FOP Design: Layout

$Revision: 197954 $

by Keiron Liddle

Table of contents

1 Introduction....3
2 Design Issues....3
  2.1 Keep Layouts Simple....3
  2.2 Keep ID References Simple....3
  2.3 Render Pages ASAP....4
3 Layout Managers....4
4 Creating Managers....4
5 Using Managers....5
6 Page Layout....5
7 Balancing Page....6
8 Finding Break....6
9 Optimising....6
10 Multi-Column Pages....6
11 Completing Page....7
12 Line Areas....7
13 Before Floats and Footnotes....7
14 Side Floats....8
15 Unresolved Areas....8
16 ID and Link Areas....8
17 Inline Areas....9
18 Fixed Areas....9

19 Stretch Areas........ 9
20 Character Areas........ 9
21 Anchor Areas........ 9
22 Unresolved Page Numbers........ 9
23 Block Areas........ 10
24 Page Areas........ 10
25 Test Cases........ 10
  25.1 Simple Pages........ 10
  25.2 Before Floats/Footnotes........ 11
  25.3 Multicolumn........ 11
  25.4 Last Page........ 11
26 Status........ 11
  26.1 To Do........ 11
  26.2 Work in Progress........ 12
  26.3 Completed........ 12

1. Introduction

The role of the layout managers is to build the Area Tree by using the information from the FO Tree. The layout managers decide where information is placed in the area tree.

A layout manager is typically associated with an FO Object but not always.

The layout managers are in between the FO Tree and the Area Tree. They get information from the FO Tree and create areas and build the pages. They hold the state of the layout process as it builds up the areas and pages. They also manage the handling of breaks and spacing between areas.

FO Objects can have two types of properties, ones that relate to the layout and ones that relate to the rendering. The layout related properties area used by the layout managers to determine how and where to create the areas. The render related properties should be passed through to the renderer in the most efficient way possible.

The aim of the layout system is to be self contained and allow for easy changes or extensions for future development. For example the line breaking should be decided at a particular point in the process that makes it easier to handle other languages.

The layout begins once the hierarchy of FO objects has been constructed. Note: it may be possible to start immediately after a block formatting object has been added to the flow but this is not currently in the scope of the layout. It is also possible to layout all pages in a page sequence after each page sequence has been added from the xml.

The layout process is handled by a set of layout managers. The block level layout managers are used to create the block areas which are added to the region area of a page.

The traversal is done by the layout or structure process only in the flow elements.

2. Design Issues

2.1. Keep Layouts Simple

Layout should handle floats, footnotes and keeps in a simple, straightforward way.

2.2. Keep ID References Simple

2.3. Render Pages ASAP

The issue here is that we wish to recycle the Area Tree memory as much as possible. The problem is that forward references prevent pages from being resolved until the forward references are resolved. If memory is insufficient to store unresolved pages, Area Tree fragments must be serialized until resolved.

FOP developers have discussed adding the capability of using an Area Tree to render to more than one output target in the same run, which would be a complicating factor in disposal of pages as they are rendered.

3. Layout Managers

The layout managers are set up from the hierarchy of the formatting object tree. A manager represents a hierachy of area producing objects. A manager is able to handle the block area(s) that it creates and organise or split areas for page breaks.

Normally any object that creates a block area will have an associated layout manager. Other cases are tables and lists, these objects will also have layout managers that will manager the group of layout managers that make up the object.

A layout manager is also able to determine height (min/max/optimum) and keep status. This will be used when organising the layout on a page. The manager will be able to determine the next place a break can be made and then be able to organise the height.

A layout manager is essentially a bridge between the formatting objects and the area tree. It will keep a list of line areas inside block areas. Each line area will contain a list of inline areas that is able to be adjusted if the need arises.

The objects in the area tree that are organised by the manager will mostly contain the information about there layout such as spacing and keeps, this information will be thrown away once the layout for a page is finalised.

4. Creating Managers

The managers are created by the page sequence. The top level manager is the Page manager. This asks the flow to add all managers in this page sequence.

For block level objects they have a layout manager. Neutral objects don't represent any areas but are used to contain a block level area and as such these objects will ask the appropriate child to create a layout manager.

Any nested block areas or inline areas may be handled by the layout manager at a later stage.

5. Using Managers

Block area layout managers are used to create a block area, other block level managers may ask their child layout managers to create block areas which are then added to the area tree(subset).

A manager is used to add areas to a page until the page is full, then the manages contain all the information necessary to make the decision about page break and spacing. A manager can split an area that it has created will keep a status about what has been added to the current area tree.

