FOP Design: Area Tree

Version 627324

by Keiron Liddle

Table of contents

1 Introduction

The Area Tree is an internal representation of the result document, representing pages and their contents. To make the concepts clearer and easier to understand, the code to implement the area tree matches the areas defined in the XSL-FO specification.

The area tree is created by the layout managers once the layout is decided for a page. Once a completed page is finished it can then be added to the area tree. From that point forward, the area tree model can then handle the new page. The data in the area tree must be minimal and independant. This means that the data uses less memory and can be serialized to an output stream if needed.

The Area Tree consists of a set of pages, which the actual implemenation places in a set of page sequences.

2 Structure

The area tree is a root element that has a list of page-viewport-areas. Each page viewport has a page-reference-area which holds the contents of the page. To handle the processing better FOP does not maintain a list at the root level but lets the area tree model handle each page as it is added.

3 Page

A page consists of a page+viewport pair.

The PageViewPort and Page with the regions is created by the LayoutMasterSet. The contents are then placed by the layout managers. Once the layout of a page is complete then it is added to the Area Tree.

Inside the page is a set of RegionViewport+Region pairs for each region on the page.

A page is made up of five area regions. These are before, start, body, end and after. Each region has a viewport and contains the areas produced from the children in the FO object heirarchy.

For the body area there are more subdivisions for before floats, footnotes and the main reference area. The main reference area is made from span areas which have normal flow reference areas as children. The flow areas are then created inside these normal flow reference areas.

Since the layout is done inside a page, the page is created from the pagemaster with all the appropriate areas. The layout manager then uses the page to add areas into the normal flow reference areas and floats and footnotes. After adding the areas for the body region then the other regions can be done layed out and added.

4 Block Areas

Block level areas contain either other blocks or line areas (which is a special block area).

A block is either positoned or stacked with other block areas.

Block areas are created and/or returned by all top level elements in the flow. The spacing between block areas is handled by an empty block area. A block area is stacked with other block areas in a particular direction, it has a size and it contains line areas made from a group of inline areas and/or block areas.

5 Line Areas

A line areas is simply a collection of inline areas that are stacked in the inline progression direction. A line area has a height and a start position. The line area is rendered by handling each inline area.

A line area gets a set of inline areas added until complete then it is justified and vertically alignedi when adding the areas. If the line area contains unresolved areas then there will be a line resolver that retains the justification information until all areas in the line are resolved.

6 Inline Areas

There are a few different types of inline areas. All inline areas have a height and width.

Unresolved areas can reserve some space to allow for possible sizes once it is resolved. Then the line can be re-justified and finalised.

Inline areas are stacked in a line area. Inline areas are objects such as character, viewport, inline-container, leader and space. A special inline area Word is also used for a group of consecutive characters.

The image and instream foreign object areas are placed inside a viewport. The leader (with use content) and unresolved page number areas are resolved to other inline areas.

Once a LineArea is filled with inline areas then the inline areas need to be aligned and adjusted to fill the line properly.

7 Repeated Areas

There are cases where the same subtree could be repeated in the area tree. These areas will be returned by the same layout managers. So it is possible to put a flag on the created areas so that the subtree data can be cached in the output. Examples of this are: static areas, table header/footer, svg.

8 Traits

A trait is information associated with an area. This could be information such as text colour or is-first.

Traits provide information about an area. The traits are derived from properties on the formatting object or are generated during the layout process. Many of the layout traits do not have actual values but can be derived from the Area Tree. Other traits that apply when rendering the areas are set on the area. Since setting the same value on every area would use a lot of memory then the traits are derived from default or parent values.

A dominant trait on a block area is set, for example font colour, so that every line area with the same dominant value can derive it. The text inline areas then get the font colour set on the inline area or from the line area or from the block area.

9 Classes

The following class structure will be used to represent the area tree.

9.1 Page Area Classes

The page area classes hold the top level layout of a page. The areas are created by the page master and should be ready to have flow areas added.

9.2 Block Area Classes

The block areas hold other block areas and/or line areas. The child areas are stacked in a particular direction.

Areas for tables, lists and block container have their child block areas stacked in different ways. These areas a placed with an absolute positioning. The absolute positioning is where the blocks are placed with an offset from the parent reference area.

9.3 Inline Area Classes

The inline areas are used to make up a line area. An inline area typically has a height, width and some content. The inline area is offset from the baseline of the current line area. The content of the inline area can be other inline areas or a simple atomic object.

10 Forward References

The Area Tree maintains a set of mappings from the reference to pages.

The PageViewPort holds the list of forward references that need resolving so that if a references is resolved during layout the page can be easily found and then fixed. Once all the forward references are resolved then the page is ready to be rendered.

To layout a page any areas that cannot be resolved need to reserve space. Once the inline area is resolved then the complete line should be adjusted to accomodate any change in space used by the area.

11 Caching

We may need to cache pages due to forward references or when keeping all pages.

This is done by serializing the Page. The PageViewport is retained to be used as a key for page references and backward references. The Page is serialized to an object stream and then all of the page contents are released. The Page is then recoved by reading from the object stream.

The PageViewport retains information about id areas for easy access.

12 Extensions

The Area Tree holds the Output Document extensions. This is information such as pdf bookmarks or other output document specific information that is not handled by XSL:FO.

It is also possible to create custom areas that extend a normal area. The actual data that is rendered could be set in a different way or depend on resolving a forward reference.

13 Area Tree Handlers

To handle different situations the handler for the Area Tree handles each page as it is added.

The RenderPagesModel sends the page directly to the renderer if the page is ready to be rendered. Once a page is rendered it is discarded. The StorePagesModel stores all the pages so that any page can be later accessed.

The Area Tree retains the concept of page sequences (this is not in the area tree in the spec) so that this information can be passed to the renderer. This is useful for setting the title and organising the groups of page sequences.

14 Status

14.1 To Do

14.2 Work in Progress

14.3 Completed

- new area tree model
- changed area tree xml format to match the area tree hierarchy