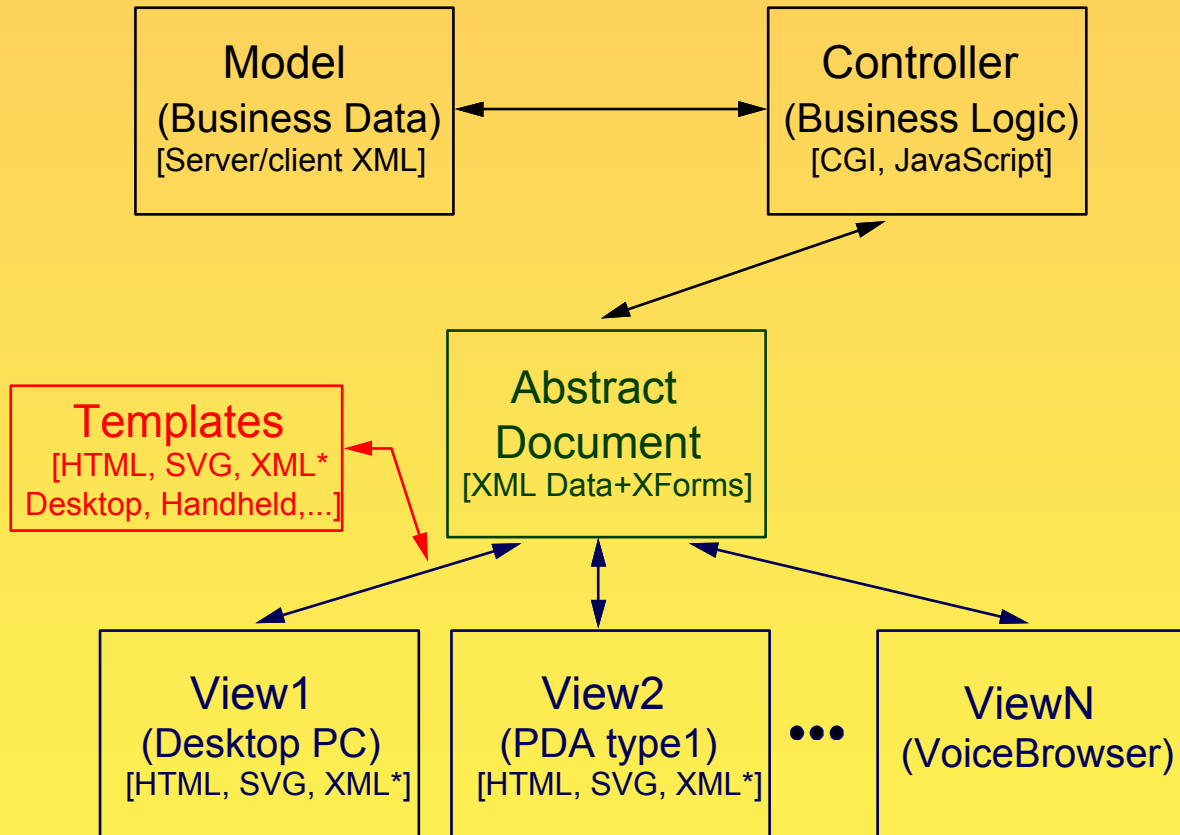
- No fallback for constrained devices
  - No dynamic layout features
  - What happens for devices that can't run latest Flash player?
- Idiosyncratic == high cost to develop and maintain
- Proprietary, a strong negative for governments and enterprise markets

# SWF or SVG Example: Mapping



**SWF Limitations: Proprietary, idiosyncratic, not XML, not text, doesn't leverage other standards, expensive to deploy and maintain, etc**

# Model/View/Controller: New Document



# Where Does SVG Fit In?

- **SVG is about presentation**
  - Therefore, its main use is View
    - But it can carry data and business logic to the client
  - Graphically rich page templates
    - This is really key – SVG has huge value in server workflows
- **Primary presentation language:  
SVG vs. HTML vs. ...**
  - Web pages in SVG [+XForms+SMIL+...] ???
    - In many cases, you can do it all with SVG today (and JS)
    - With SVG 1.2/2.0, this will become much easier
  - Web pages in HTML [+SVG+SMIL+....] ???
    - HTML for mostly text presentation
    - SVG for graphics or sophisticated UI

# Conclusion

- **Today's connected world requires a new definition of "document"**
- **Cross-media is here to stay**
- **When architecting solutions, think Model/View/Controller**
- **SVG, HTML, (XForms, SMIL, ...) play key roles in Viewing/Presentation and templates**