6. Page Layout

Once the Page layout manager, belonging to the page sequence, is ready then we can start laying out each page. The page sequence will create the current page to put the page data, the next page and if it exists a last page.

The current page will have the areas added to it from the block layout managers. The next page will be used when splitting a block that goes over the page break. Note: any page break overrides the layout decided here. The last page will be necessary if the last block area is added to this page. The size of the last page will be considered and the areas will be added to the last page instead.

The first step is to add areas to the current page until the area is full and the lines of the last block area contain at least n(orphans) and at least n(orphans) + n(widows) in total. This will only be relevant for areas at the start or end of a particular reference area.

The spacing between the areas (including spacing in block areas inside an inline-container) will be set to the minimum values. This will allow the page to have at least all the information it needs to organise the page properly.

This should handle the situation where there are keeps on some block areas that go over the end of the page better. It is possible that fitting the blocks on the page using a spacing between min and optimum would give a closer value to the optimum than putting the blocks on the next page and the spacing being between optimum and max. So if the objects are placed first at optimum then you will need to keep going to see if there is a lower keep further on that has a spacing that is closer to the optimum.

The spacing and keep information is stored so that the area positions and sizes can be adjusted.

7. Balancing Page

The page is vertically justified so that it distributes the areas on the page for the best result when considering keeps and spacing.

8. Finding Break

First the keeps are checked. The available space on the page may have changed due to the presence of before floats or footnotes. The page break will need to be at a height <= the available space on the page.

A page break should be made at the first available position that has the lowest keep value when searching from the bottom. Once the first possible break is found then the next possible break, with equally low keep value, is considered. If the height of the page is closer to the optimal spacing then this break will be used instead.

Keep values include implicit and explicit values when trying to split a block area into more than one area. Implicit keeps may be such things as widows/orphans.

If the page contains before floats or footnotes then as each area or line area is removed the float/footnote should also be removed. This will change the available space and is a one way operation. The footnote should be removed first as a footnote may be placed on the next page. The lowest keep value may need to be reassessed as each conditional area is removed.

The before float and footnote regions are managed so that the separator regions will be present if it contains at least one area.

9. Optimising

Once the areas for the page are finalised then the spacing will need to be adjusted. The available height on the page is compared with the min and max spacing. All of the spacing in all the areas on the page is then adjusted by the appropriate percentage value.

10. Multi-Column Pages

In the case of multi-column pages the column breaks and eventually the page break must be found in a slightly different way.

The columns need to be layed out completely from first to last but this can only be done after a rough estimate of all the elements on the page in case of before floats or footnotes.

So first the complete page is layed out with all columns filled with areas and the spacing at a minimum. Then if there are any before floats or footnotes then the availabe space is adjusted. Then each the best break is found for each column starting from the first column. If any before floats or footnotes are removed as a result of the new breaks and optimised spacing then all the columns should still be layed out for the same column height.

11. Completing Page

After the region body has been finished the static areas can be layed out. The width of the static area is set and the height is inifinite, that is all block areas should be placed in the area and their visibility is controlled be other factors.

The area tree for the region body will contain the information about markers that may be necessary for the retrieve marker.

The ordering of the area tree must be adjusted so that the areas are before, start, body, end and after in that order. The body region should be in the order before float, main then footnote.

12. Line Areas

Creating a line areas uses a similair concept. Each inline area is placed across the available space until there is no room left. The line is then split by considering all keeps and spacing.

Each word (group of adjacent character inline areas) will have keeps based on hyphenation. The line break is at the lowest keep value starting from the end of the line.

Once a line has been layed out for a particular width then that line is fixed for the page (except for unresolved page references).

13. Before Floats and Footnotes

The before float region and footnote region are handled by the page layoutmanger. These regions will handle the addition and removal of the separator regions when before floats/footnotes area added and removed.

Footnotes and Before Floats are placed in special areas in the body region of the page. The size of these areas is determined by the content. This in turn effects the available size of the main reference area that contains the flow.

A layout manager handles the adding and removing of footnotes/floats, this in turn effects the available space in the main reference area.

14. Side Floats

If a float anchor is present in a particular line area then the available space for that line (and other in the block) will be reduced. The side float adds to the height of the block area and this height also depends on the clear value of subsequent blocks. The keep status of the block is also effected as there must be enough space on the page to fit the side float.

Side floats alter the length of the inline progression dimension for the current line and following lines for the size of the float.

This means that the float needs to be handled by the block layout manager so that it can adjust the available inline progression dimension for the relevant line areas.

15. Unresolved Areas

Once the layout of the page is complete there may be unresolved areas.

