4. Introduction to Stylesheets

- Discussed recently:
  - Programmatic manipulation of (data-oriented) documents
- Now a more human-oriented point of view:
  - How to specify formatting or rendering 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

(Based on Brüggemann-Klein, A., Wood, D., Electronic Style Sheets. Universität Freiburg, Institut für Informatik, Bericht 45, Januar 1992.)

- Declarative markup of structured documents indicates purely syntactic structure
  - no semantics (processing, formatting, …)
- Electronic style sheets
  - specify layout and appearance of document content
  - e.g. FrameMaker templates, Word styles, or LaTeX style files; CSS and XSL style sheets

Style Sheets?

- In traditional publishing
  - a set of rules about diction and language for some manuscript
- Electronic style sheets
  - deal with graphical layout of documents
    (In near future 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 longevity and multi-use of documents
  - (unnecessary complication for single-purpose and 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, hand-held devices, …)

Tasks of a Style Sheet

- Style sheet guides the transcription process
  - translating descriptive markup into formatter input (stream of formatting commands and text)
  - input a document tree of typed elements and symbolic cross references
- Formatter is a device (program) capable of
  - hyphenation (breaking of character sequences)
  - typeface attribution (characters into glyphs)
    - e.g., single glyph "fi" for "fi"
  - line breaking (assembling glyphs into lines)
  - page breaking (lines into pages)

Process of Transcription
Process of Formatting

- Creates a detailed description of presentation

Formatter input

- TeX, FOP, ...

(Formatter)

Formatter output

- DVI, PS, PDF

(Formatter)

Descr. of presentation

- DVI, PS, PDF

(Script of presentation)

SDPL 2001 Notes 4: Intro to Stylesheets 7

Process of Rendering

- Display/play the document on output medium

Printer driver

Display driver

SDPL 2001 Notes 4: Intro to Stylesheets 8

Style Rules

- Entries of a style sheet
- Attach transcriptions to logical elements
- Most style rules in practise structure-based
  - associated to instances of element types in the structure tree
- Also grammar-based style rules
  - associated to occurrences of element types in the document grammar

SDPL 2001 Notes 4: Intro to Stylesheets 9

Transcription types

1. calls of parameterised formatting tasks
2. automatic numbering
3. generation of text
4. rearrangement of elements

- Application of transcriptions can depend on element context

SDPL 2001 Notes 4: Intro to Stylesheets 10

Context Specification (1)

- General rule
  - semantically equivalent structures (instances of the same element type) should be formatted identically
- Exceptions (due to conventions, aesthetics, etc)
  - paragraphs indented, except the first one
  - heading numbers 1, 2, … in body vs. A, B, … in appendices
  - author lists in references
  - (“Aho et al.” vs “Aho and Ullman”)
  - indication of target element type (Table, Figure, Section, …) for cross references

SDPL 2001 Notes 4: Intro to Stylesheets 11

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 (Cf. XPath expression language in XSL)

SDPL 2001 Notes 4: Intro to Stylesheets 12
Parameterised Formatting Tasks

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

Numbering

- 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
- often via named counters with a start value and a scope

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

Subelement Rearrangement

- **Examples**
  - reverse the order of titles and authors in references
  - sorting of the reference list
    - requires functions operating on textual contents
- **Subelement suppression** a special case
- Not supported by the weakest style systems

Viewpoint on Style Languages

- Next we’ll have 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?