

An Exploratory Mapping Strategy for Web-Driven Magazines

Fabio Giannetti
Hewlett-Packard Laboratories
1501 Page Mill Road – M/S 1161
Palo Alto, CA – 94304
+1 650 857 5085
fabio.giannetti@hp.com

ABSTRACT

“There will always (I hope) be print books, but just as the advent of photography changed the role of painting or film changed the role of theater in our culture, electronic publishing is changing the world of print media. To look for a one-to-one transposition to the new medium is to miss the future until it has passed you by.” - Tim O'Reilly [1].

It is not hard to envisage that publishers will leverage subscribers' information, interest groups' shared knowledge and others sources to enhance their publications. While this enhances the value of the publication through more accurate and personalized content, it also brings a new set of challenges to the publisher. Content is now driven by web and in a truly automated system no designer “re-touch” intervention can be envisaged. The paper introduces an exploratory mapping strategy to allocate web driven content in a highly graphical publication like a traditional magazine. Two major aspects of the mapping are covered, which enables different level of flexibility and addresses different content flowing strategies. The last contribution is an evaluation of existing standards, which potentially can leverage this work to incorporate more flexible mapping, and subsequently, composition capabilities.

Categories and Subject Descriptors

I.7 DOCUMENT AND TEXT PROCESSING - I.7.2 Document Preparation: Format and notation and markup. I.7.4 Electronic Publishing: Print publishing for variable data driven templates.

General Terms

Documentation, Design, Standardization, Languages, Theory.

Keywords

Variable Data Print, Template, XSL-FO, SVG, Print, XML, Layout, Transactional Printing, Content Driven Pagination, XPS.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DocEng '08, September 16-19, 2008, São Paulo, Brazil.

Copyright 2008 ACM 978-1-60558-081-4/08/09...\$5.00.

1. INTRODUCTION

In the era of Web pervasiveness, more and more content is generated every day. Web users are posting and retrieving content for their own consumption.

Several web sites are now using automatic feed generation for collection and syndication of news, reviews, etc...

This expanded and expanding amount of information brings several advantages and disadvantages. The typical disadvantages are the complexity in retrieving relevant content as well as verifying the authenticity of its source.

As advantages, users can benefit from aspects like continuous up-to-date content and collaborative environment, where they can submit either new content or comment on existing one.

In this scenario traditional publishing will not be unaffected. Several publishers have already introduced a web version of their publications in order to preserve and convey interest on the topic and cross reference among the different media. This is just a small step towards cross media integration, where professional content intended for the publication and web user content are mixed and produced in printed media.

Just to give a concrete example, we would like to introduce the “travel guide”. Up to now, travel guides [2] [3] are composed by journalists that travel and report their experiences. The most successful travel guides integrate text and pictures, 3D itineraries, maps and so on.

In a web-driven publication this content could be integrated by sections linked to other people's experiences and ratings. People that already visited those places experienced those tours and can provide not only textual feedback, but also rate the overall experience. Moreover, experienced or insightful travelers can provide alternative locations, restaurants or shopping tips that the guide journalist did not know or found. Clearly, all this content needs to be validated and placed in perspective, a role that definitely lies on the publisher's shoulders.

In this paper we introduce a mechanism to map a mix of professional content and web driven content to a professional designed layout. In a magazine publication, for instance, the articles can be gathered from feeds, and joined together with images. This approach opens the possibility for personalized magazines, where among the large selection of content only a targeted sub-set is selected to match the recipient interests. In order to produce these

publications a print production workflow must receive an unbounded sequence of content, comprised of text and images, and through a Variable Data Print (VDP) Workflow compose the document and generate a Page Description Language [4]. Figure 1 illustrates high level end-to-end pre-RIP (Raster Image Processing) steps of a Web-Driven Magazine publication.

