Advanced Layouts in a Content-Driven Template-Based Layout System

ISTVÁN ALBERT, HASSAN CHARAF, LÁSZLÓ LENGYEL
Department of Automation and Applied Informatics
Budapest University of Technology and Economics
HUNGARY
{ialbert, hassan, lengyel}@aut.bme.hu

Abstract: - The current capabilities of mobile device screens facilitate that online reading experience rival with the printed media. Commercially printed magazines and newspapers commonly apply different grid-based page designs. In case of the online magazines the variable conditions, e.g. screen resolution, actual content and user settings require to provide adaptive online document formatting solutions. In this work, we present a new approach to adaptive content-driven template-based document layout that is about to address the current shortcomings. In our approach, an adaptive layout is encoded as a set of column templates that know how to adapt to different device capabilities and viewing conditions, i.e. screen sizes and resolutions. Templates include various types of layout elements, such as text, image, or caption areas, and with the help of layout rules, we define how these areas should be laid out. Besides the layout definitions, the solution takes into account the conditions (screen properties and user settings, e.g. the font size), and the actual content that also affects the final layout. In this paper we discuss our solution that targets tablet devices with online magazine adaptive layouts. We also provide several challenging advanced layouts, which inspired the capabilities of our approach.

Key-Words: - Adaptive Layout, Content-Driven Layout, Template-Based Layout, Online Magazine Layout.

1 Introduction

When viewing any document online, often there is the choice between the carefully formatted version, and the version that flows more gracefully onto the screen on which it is viewed. Documents belonging to the first case have a fixed layout, like Adobe's Portable Document Format (PDF), i.e. they are formatted only for one specific page size, e.g. letter or A4. This inflexibility leads to a poor online reading experience, since the size and resolution of monitors and mobile devices require readers to scroll around in order to read a page. In the second case all of the design information, e.g. grid-based arrangement, is lost [1].

High-quality automatic document formatting is an extremely difficult problem [2]. The diversity of mobile platforms and the mobile device capabilities, which are the main consumers of the online data, requires providing automatic layout solutions for online content. Readers would like to utilize the special capabilities of their own devices.

In this paper, we propose an approach to adaptive column-based document layouts. Layouts incorporate templates, display and adaptation rules, and style attributes as well. Templates are assembled from rectangular areas, such as text, image and caption. Column-based means that a template defines the arrangement of one or more columns, but not a part of a column, i.e. the basic building block that a template can define is a column. Each adaptive layout definition is designed to adapt to a range of display dimensions, as well as to other viewing conditions, such as the font size. The document content is formatted dynamically to fit the actual viewing conditions, i.e. the device being used, and as well as other user preferences.

In order to make this approach practically applicable we faced a variety of research challenges. (i) We need a domain-specific language that is flexible enough to represent the template areas, layout definitions, the related adaptation rules, and the styling information. (ii) We need a layout authoring tool that helps the work of magazine editors to define the templates and assemble the layouts. The tool should provide an easy to understand and use interface that supports the template authoring and also the testing of the prepared layouts. (iii) We need a layout engine that based on the layout definitions, the device properties, the user settings, and also taking into account the actual content renders the final layout.

There are different technologies and approaches targeting the online document reading and adapting the actual content to different conditions. HyperText

Markup Language (HTML) [3] is the main and most popular markup language for displaying web pages and other information that can be displayed in a web browser. HTML documents have a *what you see is all you get* nature. Web page layout is quite important to provide better look to websites. There are many websites, which have put their content in multiple columns thus they are formatted like a magazine or newspaper. This is easily achieved by using tables or division or span tags. Often we apply Cascading Style Sheets (CSS) [4] [5] [6] as well to position various elements or to create backgrounds or colorful look for the pages. Windows Presentation Foundation (WPF) [7] [8] provides developers with a unified programming model for building Windows smart client user experiences that incorporate UI, media, and documents. The core of WPF is a resolution-independent and vector-based rendering engine.

The rest of this paper is organized as follows. Section 2 discusses the main features of our adaptive layout approach. In Section 3, we introduce layouts as advanced challenging requirements related to the approach. Finally, conclusions and the actual work are elaborated.

2 A Content-Driven Template-Based Layout System

The principal challenge behind adaptive document layout for dynamically aggregated content lies in the design of layout templates that are flexible to many different display capabilities and user settings, such as text size. Editors must be able to incorporate layout preferences and conditions with minimal effort. The layout engine must be able to make reasonable choices about content display without intervention.

