4. Introduction to Style Sheets

- Discussed recently:
  - Programmatic manipulation of documents
- Now a more human point of view:
  - How to specify the visual representation of structured documents?
- Concepts, properties and requirements of style systems on a general level
  - later examples of concrete style languages (CSS, XSL)

Introduction to Style Sheets


- Declarative markup of structured documents indicates only the syntactic structure
  - no semantics (processing, formatting, …)
- Electronic style sheets

Style Sheets?

- (Tyylisivu tai tyylisäännöstö)
- In traditional publishing: set of rules about diction and language for some manuscript
- Electronic style sheets
  - deal with graphical layout of documents
    - In future with aural properties, too?
  - setting and changing of properties controlling layout and appearance of document content
  - define a mapping from documents
    - (structure+content) to external representation on a presentation medium (paper, screen, audio, …)

Why Style Sheets?

- Separation of content and presentation is a basic "dogma" in structured documenting
  - supports maintenance and multi-use of documents
  - (unnecessary complication for single-purpose or single-use documents)
- Relationship between documents and style sheets is many-to-many
  - single style for many documents
    - supports manageability (of, say, consistent look and feel of a corporate Web site)
  - many styles for a single document
    - supports multiple output media (print, different Web clients, handheld devices, …)

Tasks of a Style Sheet

- Style sheet guides the transformation of
  - descriptive markup into formatter input (stream of formatting commands and text)
    - called transcription by Brüggemann-Klein & Wood
- "Styling = transforming + formatting"
  - matches Extensible Stylesheet Language:
    - XSL = XSLT (transformation language)
    + XSL FO (formatting language)

Formatter (muotoilin, muotoiluohjelma)

- Formatter is a device (program) capable to format a document (obviously!), which involves
  - hyphenation (breaking of character sequences)
  - typeface attribution (characters into glyphs)
    - e.g., single glyph [for " : "]
  - line breaking (assembling glyphs into lines)
  - page breaking (grouping lines onto pages)
Process of Transformation (muunnos)

Structured document

Transformation

Style sheet

Formatter input

TeX

FO

(XSL formatting object tree)

Formatter input

Style sheet

Formatter

Display/play the document on output medium

Printer driver

Display driver

Process of Formatting (muotoilu)

Style sheet is a set of style rules
- Attach transformations to document elements
- Most style rules in practice structure-based
  - associated to instances of element types in the document structure tree
- Also grammar-based style rules have been studied
  - associated to occurrences of element types in the document grammar

Style Rules

Transcription types

Fundamental operations of style rules
- calls of parameterised formatting tasks
- generation of text
- automatic numbering
- rearrangement of elements

Application of transcriptions can depend on the context the element

Context Specification (1)

General rule:
- semantically equivalent structures (instances of the same element type) should be formatted identically

Exceptions (due to conventions, aesthetics, etc)
- all paragraphs indented, except for the first one
- headings numbered 1, 2, ..., in body; but A), B), ..., in appendix
- author lists in references: "Aho and Ullman" (just a few) vs "Aho et al." (if several authors)
- indication of target element type (Table, Figure, Section, ...) for cross references
Context Specification (2)

- Need access to ancestors, siblings, descendants, targets of cross references
- Context conditions by a context predicate
  - Boolean expression built of
    » a reference to the current element
    » functions like parent, leftSibling, leftMostSibling, children, and ref
  - Rule applied if the context condition is true
- Context specification orthogonal to transcription types
  (in XSL through the XPath expression language)

Parameterised Formatting Tasks

- The most common transcription type:
- set formatting characteristics for sub-elements
  - typeface attribution (for strings or inline-elements)
  - line breaking (for paragraphs or blocks)
  - page breaking (for documents)
  - parameterised by type size, line length, indentation, page height…
- Hierarchy of elements
  - hierarchy of nested formatting tasks
  - hierarchy of nested presentation areas

Hierarchy in Formatting

Derived or Inserted Content

- boilerplate text
  - text not present in the source document
  - e.g., letter headings, © marks, bullets,…
- textual content inserted at the beginning or at the end of the current element
- table of contents, indexes
  - need to specify the source of included material

Numbering (1)

- Different schemes
  - consecutively through document
    » same numbering sequence, possibly common to a set of element types (e.g., for Theorems and Examples)
    » nested numbering for, say, nested lists
      » relative to occurrences of another element type higher in document hierarchy

Numbering (2)

- Example of different numbering schemes:
  - Section 1
    - Theorem 1, or 1.1
    - Figure 2, or 1, or 1.2, or 1.1
  - Section 2
    - Theorem 3, or 2, or 2.1
- Often via named counters with a start value and a scope (e.g., in CSS2)
  - In XSLT: special expressions for generating numbers
Sub-element Rearrangement

- Examples
  - reverse the order of titles and authors in references
  - sorting of the reference list
    → requires functions operating on textual contents
- Sub-element suppression a special case
- Rearrangement not supported by the weakest style systems (e.g., CSS)

Viewpoint on Style Languages

- Style languages should support traditional stylistic conventions discussed above
- Next a look at CSS, and later at XSL
  - How do the discussed concepts appear in the languages?
  - How do the languages support these general requirements?