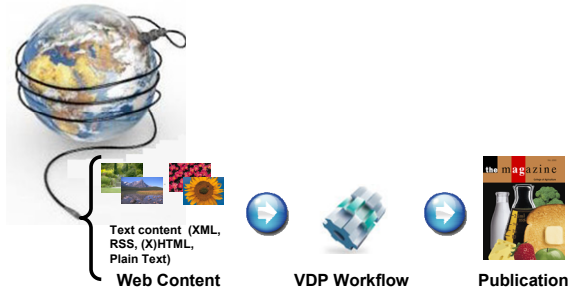


Figure 1: Web-Driven Magazines Scenario

State of the art VDP workflows emerged from different business needs. The historical VDP workflows are driven by the transactional business of bank statements, bills and so on. System to address this need were introduced as early as the late sixties and mainly driven by IBM using their Advanced Function Presentation (AFP) [5] language. More recently the direct mail business introduced new VDP workflow needs that build on traditional graphic artist (GA) oriented layout. These new products are targeted to add variability to static layouts and may provide different levels of complexity. Some only provides copy-holes substitution others move deep into in-page flexibility. We believe that publications like web-driven magazines require a different mix of the VDP capabilities available in these tools, something that marries the transactional aspect of content driven pages and the layout features of a direct mailing. Next section compares the typical layout models found in transactional and direct mail templates.

2. LAYOUT-DRIVEN VS. CONTENT-DRIVEN DESIGN

Magazines require a set of specific composition capabilities, which are more part of the “high-touch” document family than the book, letters or transactional set. Nevertheless, when the content is driven by web feeds, content syndication or even mash-ups, magazines are seen more as a transactional document than a traditional GA’s work.

In order to successfully publish this new breed of documents it is necessary to maintain a concept of unbounded number of pages and a highly graphical design. In the traditional magazine creation business, GA prepares a set of spreads. A spread is a set of at least two pages that are designed to host an article maintaining a visually pleasant and stimulating design. The content is then manually placed inside its corresponding “frames”, usually

done for images, or for flows between a pre-defined set. The result of this operation is the final composed spread (as illustrated in Figure 2) [6]. This design environment is commonly found in professional print design tools, like Adobe InDesign [7] and Quark xPress [8].

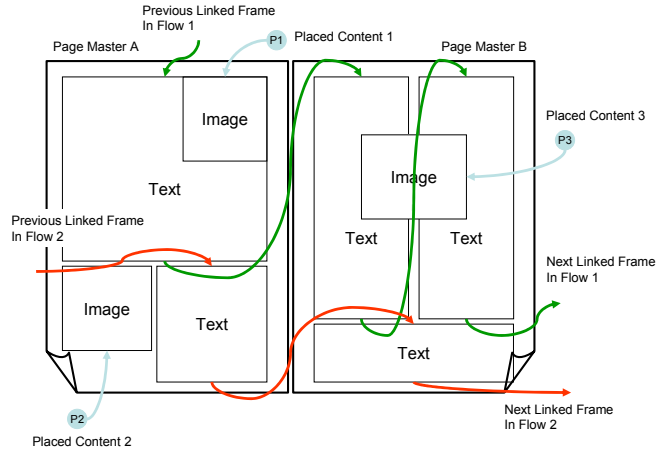


Figure 2: Layout Driven Design Schematization

Several spreads are joined together to produce the final magazine.

In a content driven scenario the layout is usually comprised of a main area, usually referred as body, and the content flows throughout pages until it is exhausted. The layout is trivial and, since the content is dynamic, many of the more complicated formatting capabilities are embedded in the content flow itself (e.g., anchored images). Figure 3 illustrates a typical example of content driven layout (e.g., Microsoft Word page template).

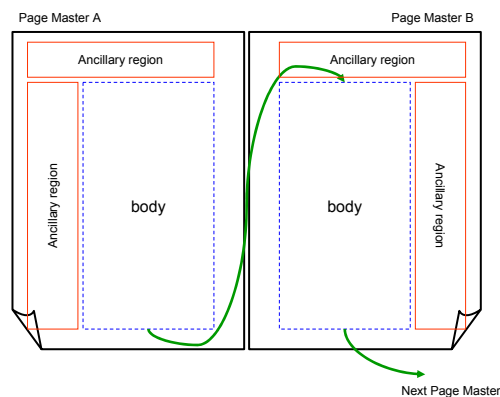


Figure 3: Content Driven Layout Schematization

The main difference between the approaches introduced in Figure 2 and Figure 3 are:

- in layout driven the designer expresses a finite number of connections between the different frames, fixing the overall number of pages in the publication

- in the content driven the designer only links the content with the “body” region and the publication pages are automatically generated.