The significant areas and features of our online document layout approach incorporates the followings: layout template definition, managing images to keep the appropriate aspect ratio, handling empty spaces, providing multiple content streams, working with margins and paddings, defining background properties, and the nature of non-rectangular areas.

We maintain a clean separation between document content and presentation style in our solution. The document content is represented as a set of individual streams, each of which contains content that is laid out sequentially. Streams are differentiated by media type and also by purpose. A magazine article may have several streams. In our system, a document is represented as a set of parallel streams of content. For example, the body text of an article could be one of the streams in the document. Other streams might contain images or headlines. We use a classic semi-structured, hierarchical data representation to describe the full content of an article.

We allow that content elements be annotated with custom attributes. For example, images in a stream are ordered. The purpose of prioritizing them is to always present the most important ones. This is because based on the actual conditions and the allowed adaptation methods the layout engine can modify the number of presented images. Of course, not presented images are available in the gallery. Furthermore, to be able to keep the aspect ratio of the images we use ideal cuts with focus points on the original image. When a different aspect ratio is required by the layout engine these cuts are extended horizontally or vertically according to the focus point of the original image. The designer defines several prioritized wide and also tall rectangular areas that are attached to the ideal cuts. When the layout engine requires a different aspect ratio the cut is based on one of these rectangular areas and the focus point. The layout engine based on these properties can resize and cut images while keeping their aspect ratio and always showing their relevant part and if the available space makes possible then more.

The layout is defined with the help of the templates. We use *templates* to define one or more columns, i.e. a template covers the whole column from the top till the bottom. A *layout* is assembled from one or more templates. The paper introduces typical layouts that editors are about to realize. *Rules* (constraints) are related to both templates and layouts.

The basic building blocks of the templates are *rectangular areas* that are arranged in the template and filled with content. Each area receives content from one of the document streams. More than one element may consume content from the same stream, in this case the content flows from one element into the next. Figure 1 introduces a sample layout. This layout contains a title text area, a highlighted text area, an image and four text areas that presents the body text. We use grey background to indicate that a text area is a highlighted text area. This means that the length of the text in this area is fixed, i.e. the text does not flow into this area and flow out from it. The arrows show the flow of the body text among text areas.

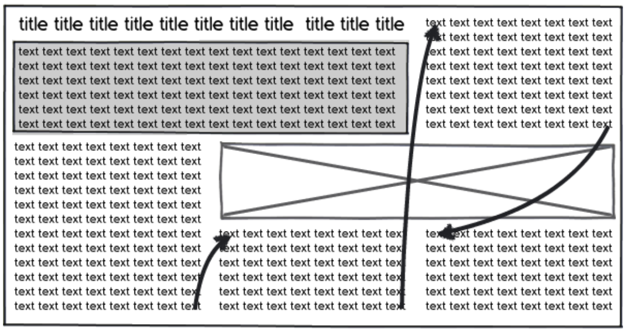

Figure 1. Sample basic layout.

Templates can be hierarchical. Rectangular areas that refer to templates can specify a set of templates to format the hierarchical collection of content displayed by that area.

A template may have substitute templates. They are ordered, and if the original template cannot be applied for the actual content or screen size/resolution then the layout engine uses the next substitute template. For each template it tries to apply the adaptation methods (e.g. image resizing, line space modification, column width modification, insert/remove columns, etc.) to render an appropriate layout. If the template cannot be adapted to the actual conditions then the algorithm moves to the next template. This means that when rendering the final layout, the layout engine chooses the template that best fits the particular content.

We apply conditional columns, which are automatically chosen whether to appear in a given layout based upon the availability of content and any other arbitrary rules. Conditional columns result that less template definition is necessary to properly display the same content on different devices. The layout engine generates as many columns as it is required by the length of the text and the conditions effected by the user settings.

One important property of our system is the separation of content and style, which allows designs to accommodate unknown content.

The layout engine based on the layout definitions, the device properties, the user preferences, and also based on the actual content renders the final layout. Comparing to other solutions, we emphasize that our approach takes into account the actual content and the final layout is partly driven by it.

3 Advanced Layouts

Based on the layouts of several popular printed magazines we have assembled a list of challenging layout requirements. The list contains 51 different layouts. These layouts fall in three categories: (i) Basic layouts, (ii) Advanced layouts, and (iii) Full column layouts. In this section we introduce the advanced layout snippets and discuss their properties. Basic and full column layouts are the subject of different publications.

