Structured-Document Processing Languages

(3 cu/6 cp), Spring 2006
Pekka Kilpeläinen
University of Kuopio
Department of Computer Science
Pekka.Kilpelainen@cs.uku.fi

Introduction

First: Overview and Arrangements
What is this course about?

1.1 Review of Structured-Document Basics
2.1 XML & XML Documents

Goals of the Course

- To get familiar with the most important models and languages for
  - manipulating
  - representing or styling
  - transforming and
  - querying
structured documents (or XML)
- “Generic XML processing technology”
- very little about specific XML applications or commercial systems

NOT an Exhaustive Survey

- Emphasis on processing data in the form of documents, rather than describing it
- Bias in selecting course topics:
  - estimated usefulness/value
    » centrality (implying longer-lasting value)
    » maturity: Stable specifications?
  - Existing implementations?
  - Lecturer up-to-date?

Motivation?

- Academic interest in models of information processing
- Practical relevance: “eBusiness” is HOT!

Course Outline

1 Introduction
  Overview and Arrangements
  1.1 Structured Documents
2 Document Instances and Grammars
  2.1 Trees and their Grammars
  2.2 Review of XML basics: DTDs, Namespaces, Schemas
3 Programmatic Manipulation of Structured Documents (XML APIs)
  3.1 SAX
  3.2 DOM; 3.3 JAXP

Course 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 Querying Structured Documents
  - W3C XML Query Language XQuery

Methodological Goals

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

- Document
  - a structured representation of information on some medium (=message)
  - normally for a human reader
    - memos, manuals, articles, books, ...
  - also application-to-application messages
    - e.g., between client and server in Web Services
  - "prose-oriented XML" vs. "data-oriented XML"
  - can be treated as a unit
    - e.g., a web page vs a web site

- Basic knowledge of structured documents and document standards
  - Course "Introduction to Document standards"?

- Programming languages and concepts
  - Java? OO programming?
  - Unix/Linux vs Windows?

- Formal language theory
  - Theory of Computation "Ohjelmoinnin ja laskennan teoria"?
    - regular expressions, automata?
    - context-free grammars, parse trees?

- Formal language theory
  - regular expressions, automata?
  - context-free grammars, parse trees?

- Basic knowledge of structured documents and document standards
  - Course "Introduction to Document standards"?

- Programming languages and concepts
  - Java? OO programming?
  - Unix/Linux vs Windows?

- Formal language theory
  - Theory of Computation "Ohjelmoinnin ja laskennan teoria"?
    - regular expressions, automata?
    - context-free grammars, parse trees?

- Formal language theory
  - regular expressions, automata?
  - context-free grammars, parse trees?

- Basic knowledge of structured documents and document standards
  - Course "Introduction to Document standards"?

- Programming languages and concepts
  - Java? OO programming?
  - Unix/Linux vs Windows?

- Formal language theory
  - Theory of Computation "Ohjelmoinnin ja laskennan teoria"?
    - regular expressions, automata?
    - context-free grammars, parse trees?

- Formal language theory
  - regular expressions, automata?
  - context-free grammars, parse trees?

- Basic knowledge of structured documents and document standards
  - Course "Introduction to Document standards"?

- Programming languages and concepts
  - Java? OO programming?
  - Unix/Linux vs Windows?

- Formal language theory
  - Theory of Computation "Ohjelmoinnin ja laskennan teoria"?
    - regular expressions, automata?
    - context-free grammars, parse trees?

- Formal language theory
  - regular expressions, automata?
  - context-free grammars, parse trees?
Presentation vs Structure

- Presentation informs the human reader about the meaning of text and the role of its parts
- **Markup** (merkkaus) indicates the presentation or the meaning of different parts of text
  - originally hand-written annotations for the typesetter
  - nowadays primarily codes embedded in digital documents; `<Tags>`

Markup

- **Procedural markup**
  - formatting commands (start boldface, produce an empty line, indent \(5\text{ mm}\), ...)
  - 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, ...)

Structured Documents?

Most liberally, any document is structured (punctuation, words, sentences, fields, ...) but especially descriptively marked-up documents ... (e.g. well-formed XML) especially if they adhere to a rigorous specification of structure (e.g. XML+DTD)

Structure in Documents

- **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 are these modelled?