In a Web-Driven Magazine application, it is required to have the best of both worlds [9]. In fact, magazine readers expect a highly graphical publication meanwhile, the unpredictability of the content requires a content driven strategy to create an unbounded number of pages.

3. OBJECTIVES

In this work we propose a strategy to map variable content to highly graphical design. This strategy enables a degree of separation between formatting content and layout. The main objective is to allow designers to create complex spread layouts and at the same time unify the content linking in a more content driven approach. Figure 4 illustrates the addition of the content-to-layout mapper to a generic VDP workflow.

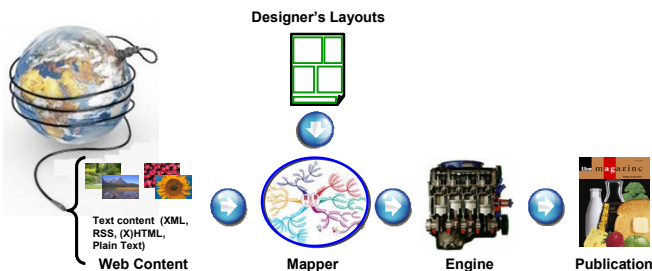


Figure 4: Introduction of Content-to-Layout Mapper in a VDP Workflow

The content-to-layout mapper abstracts the unknown content length and type, e.g. text or graphics, with the fixed frames available from the design stage. Even with the introduction of a mapper there are still different situations to be handled. In this work we have identified the following:

1. the designer names frames and links them as in a static layout approach, s/he also enables an automatic flow among a set or sub-set of repeating frames in different pages, we would refer to this approach as Repeating Frame Mapper (RFM);
2. the designer identifies “body” like frames for one or multiple concurrent flows, s/he also defines extra optional regions that contains specific potential out-of-flow elements, we would refer to this approach as With Purpose Frame Mapper (WPFM).

Clearly these situations enable different ways for fulfilling the layout with the variable data. We believe that there may be value in both. Each situation and proposed solution is described in the following sections.

4. REPEATING FRAME MAPPER (RFM)

Publications as magazines are usually formed by a set of articles. These articles are either presented within a spread or can span among several spreads. If the article content is of variable length, then a designer does not have the faculty of linking its content to a fixed number of spreads. On the other hand, just repeating the same spread or page several times creates an uninteresting and unpleasant publication. The designer needs to provide several page layouts that can accommodate an article. S/he can decide the order of these pages but does not know the final number. In this case there is a need to express the linkage between the content and the set of interconnected frames that may or may not present in the same page at the same time. This ensures a certain degree of freedom for the designer to re-position, change dimension and integrate a flowing article with other content making. That results are that the overall publication is nicely laid out and interesting for the reader. Figure 5 illustrates a schematic representation of a RFM approach.

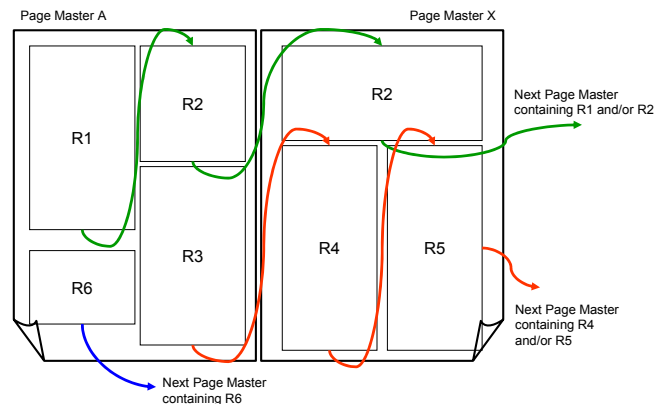


Figure 5: Schematic RFM representation

It should be taken into account that content allocated to frames R1 and R2 can flow in any Page Master’s definition containing one or more of these frames. The mapper expresses the possibility of either forcing the presence of all frames, or enabling the flexibility for some or all to be optional.