Basic layout patterns define the essential layouts. An example is provided in Figure 1. Advanced layout patterns cover more complex arrangements of the rectangular areas and rules related to them. In case of full column templates all the available space is fully covered, i.e. the columns end at the bottom of the display. This is true for the last column as well. In order to achieve this goal the layout engine applies different adaptation methods, resizes the areas, insert or eliminate columns according to the template rules.

The advanced layouts presented in this section are categorized as follows (the numbering starts from 33, because the numbers 1–32 are allocated for basic layout patterns):

- Layouts 33–36 contain templates that have alignment related influence on each other.
- Layouts 37–41 apply repetitive elements.
- Layout 42 contains rotated areas.

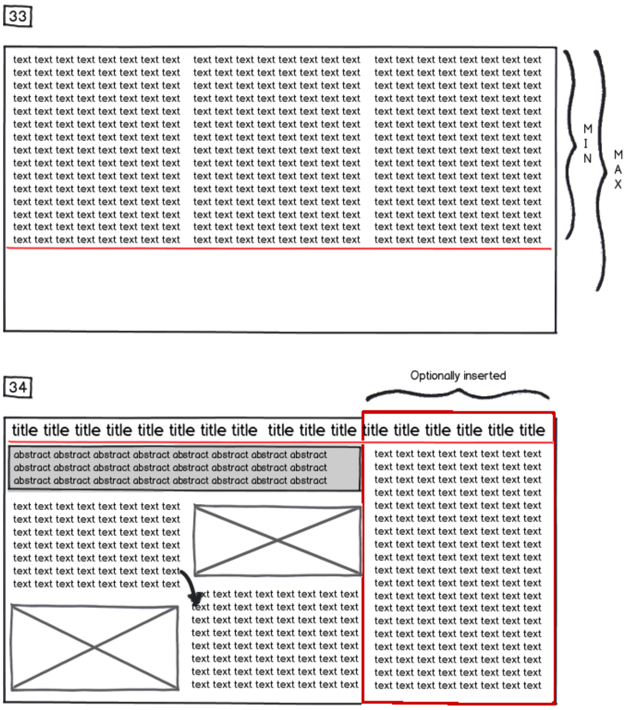

Figure 2. Advanced layouts 33 and 34.

In Layout 33 (Figure 2) the bottom of the text areas are on the same base line (red line in the Figure). The actual height of the columns can be varied between the minimum and the maximum allowed heights. Text areas can contain fixed texts from different streams, or they can have the same stream, which results that the text flows from one area into the other.

Layout 34 (Figure 2) presents and article with a title, an abstract, figures, and with text areas. Based on the length of the text and the actual text size the template can be rendered in two columns or in three columns. If there are three columns, then the top of the third column is aligned to the top of the previous column. The layout engine shows or hides the third column based on the actual conditions. The algorithm decides when to hide one of the figures to provide better layout.

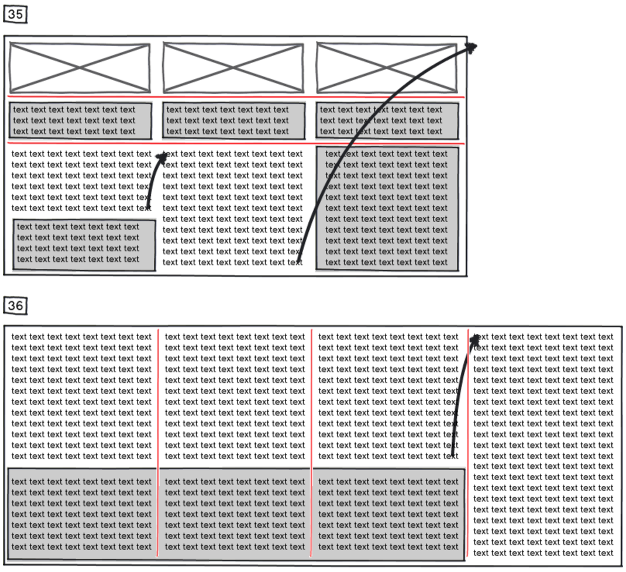

Figure 3. Advanced layouts 35 and 36.

Layout 35 (Figure 3) shows an example where the rectangular areas in the columns next to each other follow the same base lines (red lines in the Figure). The layout contains different templates, i.e. different columns are realized with different templates. The example highlights that the rules, in this case the base line rules, can have effect across several templates.

