BreakPossSignallingFullPage

/!\ This page does not apply to the current design anymore. It was written prior to switching to the Knuth approach. It may still hold some interesting information, though.

BP signalling a full page

The layoutengine testcase test-markers5a.xml is a carefully constructed page, which can hold 4 blocks of BPD 14400 = 57600 total (4 lines). This is the Area Tree of the body of page 1:

```
<regionViewport ipd="360000" bpd="57600" ipda="360000" bpda="57600"
bap="0 0 0 0" is-viewport-area="true" rect="0 36000 360000 57600">
<regionBody ipd="360000" bpd="57600" ipda="360000" bpda="57600"
bap="0 0 0 0" is-reference-area="true">
<mainReference ipd="0" bpd="0" bap="0 0 0 0" is-reference-area="true"
columnGap="0" width="0">
<span ipd="360000" bpd="0" ipda="360000" bap="0 0 0 0"
is-reference-area="true">
<flow ipd="360000" bpd="0" ipda="360000" bap="0 0 0 0"
is-reference-area="true">

<!-- block 1 -->
<block ipd="360000" bpd="14400" ipda="360000" bpda="14400" bap="0 0 0 0">
<lineArea ipd="0" bpd="14400" bpda="14400" bap="0 0 0 0">
<text font-size="12000" font-family="F1" color="#000000">1: 1.00
</text>
</lineArea>
</block>

<!-- block 2 -->
<block ipd="360000" bpd="14400" ipda="360000" bpda="14400" bap="0 0 0 0">
<lineArea ipd="0" bpd="14400" bpda="14400" bap="0 0 0 0">
<text font-size="12000" font-family="F1" color="#000000">2: 1.00
</text>
</lineArea>
</block>

<!-- block 3 -->
<block ipd="360000" bpd="14400" ipda="360000" bpda="14400" bap="0 0 0 0">
<lineArea ipd="0" bpd="14400" bpda="14400" bap="0 0 0 0">
<text font-size="12000" font-family="F1" color="#000000">3: 1.00
</text>
</lineArea>
</block>

<!-- block 4 -->
<block ipd="360000" bpd="14400" ipda="360000" bpda="14400" bap="0 0 0 0">
<lineArea ipd="0" bpd="14400" bpda="14400" bap="0 0 0 0">
<text font-size="12000" font-family="F1" color="#000000">4: 1.00
</text>
</lineArea>
</block>

<!-- block 5start -->
<block ipd="360000" bpd="0" ipda="360000" bap="0 0 0 0"/>
</flow>
</span>
</mainReference>
</regionBody>
</regionViewport>
```

When block 4 ends, the target BPD of 57600 has not been exceeded. Therefore block 5 starts `getNextBreakPoss`. When the first BP is returned, the target BPD is exceeded, the child LM is reset, and a BP is created with the flag `BreakPoss.NEXT_OVERFLOWS` set to true. The BPD of this BP is zero. Therefore this BP's only role is to signal the page break. Nevertheless an area is created for it. This area acts as the first area generated by the block fo, and therefore the corresponding marker is set. A recent patch prevents this by using the flag `isBogus()`.

This bogus BP might also get the `BreakPoss.ISFIRST` flag, which would also be inappropriate.

```
BlockLM.getNextBreakPoss():
loop while ((curLM = getChildLM()) != null)
  loop while (!curLM.isFinished())
    break if over
  reach here if over || curLM.isFinished()
  if (over || getChildLM() == null)
    return BP with childBreaks.size() - 1, which may be -1
  iterate if (getChildLM() != null)
reach here if (getChildLM() == null) and if we never entered the loop,
i.e. empty blocks or call was started at the end of the content (is
that possible?)
return BP with FINISHED_LEAF_POS

BlockLM.addAreas:
return if (lfp.getLeafPos() == FINISHED_LEAF_POS)
option: do the same if (lfp.getLeafPos() == -1)
```

A `BlockLM` may return a BP which does not generate an area in the following cases:

1. The returned BP causes overflow and is reset. This was the first BP of this block. That means that the block was started just before the page break. See FOP's testcase markers5a.xml. The BP has a position of `childBreaks.size() - 1 = -1`.
2. The Block is empty. The BP has a position of `FINISHED_LEAF_POS = -2`. Such a BP does not generate an area.
3. The Block has a page break just before its end; after the page break its childLMs return no further BPs. That is not possible, because a page break is decided by:
    a. a BP that causes overflow of the page; this BP is reset and reappears on the next page;
    b. a BP that has the `NEXT_OVERFLOWS` flag set; that means that the childLM has more content.

Would it be a good idea to just omit the BPs of cases 1 and 2?

This is a nasty case:

```
<fo:block><fo:block/>Several lines of text</fo:block>
```

when the text 'Several...' starts on the new page. There would again be an empty area on the previous page, which would qualify as `is-first`. It is a combination of 2 and 1, but the position of the BP is 0. This can be aggravated by having several empty blocks at the end of the page.

Can an empty FO element have markers? [1] If so, BPs under case 2 should be generated and be allowed to set markers (current situation). Around a page break it would not be defined to which page the marker would belong. The spec does not seem to rule it out. But it speaks about the area to which a marker is associated. A marker in an empty block would not be associated with an area, or is an area of bpd = 0 a valid area? [2]

`BlockLM` has a boolean member `isfirst`, which starts out as true. When a BP is returned which generates a real area and `isfirst == true`, the BP gets the flag `BreakPoss.ISFIRST`, and isfirst is set to false. When the LM `isFinished()`, the BP gets the flag `BreakPoss.ISLAST`. Note that an area can have both the `is-first` and `is-last` traits.

---

[1] Yes, this is valid. fo:markers are permitted on any descendant of an fo:flow. The question is rather if this makes sense. BTW, markers5b.xml is such an example. (JM)

[2] I think this is valid. Reason: 6.5.2 says, that fo:block generates one or more normal block areas. An empty fo:block creates exactly one area with bpd=0. (JM)