Before the mapper structure is introduced in this paper, it would be beneficial to use a pseudo Document Description Language (DDL) based on eXtensible Markup Language (XML) [10]. Figure 6 illustrates a self-explanatory notation for the page masters definitions introduced in Figure 5.

```

<page-description id="A" width="..." height="...">
  <layout>
    <frame id="R1" x="..." y="..." width="..." height="..."/>
    <frame id="R2" x="..." y="..." width="..." height="..."/>
    <frame id="R3" x="..." y="..." width="..." height="..."/>
    <frame id="R6" x="..." y="..." width="..." height="..."/>
  </layout>
</page-description>
<page-description id="X" width="..." height="...">
  <layout>
    <frame id="R2" x="..." y="..." width="..." height="..."/>
    <frame id="R4" x="..." y="..." width="..." height="..."/>
    <frame id="R5" x="..." y="..." width="..." height="..."/>
  </layout>
<page-description id="Y" width="..." height="...">
  <layout>
    <frame id="R2" x="..." y="..." width="..." height="..."/>
    <frame id="R3" x="..." y="..." width="..." height="..."/>
    <frame id="R4" x="..." y="..." width="..." height="..."/>
    <frame id="R5" x="..." y="..." width="..." height="..."/>
  </layout>
</page-description>
<page-description id="Z" width="..." height="...">
  <layout>
    <frame id="R2" x="..." y="..." width="..." height="..."/>
    <frame id="R3" x="..." y="..." width="..." height="..."/>
    <frame id="R4" x="..." y="..." width="..." height="..."/>
    <frame id="R5" x="..." y="..." width="..." height="..."/>
    <frame id="R6" x="..." y="..." width="..." height="..."/>
  </layout>
</page-description>

```

Figure 6: Page Master Layout Definition

Associated with the page masters definition there is a need of defining the sequence of these page masters. This could be done using a separate structure or simply adding this information in the page master definition itself. For the sake of simplicity we express a pagination model using the latter. Figure 7 illustrates a simple repetition strategy for an even-odd with a separate first and last page. This pagination model represents the simplest for a magazine like publication.

```

<page-description id="A" width="..." height="..." repeat="first">
  ...
</page-description>
<page-description id="X" width="..." height="..." repeat="even">
  ...
</page-description>
<page-description id="Y" width="..." height="..." repeat="odd">
  ...
</page-description>
<page-description id="Z" width="..." height="..." repeat="last">
  ...
</page-description>

```

Figure 7: Pagination model with a simple sequence

The mapper, as described in Figure 8, is joining the frames described in Figure 6 with a hypothetical content flow.

The notation represented in the above illustration simply provides a mechanism to reference some flow content with a list of frames. The order in which the flow content and the frames are listed represents the order in which the content will appear and the frames are linked.

```

<mappers id="rfm-mappers">
  <mapper id="green">
    <content-flow-sources>
      <content-flow-source ref="xpointer(...)" />
      ...
    </content-flow-sources>
    <frame-targets>
      <frame-target id="R1" conditional-instances="optional" />
      <frame-target id="R2" conditional-instances="optional" />
    </frame-targets>
  </mapper>
  <mapper id="red">
    <content-flow-sources>
      <content-flow-source ref="xpointer(...)" />
      ...
    </content-flow-sources>
    <frame-targets>
      <frame-target id="R3" conditional-instances="1" />
      <frame-target id="R4" conditional-instances="always" />
      <frame-target id="R5" conditional-instances="optional" />
    </frame-targets>
  </mapper>
  <mapper id="blue">
    <content-flow-sources>
      <content-flow-source ref="xpointer(...)" />
      ...
    </content-flow-sources>
    <frame-targets>
      <frame-target id="R6" conditional-instances="always" />
    </frame-targets>
  </mapper>
</mappers>