Layout 36 (Figure 3) contains text columns and a highlighted text area, which also contains text columns that are aligned to the columns above them. The highlighted text area has a different content stream, and the number of the columns is automatically calculated. All of the columns within the highlighted text area will fill the full length. To achieve it the height of the highlighted text area is set based on the length of the text and the actual text size.

Layout 37 (Figure 4) is used to present a conversation, a dialogue, or ideas of different people related to the same topic. On the top there is the title and two highlighted text areas. The rest of the template contains images on both sides and the elements of the conversation or the ideas of the different people. The template can be repeated without the header part (title and highlighted text areas) optional time.

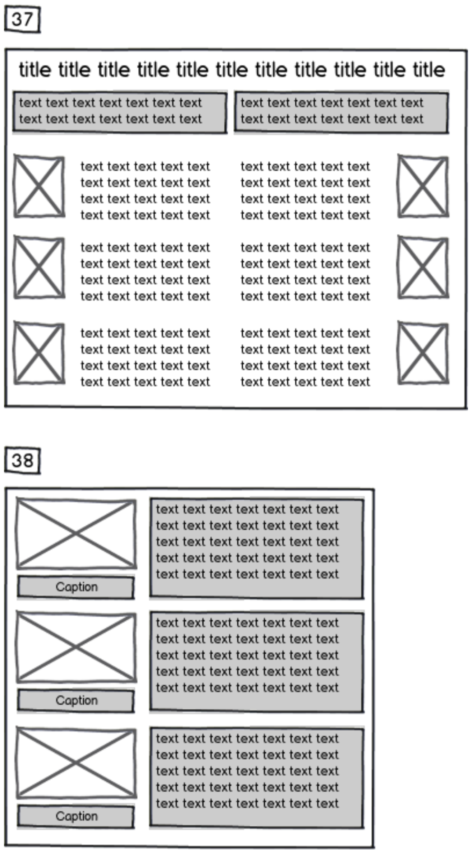
37
title title title title title title title title title title title
38
Caption
Caption
Caption

Figure 4. Advanced layouts 37 and 38.

In Layout 38 (Figure 4) there are two columns. The first column contains images and captions, while the second column contains highlighted text areas. The columns may have different background, e.g. different color or pattern. The template is repeated while the content requires it. The images and the text content of the highlighted text areas are linked. Therefore, the appropriate image is placed next to the right text.

Layout 39 (Figure 5) contains three columns. On the left and on the right sides there are text columns and in the middle column there are image areas. The number of the images depends on the actual content, i.e. on the image stream. The arising questions are the followings: How the layout engine should handle the images? Should it keep the column with or should it change to be able to show all of the images? Should it enable vertical scrolling only in the image column? Or should it throw less important images to achieve quality layout?

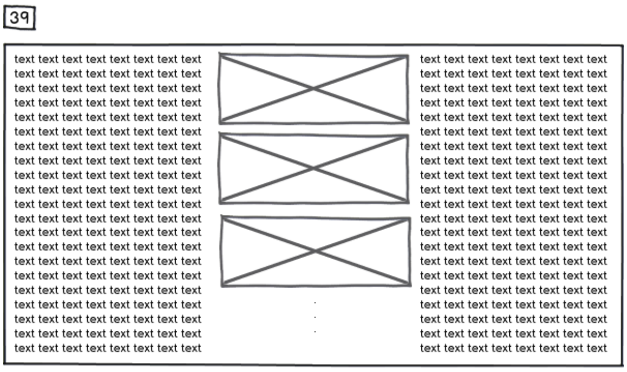

Figure 5. Advanced layout 39.

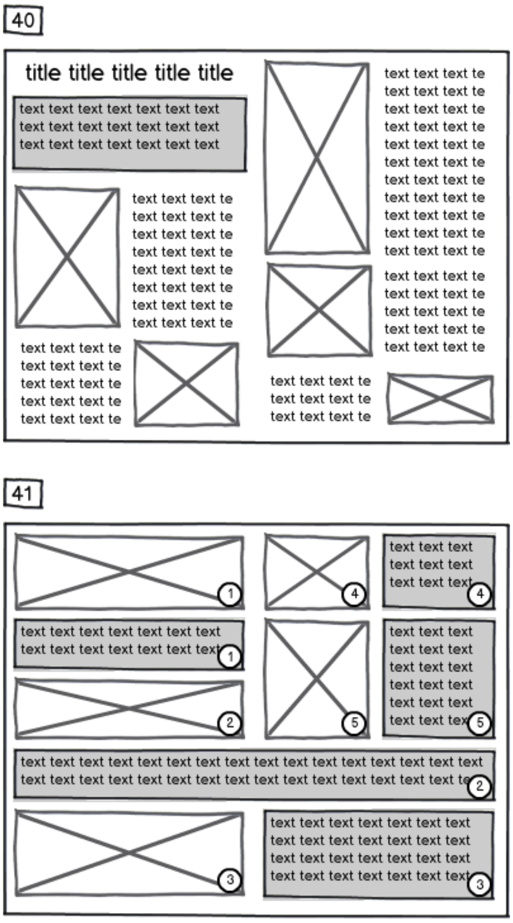
40
title title title title title
text text text text text text text
text text text te
41
text text text
1
2
3
4
5

