

# Design your own XML DTD

A simple XML compiler (or XML processor) has as inputs, e.g. files containing (unstructured) documents, and at least an XML DTD, see Figure 1. In Figure 1 the first part of the text processing process applying XML technologies is shown<sup>1</sup>.

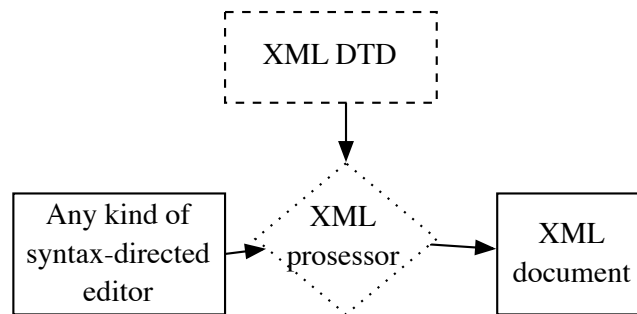


Figure 1: Inputs of an XML processor

Your current task is to design a suitable XML DTD that will implement your identified class of documents as done in the first exercise. A document class determines documents of the same type. A type of a document is given by its elements. The elements reflect the structure of the considered documents. It is not required that each element occurs in every document, but the XML DTD should design a typical structure of a document class.

To design *your* XML DTD some results of the initial task are needed<sup>2</sup>. The following task can also be done by reading the following text carefully.

## Structures modeling documents

You may have identified the following parts doing the initial task:

**Main parts** *content, chapter, section, and paragraph;*

**Subparts** *section, subsection, paragraph, image and table;*

**Elements** *title, table entry, link and paragraph.*

---

<sup>1</sup>Check the lecture notes for detailed information, if needed.

<sup>2</sup>Check on the Web <http://fnd3.fontysvenlo.org> for the initial task.

**Note that this description of main parts, subparts and elements is not necessarily complete.**

## **Relations between structures**

We also like to know something about the relationships between the identified structures. Therefore it was also asked for a short description that describes the relations between the identified structures. An excerpt of your text might be the following description:

*A document can simply consists of text. The part of text is often called content in XML<sup>3</sup>. A simple document can contain a title and text (respectively its content). A higher-structured document might be structured by chapters. Each chapter might contain sections. A section might be structured by subsections and/or paragraphs. Every section, subsection or paragraph might have a title.*

By reviewing the text that determines the relationships between main parts, subparts and elements it is possible to define rules. These rules must represent the observed relationships. These rules can be applied to define a specific syntax for your class of documents. In a first step these rules are denoted by a context-free grammar. Rules of a context-free grammar are often called productions. We studied the definition of a grammar in the course *Mathematics 3 (MAT3)*.

## **On the way to define an XML DTD**

**Definition 1** *A grammar is denoted by 4-tuple  $G = (V, \Sigma, P, S)$ , where*

*$V$  is a vocabulary (or an alphabet);*

*$\Sigma$  is a finite set of terminal symbols (or a terminal alphabet);*

*$P$  is a finite set of productions; and*

*$S$  is a start symbol (or an initial symbol).*

---

<sup>3</sup>but this term “content“ has nothing to do with the real content of a document. The real content determines the meaning of a document. And we are interested in the structures of documents!

The vocabulary contains terminal and nonterminal symbols. Nonterminal symbols can be understood as variables. These variables are replaced by some string regarding to an applied production in a derivation process. A derivation starts in the initial symbol  $S$  and ends, if the current string only consists of terminal symbols.

A grammar is context-free, if it only contains productions that are given by the relation  $P \subseteq N \times V^*$ , where  $V^*$  is string over the vocabulary and  $N$  is a set of nonterminal symbols with  $V = (N \cup \Sigma)$ .

Some rules (respectively productions) defining the structure of a document can be denoted as follows. The following list of productions represents only the main parts as given above. This means that the specification by the set of productions is also incomplete. The nonterminal CONTENT corresponds to the start symbol.

CONTENT	→	<i>text</i>
CONTENT	→	CHAPTER CHAPTERS
CONTENT	→	SECTION
CONTENT	→	PARAGRAPH
CHAPTERS	→	CHAPTER
CHAPTERS	→	CHAPTER CHAPTERS
CHAPTER	→	<i>text</i>
CHAPTER	→	TITLE SECTIONS
SECTIONS	→	SECTION
SECTIONS	→	SECTION SECTIONS
SECTION	→	TITLE PARAGRAPH
SECTION	→	<i>text</i>
PARAGRAPH	→	<i>text</i>
TITLE	→	<i>text</i>

Terms (or structures) in upper cases denote nonterminals and terms in lower cases denote terminal strings. Note that the only terminal string in an XML DTD is *text*. This *text* is later retrieved from some database or read from some file. Again, the given set of productions is incomplete. And the set of productions not necessarily match with the structures that *you* have identified.

To transfer the structure, which is described by a context-free grammar, a slightly different notation is applied in the XML DTD. Repeating an identified structure it may be denoted by operations  $\star$  and  $+$ , where  $\star$  means that a structure (or string) can be repeated zero or

more times. The operation  $+$  means that a structure (or string) must occur at least once, but can also be repeated many times. Elements carrying the “real“ content are denoted by  $\#PCDATA$  using XML. You already find a corresponding example in the lecture notes as well.

### Your current tasks

1. Review your identified parts. Is your list complete?
2. Add phrases to the text that determine the relationships between the parts, subparts and elements that you have identified.
3. Start to denote a context-free grammar that denotes the relationships between parts, subparts and elements as given above and matches with your first textual description of a structured document.
4. Identify the parts that might be repeated and the elements that might refer to *text*.
5. Try to find alternative descriptions for the repeating parts using the operations  $\star$  and  $+$ . (Hint: The operations  $\star$  and  $+$  have the same meaning as defined for denoting regular expressions.)
6. Finally, define *your* XML DTD!

Some hints:

- An additional requirement to define structured documents by XML requires that the document class denoted by the XML DTD has a unique root. This requirement is reflected by the definition of an XML DTD, in which the defined grammar has exactly one start rule with a unique start symbol.
- May you add a production like  $S' \rightarrow S$ , where  $S$  is a current initial symbol that occur in more than one rule and  $S'$  is a new initial symbol.