```

Figure 8: Repeating Frame Mapper description

The novel part is related to the `conditional-instances` attribute. This attribute has been introduced to allow the repeatability mechanism, but also to provide flexibility. The available attribute values are:

- `always`, this means that any given page must contain the referred frame;
- `optional`, this means that any given page can or cannot contain the referred frame;
- `<<number>>`, this means that there must be exactly `n` pages in the sequence containing the referred frame;
- `<<number>>+optional`, this means that there must be at least `n` pages in the sequence containing the referred frame;
- `<<number>>+none`, this means that there must be no more than `n` pages in the sequence containing the referred frame.

In order to visualize the example presented in Figure 6, it is worth to analyze what happens to the “blue” mapper. In this case, all the content is forced to flow between pages that have an R6 frame. Considering the pagination presented in Figure 7: the flow for the “blue” mapper will be rendered in the first page (page master A) and in the last page (page master Z) with an arbitrary number of even and odd pages in the middle driven by the other concurrent flows (“green” and “red” mappers). This behavior emulates the typical news article that starts in the first page and continues elsewhere. Figure 9 illustrates a representation of the final pagination result highlighting the “blue” flow.

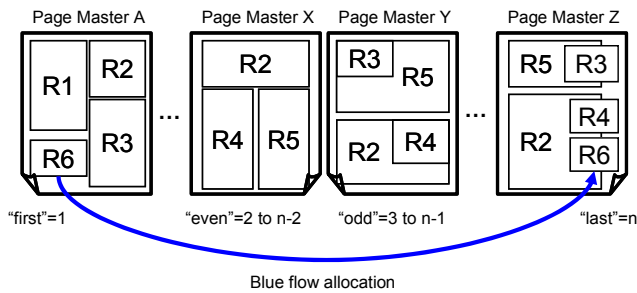


Figure 9: Pagination result with “blue” flow.

5. WITH PURPOSE FRAME MAPPER

This second type of mapper introduces a further level of complexity. In the RFM flexibility has been introduced allowing conditional usage of regions to achieve the content flow. Meanwhile this is useful where the flow-able content can be presented within the boundaries of a frame or set of frames, it is not possible to automate the flow in case of more dynamic layout. A simple case is related to a two columns article that has an image positioned in the middle creating a run-around. This layout works perfectly fine if the article is referring to the image within the same page. Since the content length is unknown, the image reference can appear anywhere in the paginated result, leaving possible that the two column layout may have or not an image reference within the boundaries of the page. If the image is not referenced, the layout is clearly odd, since there will be an empty region creating a runaround. Ideally it would be necessary to have different versions of page masters, one in order to accommodate the image and one without. Figure 10 illustrates the two cases.

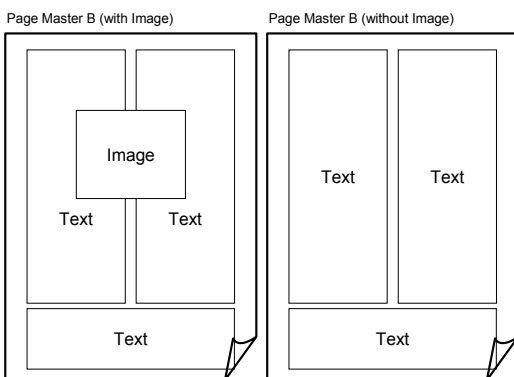


Figure 10: Page Master with or without the image frame

Synchronizing the flow with the appropriate page master during composition is a complex task; moreover the designer needs to provide and identify all the possible alternatives before hand.

The proposed solution avoids this complexity allowing certain type of content to be redirected automatically to some regions when it is available, as illustrated in Figure 11. This content is usually referred as “out-of-flow” since it is generated within the flow, but it pertains to a different area or

regions [11]. Within this category are: footnotes, quotes, floats and other content elements that are separated from the main flow for reading clarity purposes or simply aesthetics. Another aspect of the proposed solution is that out-of-flow related frames only exist if there is appropriate content and only receive the selected type of content, for this reason they are “with purpose”. A composition engine can now identify the existence of these elements in the flow using either a looking ahead strategy or parallelizing the composition task of different alternatives.

