1 Introduction

1.1 Structured Documents

Review of basic notions
Preliminary Outline (2)

4 Styling Structured Documents I
   4.1 Essentials of Cascading Style Sheets
5 Transforming Structured Documents
   5.1 Addressing: XPath
   5.2 XSLT
6 Styling Structured Documents II: XSL
7 XML wrapping (or translating data to XML)
8 Querying Structured Documents
   - W3C XML Query Language XQuery

Methodological Goals

- Some central professional skills
  - consulting of technical specifications
  - experimenting with SW implementations
- Ability to think...?
  - to find out relationships
  - to apply knowledge in new situations
- "Pidgin English" for scientific communication

Administration

- An elective graduate-level (laudatur) special course
  - suitable for all specialisation lines (esp. CS/SWE)
- 3 cu (≈ 120 hours of work)
- Lectures March 9 – May 6, MT2/E26–27
  - Lecturer: Pekka.Kilpelainen@uku.fi
  - Assistant: Tommi.Penttinen@uku.fi

Administration: Exercises

- Exercises March 24 – May 12, MT2/E26–27
  - essential for familiarizing with the technology
  - mainly normal homework assignments, some hands-on practice; Solutions discussed in class
  - a "mini-project"
    - programming/modifying a document processing application (XML/Java/DOM/JAXP/XSLT)
    - individually or in small groups
    - to be handed-in to lecturer
    - credited like other exercises (grading based on quality by a factor in [0, 1.5])

Administration: Grading

- Course exam on Tuesday, May 18, in SL
  - minimum of 50% of exam points to pass the course
  - Grade = (12*Exam/MaxExam + 4*HomeWork/MaxHomeWork - 4)
- Opportunity to retake the exam
  - June 3 (again ≥ 50% to pass; grade with/without homework credits, whichever is better)

Material

- No single textbook
- Reports, articles
- Course home page
  - http://www.cs.uku.fi/~kilpelai/RDK04/
  - lecture notes, exercises, reference material, announcements, ...
- Possible background text:
Background Check

- Basic knowledge of structured documents and document standards
  - Course "Introduction to Document standards"?
  - HTML?
- Programming languages and concepts
  - Java? OO programming?
  - Unix/Linux, Windows?
- Formal language theory
  - Theory of Computation / "Onjelmienmin ja laskennan teoria"?
  - regular expressions, automata?
  - context-free grammars, parse trees?

Course Expectations?

1.1. Structured Documents

- Document:
  - a structured representation of information on some medium (e.g., message)
  - normally for a human reader
  - memos, manuals, articles, books, ...
  - also application-to-application messages
  - EDI (electronic data interchange)
  - "prose-oriented XML" vs "data-oriented XML"
  - possibly non-permanent, dynamically generated
  - processable or conceivable as a unit
  - (a web page vs a web site)

Text-Based Documents

- We concentrate on textual or text-based documents
  - character data major constituent of information content
  - as opposed to, say multimedia documents
- Next: Presentation vs Structure

Presentation vs Structure

- Presentation informs the human reader about the meaning of text and the role of its parts
- Markup (merkkaus): indicating the presentation or the meaning of different parts of text
  - originally hand-written annotations for the typesetter
  - nowadays primarily codes embedded in digital documents

Markup

- Procedural markup
  - formatting commands (start boldface, produce an empty line, indent 5 mm, ...)
  - proprietary word processor formats, nroff, TeX, ...
- Descriptive or generic markup
  - indicating the logical structure of text using chosen names
  - LaTeX: \begin{abstract} ... \end{abstract}
  - HTML: <TITLE> ... </TITLE>
- Markup language (merkkauskieli)
  - a fixed set of markup notations (e.g., nroff, TeX, HTML, SVG, ...)
Most liberally, any document is structured (punctuation, words, sentences, fields, …) but especially descriptively marked-up documents … especially if they adhere to a rigorous specification of structure.

Hierarchy or nesting is ubiquitous
- chapters of books, warnings in maintenance manuals, ...

Linear order essential in prose documents
- less important in documents representing data objects

Hypertext and cross-references

We’ll be mainly dealing with manipulation of hierarchical, or tree-like document structures

Next: How these are modelled?