CSS Layout Engine for Compound Documents

Mikko Pohja and Petri Vuorimaa
Helsinki University of Technology, Finland

mikko.pohja@hut.fi
Outline

- Introduction
- Requirements of the layout engine
- Implementation of the layout engine
- Conclusions
Introduction

- UIs of Web applications are defined by XML
- Applications are accessed by various devices
- Documents can consist of several technologies
  - Each is designed for a certain purpose
    - XHTML, XForms, SVG, SMIL, etc.
  - Compound Documents
  - Easy to combine in source, must be decided for functions and layout
- XML user agent must be able to handle all the formats
  - Layout engine is vital component of an XML user agent
- We represent requirements and implementation of a layout engine
Requirements (1/2)

- Must support CSS
  - CSS is used to style the documents on the WWW
- Must be able to lay out the compound documents
  - The XML documents may consist of several XML specifications. The layout engine must be able to handle all kind of combinations
- Supports temporal dimension of the documents
  - To enable the multimedia presentations through XML, the layout engine must reflect to temporal changes of a document
Requirements (2/2)

- Supports dynamic DOM operations
  - Since DOM can be modified dynamically, it is vital that the layout engine can reflect the changes efficiently
- Operates in the desktop computers, in the mobile devices, and in the digital television set-top boxes
  - The Web applications are accessed by the various devices nowadays
- Cooperates with the other components of an XML user agent
  - Cooperates with the other components of an XML user agent
Implementation

- Implemented in Java programming language
  - Easy to fulfill the operation environment requirement
- Integrated into open-source XML browser X-Smiles
  - Supports several XML specifications
  - http://www.x-smiles.org
Overview of the Implementation
Flowchart of the Layout Engine

XML Doc

Create DOM

Create Views

Layout

Paint

Resize

Scroll

Change
The Layout Process

- Calculates the sizes of the views
- Set the positions of the views
- Both are done during a single process
- Process starts from the root and traverses through whole view tree
  - Parent gives an origin to a child
  - Child calculates its own size
  - Parent updates its size according to the child's size
- View's position is stored as absolute coordinates
  - Painting is done with absolute coordinates
Laying out

Document source

```xml
<p>
Paragraph with some text and <a href="target">three word link</a> refering to target.
</p>
```

Document layout

Paragraph with some text and three word link refering to target.
Compound Documents

- Can lay out documents using single layout
  - E.g., XHTML + XForms
- Can preserve a region for other layout model
  - E.g., XHTML + SVG
- Overlay layout has been left as a future work
  - Alpha blending
Temporal Control

- To support multimedia applications
- Through Synchronized Multimedia Integration Language (SMIL)
  - Layout engine can be a media element in a SMIL document
- Through Timesheets
  - Like style sheets but controls temporal dimension
  - Assigns temporal relations between the elements
  - Sets styles for the elements in temporal manner
Integration into X-Smiles Browser

- DOM – View interface is bidirectional
- CSS Engine provides styles for Layout Engine
- Uses Java AWT UI Toolkit
  - Both Java Swing and HAVi are based on AWT
Platforms

- Can be run on
  - J2SE 1.2 or higher
  - Personal Java 1.1 and Java 2 Micro Edition (J2ME) Personal Profile
  - Multimedia Home Platform (MHP)
Performance

Document Processing Time

- Time (s)
  - 0.00
  - 0.05
  - 0.10
  - 0.15
  - 0.20
  - 0.25
  - 0.30
  - 0.35
  - 0.40
  - 0.45
  - 0.50
  - 0.55
  - 0.60

- Document size (KB)
  - 5
  - 11
  - 22
  - 43
  - 86

- Painting
- Laying out
- Creating views
- Preprocessing
Conclusions

- We have defined requirements for a novel layout component of an XML user agent
- We were able to implement such a component
- Java is a good alternative to develop cross-platform applications