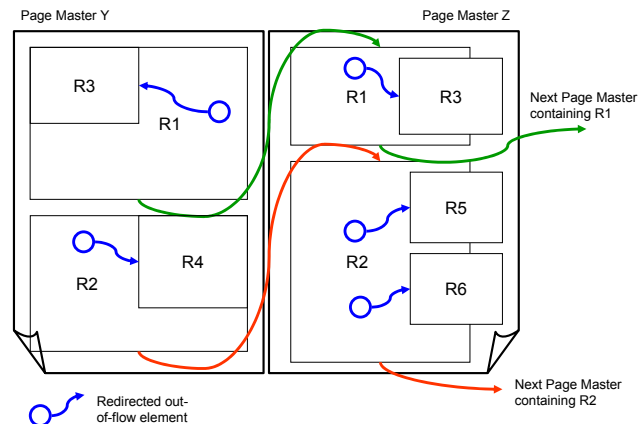


Figure 11: Out-of-Flow content redirection in relative frames.

Referring to the previously defined page masters (Figure 6) and pagination models (Figure 7), it is possible now to introduce the new definition for a WPFM.

```

<mappers id="wpfm-mappers">
  <mapper id="green">
    <content-flow-sources>
      <content-flow-source ref="xpointer(...)" />
      ...
    </content-flow-sources>
    <frame-targets>
      <frame-target id="R1" conditional-instances="always" />
    </frame-targets>
    <out-of-flow-destinations>
      <out-of-flow-destination destination-id="R3" type="float" />
    </out-of-flow-destinations>
  </mapper>
  <mapper id="red">
    <content-flow-sources>
      <content-flow-source ref="xpointer(...)" />
      ...
    </content-flow-sources>
    <frame-targets>
      <frame-target id="R2" conditional-instances="always" />
    </frame-targets>
    <out-of-flow-destinations>
      <out-of-flow-destination destination-id="R4" type="float" />
      <out-of-flow-destination destination-id="R5" type="quote" />
      <out-of-flow-destination destination-id="R6" type="float" />
    </out-of-flow-destinations>
  </mapper>
</mappers>

```

Figure 12: With Purpose Frame Mapping description.

For a visual example that exemplifies the achieved flexibility it is possible to imagine a simple page sequence with Page Master Y and Page Master Z to be on even and odd sequence. Depending on what flow content is presented on each page at composition time the resulting pages can be as illustrated in Figure 13.

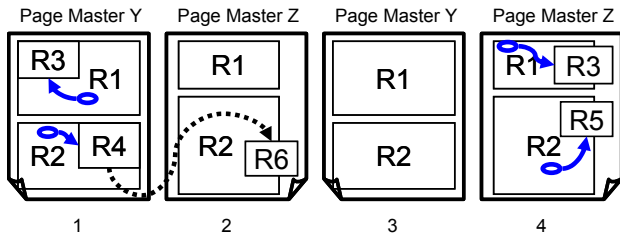


Figure 13: Composition result based on hypothetical out-of-flow content.

The optional out-of-flow related frames enable the designer to present a unified design that encompasses all the possible alternatives. This approach also leaves to the composition engine to automatically adjust the page layout to accommodate the eventual presence of out-of-flow content. A further interesting effect of the proposed solution is that whenever the out-of-flow content does not fit in a frame it may flow among frames that share the same purpose as indicated by the dotted line in Figure 13. This is achieved having multiple **out-of-flow-destination** sharing the same **type**, in this specific case: **float** (see Figure 12).

6. APPLICABILITY TO EXISTING STANDARDS

The proposed technique could be applied to any standard XML language that supports pagination and flows. Currently only the eXtensible Stylesheet Language – Formatting Objects (XSL-FO) [12] has a full support for these aspects, but the Scalable Vector Graphics (SVG) – Print [13] subset and SVG 1.2 [14] text flow capabilities could provide a base. Moreover, we believe that relatively new formats like Microsoft’s XML Paper Specification (XPS) [15] could provide such an environment too.

