Chapter 7. The Style Process

7.1. Introduction

The last phase of the DSSSL processing model is the formatting process. The style language controls the manner in which this formatting is done.

The ultimate goal of the formatting process is to generate something, from the results of the previous processes, that a person can understand. Using the classic elements of typography, spacing and layout, some parts of the content are made to stand out from the rest in a way that makes grasping their meaning easy. This is a complex task with an important artistic component of harmony and composition.

The DSSSL standard does not define the entirety of this process: it divides it into phases of which it specifies some and leaves the remainder undefined.

The formatting process, as defined by the standard, begins with an SGML source document. This document can be either the original source document or the result of an earlier transformation of the original source document.

After analyzing the document, a source grove representing it is built. The output grove of the transformation process is not used directly as the style specification may define its own grove plan which may differ to that used for the transformation. This is why an intermediate SGML document is generated between the transformation process and the formatting process, instead of passing the result grove of the transformation process directly to the formatting process.
Chapter 7. The Style Process

Figure 7-1. The style process

With the style language, one defines the desired formatting of each part of the grove. This formatting is specified using a set of preestablished flow objects created by construction rules (so called because they construct the flow objects).

Flow objects are an abstract representation of tasks that the formatter must perform. Each flow object has a class, which defines the type of task, and characteristics, which permit the parameterization of the task.

Just as a grove is a hierarchical data model, the flow objects, as they are constructed, form a tree. There are flow objects that contain other flow objects. The contained objects are attached to ports of the containing objects. There are also atomic flow objects which have no ports and so may not contain other flow objects.

The flow objects only define in an abstract manner the task that the formatter must perform. The formatter must be capable of performing the tasks as specified by the flow object tree. The DSSSL specification stops at this point. The way in which the flow object tree is represented, how the tree is passed to the formatter, how the formatter works and how it is made to perform the tasks in the specified way are not defined by the standard.

Regardless, the standard specifies that the result of formatting a flow object is a rectangular box, called an area, of fixed height and width.

The style language is formed of three types of expression:

- Construction rules: specify how, given a node in the source grove, the nodes in the flow object tree should be created. See Section 7.2.1.
- Mode definitions: group construction rules together so that they are executed when the processing mode is activated. See Section 7.2.15.
- Style definitions: group characteristic specifications so they can be used as a schema. See Section 7.2.11.

7.2. Tutorial of the Style Language

7.2.1. Construction rules

Every stylesheet consists of a series of sentences that convert structural elements into formatting objects. In DSSSL, the syntax to do this is called a construction rule because