Towards a Generic XML Content Presentation Model

Michael Pediaditakis (mp49@kent.ac.uk)
David Shrimpton (dhs@kent.ac.uk)

Computing Laboratory
University of Kent, U.K.
Applications VS Documents

- Web applications
  - Used for:
    - Extended presentation and interaction functionality.
    - Precise presentation control
  - Problems
    - Incompatible processing models
    - Generally device specific
    - No common representation

- Our view
  - A habit of the pre-XML era
  - XML can represent any type of information
  - Browsers already provide a wide range of functionality
  - We need a generic XML content presentation model
All processing steps for:
- Compound XML documents
- Variety of capabilities/preferences
- Variety of independent languages
- Rich functionality

A proposal
- Two processing steps
  - Browser
    - Presentation of a fixed set of languages
  - Pre-Processor
    - Transformation of an open set of languages
Pre-Processing: XMLPipe

- Transformation association
  - With the language not with the document
- Variety of devices / preferences
  - Device profiles based on key-value pairs
  - Associations based on profile expressions
- Integration model
  - Set of handled constructs for each language
    - Specifies the integration points
  - Syntax integration based on the handled constructs
  - Transformation association with handled constructs
    - Recursive document sub-tree transformations
<document xmlns="..." xmlns:imp="..." xmlns:alt="..." xmlns:xl="...">
  <title>Example</title>
  <section>
    <title>The root type</title>
    <p>The root language allows <em>emphasized</em> text, images: <img href="xmlPipe.gif"/> and ...</p>
  </section>
  <section>
    <title>Mixed namespaces</title>
    <p>...capabilities sensitive content:
      <alt:alt>
        <alt:case test="http://.../XMLPipe/Terms#deviceType = mobile">
          This is a mobile,
        </alt:case>
        <alt:case>This is not a mobile,</alt:case>
      </alt:alt>
      <em xl:type="simple" xl:href="http://www.cs.kent.ac.uk">links</em>
      etc</p>
  </section>
</document>
Example

1 The root type
The root language allows emphasized text images and nested sections.

1.1 Nested section
This is a nested section

2 Mixed namespaces
Imported content: Text node 1 Text node 2, capabilities sensitive content. This is not a mobile, Unix, and others...
XMLPipe Example (Mobile)
Browser architecture

- Predefined set of languages
  - Feasible at an appropriate level of abstraction
  - Profile based integration
  - Multiple devices: modules, device specific implementation
- Constraint based orchestration
Web apps VS documents:
- They are the same

Web app functionality:
- The basic presentation abstractions and an extension mechanism

Declarative VS Imperative:
- Declarative layers implemented imperatively

Profiles?
- Yes for the natively supported languages.
- No for the open set of languages

Inter-language event propagation:
- Well defined in terms of a profile definition

Generic extension architecture:
- Yes.
- Otherwise specifications will have to be continuously updated.