To exemplify the applicability of the WPRM, illustrated in Figure 12, we are creating a notation compatible with the existing XSL-FO mapper (see Figure 14).

```
<fo:flow-map flow-map-name="aQName">
  <fo:flow-assignment>
    <fo:flow-source-list source-list-name="green">
      <fo:flow-name-specifier flow-name-reference="GreenCnt"/>
    </fo:flow-source-list>
    <fo:flow-target-list>
      <fo:region-target-specifier region-name-reference="R1"
        conditional-instances="always"/>
    </fo:flow-target-list>
  </fo:flow-assignment>
  <fo:out-of-flow-assignment out-of-flow-objects="float">
    <fo:flow-source-specifier source-list-reference="green"/>
    <fo:flow-target-list>
      <fo:region-target-specifier region-name-reference="R3"
        conditional-instances="optional"/>
    </fo:flow-target-list>
  </fo:out-of-flow-assignment>
  <fo:flow-assignment>
    <fo:flow-source-list source-list-name="red">
      <fo:flow-name-specifier flow-name-reference="RedCnt"/>
    </fo:flow-source-list>
    <fo:flow-target-list>
      <fo:region-target-specifier region-name-reference="R2"
        conditional-instances="always"/>
    </fo:flow-target-list>
  </fo:flow-assignment>
  <fo:out-of-flow-assignment out-of-flow-objects="float">
    <fo:flow-source-specifier source-list-reference="red"/>
    <fo:flow-target-list>
      <fo:region-target-specifier region-name-reference="R4"
        conditional-instances="optional"/>
      <fo:region-target-specifier region-name-reference="R6"
        conditional-instances="optional"/>
    </fo:flow-target-list>
  </fo:out-of-flow-assignment>
  <fo:out-of-flow-assignment out-of-flow-objects="quote">
    <fo:flow-source-specifier source-list-reference="red"/>
    <fo:flow-target-list>
      <fo:region-target-specifier region-name-reference="R5"
        conditional-instances="optional"/>
    </fo:flow-target-list>
  </fo:out-of-flow-assignment>
</fo:flow-map>
```

Figure 14: XSL-FO 2.0 proposed version for the mapper.

The XSL-FO mapper is less compact compared to the XML notation introduced earlier. This is due for two main aspects:

- XSL-FO mapper explicitly states the content and all the regions;
- to leverage the existing XSL-FO notation it is necessary to provide separate **fo:flow-assignment** and, introduce, **fo:out-of-flow-assignment** for each individual flow and **out-of-flow-objects** object types.

7. CONCLUSIONS AND FUTURE WORK

In this work a mechanism to provide automatic re-flowing capabilities with GA quality designs has been proposed. In order to support these new paradigms, in existing composition engines, the pagination generation model needs to be modified. It is our belief that it would be a simpler task if these new concepts are applied to “transactional” composition engines rather than

“traditional” GA engines. Moreover, when the flowed content does not have any inter/intra-page copy-fitting requirements, the composition can be achieved in a single pass. Validating these assumptions and extending a composition engine is part of the future work, as well as explore potential standardization paths.

An experimental XSL-FO rendering engine is already implemented as part of the Anvil-NG [16] project.

The Anvil-NG project already implements an extended capability for regions and implements the flow-mapper as specified for XSL-FO 1.1.

An interesting further research topic is to determine how the data merging step can incorporate some of the mapper capabilities. In a simple XML based VDP workflow [17], as illustrated in Figure 15, the data merging is executed through the use of the eXtensible Stylesheet Language – Transform (XSL-T) [18].

