PB138 — What are markup languages?

Outline

- What are markup languages
- Motivation
- History
- Main representatives

What are Markup Languages?

- Formal (computer) languages that allow to use in addition to the normal text in natural languages also syntactically distinguishable constructs specifying the structure of the text, the meaning of parts, etc., and also allows the text to store its metadata (information about the origin, content, authorship, dating, rights used …).
- Known markup languages are the languages for the web (HTML, XML, XHTML, …),
- but also others such as typesetting formats of the TeX system, text (documentation/help)
- formatting tools for the UNIX-like systems `nroff`, `troff`.
- Their modern counterparts, such as Markdown, AsciiDoc, Mediawiki format,…
- Languages for page description for printing and presentation, namely PostScript or PDF have similar characteristics (text + markup or commands).

Recent development

- Recently, some programming languages (for web), such as JavaScript serve as markup languages mainly for data transfer between tiers in web applications — *JavaScript Object Notation*, **JSON**.
- **Markdown** or **Asciidoc** are examples of simple but powerful markups with minimum markup overhead suitable for (web) content creation and rendering.

What are Markup Languages?

- Distinguishing characteristics of markup languages in comparison to programming languages is superiority of *text* (in natural language) over the rest of the content (*markup*), so files are often referred to as *documents*.
- The preponderance of the text in natural language may not be true in specific applications.
- For example, XML is used as the format of business exchange (database, table) data, where the

marking more than text, and this has the character of a text-recorded data of other types (number, date, logical value).

The Nature of Markup

There are 3 main categories according to the nature of markup languages and method of their interpretation:

1. **Presentational markup** usually characterizes binary content embedded in text, eg. classical (older) formats for text editors.
2. **Procedural markup** indicating how the processor (processing applications) deals with the text. Usually a sequence of instructions that the sections of the text are to perform. This sequence is consequtively processed while the usual programming constructs (branching, loops, subroutines, variables) are available, eg. in *TeX*, *PostScript*, *JSON*.
3. **Descriptive markup** declaratively defines the document structure and meaning of its parts and does not specify exactly what step should be performed while processing - this is usually known by the applications, eg. in *HTML* but also *Markdown*, *AsciiDoc*.

History — Tagging without computers

Around the sixties, the concept of tagging was known only in non-PC contexts:

- The first markup language (informally) were used to processing texts in books and their typesetting.
- Concealers and typographers make the markup on the edge of the paper to indicate what font to select, to make proofreading marks etc.

Early computer applications of markup

- The first systems for computerized text processing suffered from the fact that their target printing facilities were very different and hence they must have been "programmed".
- The standard *GenCode* (author William W. Tunnicliffe) was therefore developed, which allowed to mark the general (generic) print output in the text, and a special compiler customized the output for a particular output device.
- The "real father" of markup languages is often considered Charles Goldfarb from IBM, which developed early seventies the **Generalized Markup Language** IBM GML.

Early computer applications of markup (contd)

- On the basis of these two languages was later **Standard Generalized Markup Language** SGML was created later, which in fact is not (one) language but a meta-language , ie. standard to define languages.

- A little different way was taken by the *TeX* markup language of Donald Knuth, 70s and 8Os, describing how a typesetting system should place text in printed documents.
- Frequently, a system of macros *LaTeX* (Leslie Lamport) is used instead, which adds descriptive / declarative character to TeX (for example, characterized the logical structure of the document).

Later markup standards — SGML

The first truly widespread and relatively widely applied (however, incomparable with today's popularity of XML …) was SGML.

- It evolved as a modernization of GML, then followed by formalization and subsequent adoption as an ISO standard.
- It is a metalanguage, ie. rules for the design of specific markup languages, SGML instances.

SGML

- Languages designed according to the rules of SGML are suitable for hand typing - there is less marking than later in XML.
- However, the existence of *DTD* and connection to it to describe the structure of each document were *obligatory*.
- SGML later, in the late 90s, became the basis for formulation of XML as a format easier for machine processing, not necessarily requiring to describe the structure of documents for each file.