FOPOverConstraintDesign

This page is for a general documentation of the design discussion regarding a proper way to resolve over-constrained and otherwise anomalous documents during the layout process. -RhettAultman

Important XSL-FO references on overconstraint:

- XSL-FO Spec 4.3.2: http://www.w3.org/TR/xsl/slice4.html#bpd-slack
- XSL-FO Spec 5.3.4: http://www.w3.org/TR/xsl/slice5.html#overcons_geom

Important FOP references to anomalous documents:

- Bug #8778, the infinite loop that started it all: http://nagoya.apache.org/bugzilla/show_bug.cgi?id=8778
- Thread from fop-dev regarding breaks, overconstraint, and anomalous documents: http://marc.theaimsgroup.com/?t=103734921500001&r=1&w=2

General Definitions

- Overconstraint: situation where explicit constraints on the document result in the production of abnormal areas (such as blocks of a required size that are incompletely-filled, excessive page-breaking, etc). In a situation of overconstraint, some constraint in the document should be relaxed in order to ensure the document processes in a more coherent fashion.
- Anomalous document: An FO document containing constraints that strongly impede functional layout. Many anomalous documents will trigger the infinite loop detector in the current maintenance branch code and will cause an infinite loop in the HEAD code.
- Convergence problems: Constraints force previous layout to be rolled back and redone, leading to oscillations between two or more possible chains of layout decisions. Canonical example: a page master for the last page has (much) smaller body region than the regular page masters, therefore when the last page is detected and rerendered with the last page master, the content doesn't fit and the last page is no longer the last page. This provides for more subtle infinite loops than the situations mentioned above. (Fortunately, FOP can't handle yet any of this conditions anyway)

Goals

- Develop a general strategy for discarding or relaxing problematic constraints at layout-time.
- Attempt to keep as incorporated with the current concepts of HEAD layout system as possible.
- Part of general solution to layout issues

Ideas

- There are two important features that I can see in a system like this- detecting situations where a constraint must be relaxed or ignored and then selecting the correct constraint. To this end, I think there would be some sort of a need to codify the concept of a constraint, which means perhaps that we need a Constraint object that maps back to the constraints in a FO object may be necessary. In this way, a decision-making algorithm of some kind could be added to LayoutManagers so that they could select constraints to ignore. -RhettAultman
- Additionally, I think that the best way to apply such a system to the current layout system is to either have anomaly detection and resolution a proactive part of the layout process or to instead allow layout to happen as normal and invoke anomaly resolution only when problems arise. The former is more complete, but will make FOP slower. The latter is better for performance, but now FOP layout managers have to be able to detect when "trouble" occurs so that the anomaly/overconstraint resolution system can take over. -RhettAultman

Status

Really, not much at this moment...the Wiki page generation is a sort of prelude to serious design work.