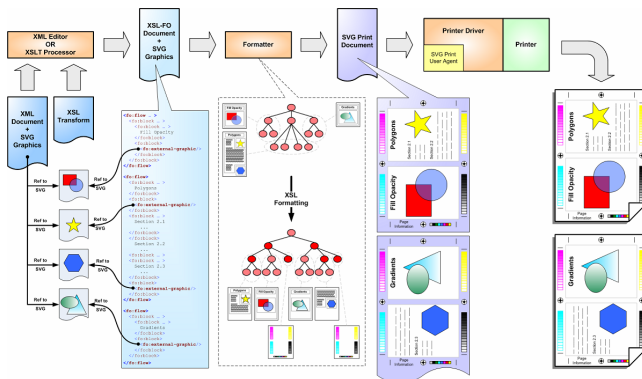


Figure 15: XML based Print Workflow

The XSL-T processor takes as input the XML data and applies a style-sheet. The style-sheet is designed to convert the data into a Document Description Language, for instance XSL-FO. During this phase the mapper is created. The XML data is represented as flows and the mapper is created to link the data with the appropriate regions. An interesting development is to examine the data during the XSL-T processing in order to investigate the presence of “out-of-flow” elements. This could steer the mapping generation to take into account these elements or not. In case some flows are lacking of these elements, the mapper can be simplified. As result, the composition engine can reduce the time required to generate the PDL. The absence of the “out-of-flow” elements eliminates the need of optional regions from the page layout converting it into a conventional case, already supported in XSL-FO 1.1.

8. REFERENCES

- [1] Tim O’Reilly, *Repeated Misconceptions About eBooks*, <http://www.oreillynet.com/pub/wlg/1664>, Jul. 06, 2002.
- [2] DK Publications, <http://us.dk.com/static/cs/us/11/travel/intro.html>
- [3] Lonely Planet, <http://www.lonelyplanet.com>
- [4] Dewitz, A., *Web-Enabled Print Architectures*, School of Print Media – RIT, Rochester, PICRM-2008-06
- [5] IBM, *Advanced Function Presentation (AFP)*, <http://www.outputlinks.com/SpecialInterest/AFPColorConso rtium/home.html>
- [6] C. Jacobs, W.Li, E. Schrier, D. Bargerion, D. Salesin, *Adaptive grid-based document layout*, ACM Trans. Graphics, Vol. 22, No. 3, 838-847, 2003
- [7] Adobe InDesign, <http://www.adobe.com/products/indesign/>
- [8] Quark xPress, <http://www.quark.com/products/xpress/>
- [9] Kingston, J. H., *The Future of Document Formatting*, Whitepaper, <http://www.it.usyd.edu.au/~jeff/nonpareil/>, 2006
- [10] W3C, *eXtensible Markup Language (XML) 1.1 (Second Edition) W3C Recommendation*, 2006, <http://www.w3.org/TR/2006/REC-xml11-20060816/>
- [11] Kahl, W., *Beyond Pretty-Printing: Galley Concepts in Document Formatting Combinators*, PADL’99, LNCS 1551, pp. 76-90, 1998.
- [12] W3C, *eXtensible Stylesheet Language – Formatting Objects (XSL-FO) 1.1 W3C Recommendation*, 2006, <http://www.w3.org/TR/xsl/>
- [13] W3C, *Scalable Vector Graphics (SVG)- Print 1.2 W3C Working Draft – Part 2: Language*, December 21 2007, <http://www.w3.org/TR/SVGPrint/>
- [14] W3C, *Scalable Vector Graphics (SVG)- Full 1.2 Specifications - W3C Working Draft*, April 13 2005, <http://www.w3.org/TR/SVG12/>
- [15] Microsoft, *XML Paper Specification (XPS)*, <http://www.microsoft.com/whdc/xps/default.mspx>
- [16] Giannetti, F., *Anvil next generation: a multi-format variable data print template based on PPML-T*, ACM Symposium on Document Engineering 2007: 93-94
- [17] W3C, *Scalable Vector Graphics (SVG)- Print 1.2 W3C Working Draft – Part 1: Primer*, December 21 2007, <http://www.w3.org/TR/2007/WD-SVGPrintPrimer12-20071221/>
- [18] W3C, *eXtensible Stylesheet Language – Transformations (XSL-T) 1.0 W3C Recommendation*, 1999, <http://www.w3.org/TR/xslt>