Layout Description Schema For Marmot

September 29, 2011

Layout description of a document page is a tree structure, where the leaves are characters, images and paths, and the root is the whole page. Internal nodes include textlines, paragraphs, tables etc.

This document briefly desctribes the layout schema of document page.

1 Basic Data Type

Before the introdution of object types, let's make several data types clear.

1.1 HexDouble

Double values are written in hexical strings. The hexical strings are little-endian and follow IEEE 754. Each hexical string has 16 letters.

1.2 Box

Box describes a rectangle area of the page. To describe a box, the coordinates of its top, bottom, left, right are stored in a hexical strings, seperated by space. It may not be convinient for human to read, but avoids loss of precision.

1.3 PhysicalID

PhysicalID is used to identify objects that are atomic in the page, i.e. characters, images and paths. For subtle reasons, PhysicalID may not always be integers. "xxx-yyy" are also legal form of PhysicalID, where "xxx" and "yyy" are non-negative integers.

1.4 LayoutID

LayoutID is used to identify an layout object. An LayoutID is a non-negative integer.

1.5 IDArray

An IDArray is an array of LayoutID's separated by spaces.

2 Physical Object Types

2.1 Leaf

Leaf represents the atomic object in the page. A physical Leaf has a corresponding layout Leaf.

Attribute	Data Type	Description
Label	`Label`	label of the group
PID	`PhysicalID`	physical ID
BBox	`Box`	bounding box
ClipBox	`Box`	clip box
Text	`xs:string`	text content
Font	`xs:nonNegativeInteger`	integer alias of font
Size	`HexDouble`	size of object

Table 1: Attributes of Physical Leaf

Remarks:
Legimate labels of physical Leaf are “Char”, “Image” and “Path”.
PID may have form “xxx” or “xxx-yyy”, where “xxx” and “yyy” are non-negative integers.
ClipBox is used to constrain display area of object. Even the bounding box of object may be large, the display area cannot exceed its clip box.
If obect has label “Char”, Text, Font and Size are available.
Text gives the code of character in Unicode.
Font is an integer value. The real font name is not given, but if two fonts has exactly the same name, their integer alias are equal.
Size is the font size used in this object, usually measured in pound.

If object has label “Path”, a sequence of PathOp will be contained.

2.2 PathOP

PathOP is used to describe path object.

OpType	Operand	Description
0	x y	The start point coordinate (x, y) of SubPath.
1	x y	Move the current point to the specified point (x, y).
2	x y	Connect a line from the current point to the specified point (x, y), and move the current point to the specified point.
3	x_1 y_1 x_2 y_2	Connect quadratic *Bézier* curve from the current point to point (x_2, y_2), and move the current point to point (x_2, y_2), and this *Bézier* curve uses the point (x_1, y_1) as its control point.
4	x_1 y_1 x_2 y_2 x_3 y_3	Connect a cubic Bezier curve from the current point to point $f(x_3, y3)$, and move current point to point (x_3, y_3), and this *Bézier* curve uses the point (x_1, y_1) and point (x_2, y_2) as control points.
5	r_x r_y *angle* *large* *sweep* x y	Connect an arc from the current point to point (x, y), and move the current point to point (x, y). r_x indicates the length of the long axis of the ellipse and r_y indicates the length of the short axis of the ellipse. *angle* is the rotated angle of the ellipse in the current coordinate system, with the positive value for clockwise, and the negative for counterclockwise, and when *large* is valued as 1, it indicates the corresponding arc is greater than 180 degrees, while 0 indicates the corresponding arc is smaller than 180 degrees. When *sweep* is valued as 1, it indicates the clockwise rotation from the arc beginning to the arc end, and 0 for counterclockwise rotation.
6		SubPath automatic closing indicates the current point is directly connected with the start point of SubPath via a line.

Table 2: Attributes of Physical Leaf

3 Layout Object Types

3.1 Content

Content is an generic abstraction of all content objects in a document page.

Attribute	Data Type	Description
Label	`Label`	layout role of object
BBox	`Box`	bounding box
LID	`LayoutID`	layout ID
PLID	`LayoutID`	parent's layout ID

Table 3: Attributes of Leaf

Remarks:
Layout roles include "Char", "Textline", "Paragraph" etc. We have no restriction on how the LID's are generated. So even though two textlines contain exactly the same characters, they may have different LID's. Number 0 is reserved. You should not create an object with 0 as its LID. However, PLID of object is allowed to be 0, which means the object has no parent.

3.2 Leaf

Leaf is sub-type of Content.

Attribute	Data Type	Description
PID	`PhysicalID`	physical ID

Table 4: Attributes of Leaf

Remarks:
Leaf objects are minimum components of a logical page. Every Leaf object has a corresponding physical object. By now, legimate labels of Leaf's are:

- Char
- Image
- Path

PID indicates the unique physical object which is a Leaf object at the same time. For subtle reasons, physical ID's may contain other characters than digits.1.3

3.3 Composite

Composite is sub-type of Content.

Attribute	Data Type	Description
CLIDs	`IDArray`	children's layout ID's

Table 5: Attributes of Composite

Remarks:
Composite objects are the interanl nodes of page's layout tree. By now, legimate labels of Composite's include:

- Matrix
- Formula
- Figure
- Textline
- List
- TableCaption
- TableFootnote
- TableBody
- Table
- Paragraph
- Footnote
- Body
- Header
- Footer
- Decoration

CLIDs means "children's layout ID's". Its value indiates which objects are directly contained.

3.4 Leafs/Composites

Leafs and Composites are groups of Leaf's and Composite's with same label.

Attribute	Data Type	Description
Label	`Label`	label of the group

Table 6: Attributes of Leafs/Composites

Remarks:
Leafs/Composites has no special meaning in the layout tree. They are just used to separate objects into homogeneous groups, so that the xml file is more readable. We suggest the objects be organized in topological order. That is, if object of type A consists of objects of type B, C, D, then ContentArray of B, C, D should appear before that of A.

3.5 Contents

Contents is the set of all Leafs/Composites objects.

3.6 Page

Page gives an overview of a page layout.

Attribute	Data Type	Description
PageNum	`positiveInteger`	page number
PageType	`positiveInteger`	page type
CropBox	`Box`	display area of the page

Table 7: Attributes of Page