Page number citations and links may require following pages to be layed out before they can be resolved. These will remain in the area tree as unresolved areas.

As each page is completed the list of unresolved id's will be checked and if the id can be resolved it will be. Once all id's are resolved then the page can be rendered.

Each page contains a map of all unresolved id's and the corresponding areas.

In the case of page number citations, the areas reserves the equivalent of 3 number nines in the current font. When the area is resolved then the area is adjusted to its proper size and the line area is re-aligned to accomodate the change.

16. ID and Link Areas

Any formatting object that has an ID or any inline link defines an area that will be required when rendering and resolving id references.

This area is stored in the parent area and may be a shape that exists in more than one page, for example over a page break. This shape consists of the boundary of all inline (or block) areas that the shape is defined for.

17. Inline Areas

This is the definition of all inline areas that will exist in the area.

18. Fixed Areas

instream-foreign-object, external-graphic, inline-container

These areas have a fixed width and height. They also have a viewport.

19. Stretch Areas

leader, inline space

These areas have a fixed height but the width may vary.

20. Character Areas

character

This is an simple character that has fixed properties according to the current font. There are implicit keeps with adjacent characters.

21. Anchor Areas

float anchor, footnote anchor

This area has no size. It keeps the position for footnotes and floats and has a keep with the associated inline area.

22. Unresolved Page Numbers

page-number-citation

A page number area that needs resolving, behaves as a character and has the space of 3 normal characters reserved. The size will adjust when the value is resolved.

23. Block Areas

When a block creating element is complete then it is possible to build the block area and add it to the paprent.

A block area will contain either more block areas or line areas, which are special block areas. The line areas are created by the LineLayoutManager in which the inline areas flow into.

So a block area manager handles the lines or blocks as its children and determines things like spacing and breaks.

In the case of tables and lists the blocks are stacked in a specific way that needs to be handled by the layout manager.

The block area has info about the following:

- all anchors including which lines they are on
- unresolved page references with line info
- id and link areas
- height (min/max/optimum) or area including floats
- holds space before/after and keep information
- widows and orphans

Once the layout has been finalised then this information can be discarded.

24. Page Areas

Contains inforamtion about all the block areas in the body, before area and footer area.

Has a list of the unresolved page references and a list of id refences that can be used to obtain the area associated with that id.

25. Test Cases

Here a few layout possibilities areas explored to determine how the layout process will handle these situations.

25.1. Simple Pages

All blocks (including nested) are placed on the page with minimum spacing and the last block has the minimum number of lines past the page end. The lowest keep value is then found within the body area limits. Then the next equally low keep is found to determine if the spacing will be closer to the optimum values.

25.2. Before Floats/Footnotes

After filling the page with the block areas then the new body height is used to find the best position to break. Before each line area or block area is remove any associated before floats and footnotes are removed. This will then adjust the available space on the page and may allow for a different breaking point. Areas are removed towards the new breaking point until the areas fit on the page. When finding the optimum spacing the removal of before floats and footnotes must also be onsidered.

25.3. Multicolumn

First the page is filled with all columns for the intial page area. Then each column is adjusted for the new height starting from the first column. The best break for the column is found then the next column is considered, any left over areas a pre-pended to the next column. Once all the columns are finished then all the columns are adjusted to fit in the same height columns. This handles the situation where before floats or footnotes may have been removed.

25.4. Last Page

If in the process of adding areas to a page it is found that there are no more areas in the flow then this page will need to be changed to the last page (if applicable). The areas are then placed on a last page.

26. Status

26.1. To Do

Task	Priority	Notes
Keep Properties	High	Need to get keep-* properties working on block level constructs
Justified Text	High	This has been completed,

		thanks largely to Luca Furini. Although there is still issue 28706 that requires further analysis.
Multi-column layout	High	
Get Markers Working	High	Main Problem is markers can be added to wrong page. LEWP is returning first on Next page!
Absolutely positioned block containers	High	
Background Images	High	
Conditional space suppression	High	
Fix break-* properties	High	
Footnotes	Medium	
Relative positioned block containers	Medium	
Fine-tuning line breaking and hypenation	Medium	
Last page layout	Medium	
Aligned leaders, especially in justified text	Medium	
Floats of all kind	Low	
Border collapsing on tables	Low	RenderX hasnt implemented this (17/03/04)
Fine-tuning all other borders	Low	Not sure what Joerg means by this, border collapse priorties? Dashed and dotted borders have been implemented in PDF
BIDI support	Low	

26.2. Work in Progress

26.3. Completed