Figure 6. Advanced layouts 40 and 41.

Layout 40 (Figure 6) provides a sample where a topic is commented by several people or based on different aspects. At the beginning there is a title and the summary as a highlighted text area. Next every comment contains an image and a description. The sequence of the comments is not fixed it can be modified based on the actual conditions. The length of the comments and the size of the images are different.

Layout 41 (Figure 6) arises an interesting topic: how to arrange images with their related text in order to (i) the appropriate text be close to the right image (above it, below it, or next to it), and (ii) to cover the available space with image and text tiles. The solution that our approach provides for it applies a genetic algorithm that searches for the first appropriate result, which covers the space in a way that the ratio of the empty spaces is less than 5%. In the figure the numbers show the coherent images and text areas.

Layout 42 (Figure 7) contains an image and a caption that are rotated.

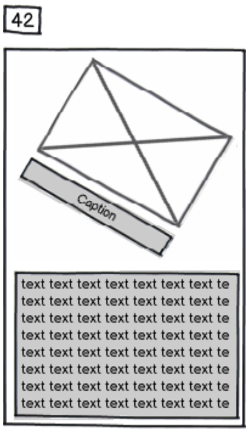

Figure 7. Advanced layout 42.

4 Conclusion

Device displays have reached the technical level which facilitates that the on-screen reading be a rival for the printed documents. Furthermore, we have several reasons to believe that when documents can look as good on the screen as they do in print, the on-screen reading experience will exceed the experience of reading on paper. This is because both desktop and mobile devices provide several opportunities for content and style customization, and there is a possibility for animation and interactivity as well. These features are going to make the on-screen reading experience superior in several ways.

In this paper, we have discussed the main features of our adaptive layout approach. Furthermore, we have presented our reference advanced layout arrangements.

The main distinctive elements of our layout approach are the followings:

- **Column-based templates**. Comparing to page-based templates, our approach facilitates to provide the same richness with less templates.
- **Tablets** are the target devices. We concentrate on **horizontal scrolling**.
- The solution focuses on the **online magazines**.
- A **high-level language** (easy to use for magazine editors) is applied to define the layout and the content related rules.
- **Content-driven layout**. The general layout rules, the rectangular areas related rules, the device capabilities, the size of the text, and the actual content drive the layout engine. Based on the actual content the algorithm determines which adaptation methods and which template to apply. Furthermore, within a template optional columns are skipped or inserted.
- **Managing aspect ratio of images**. Images have ideal cuts, tall and wide cuts, focus points, and protected areas. These properties drive the image resizing and the cutting of excess picture information. As a result images fit to the actual image areas while keeping their optimal aspect ratio.
- The **unit** that editors use during the template definition is the **height of a line**.

Currently we are working on the tool support that provides a template authoring tool for the magazine editors and hosts our layout engine algorithms.

Acknowledgements

This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Széchényi Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

References:

[1] Charles Jacobs, Wilmot Li, Evan Schrier, David Bargeron, and David Salesin. 2003. Adaptive grid-based document layout. *In ACM SIGGRAPH 2003*, ACM, New York, NY, USA, 2003, pp. 838-847.

[2] Evan Schrier, Mira Dontcheva, Charles Jacobs, Geraldine Wade, and David Salesin, Adaptive layout for dynamically aggregated documents, *In Proceedings of the 13th international conference on Intelligent user interfaces (IUI '08)*, ACM, New York, NY, USA, 2008, pp. 99-108.

[3] HTML 4.01 Specification, http://www.w3.org/TR/html401/

[4] CSS2.1 specification, http://www.w3.org/TR/CSS21/syndata.html#q10

[5] CSS 3 Multi-column Layout Module, http://www.w3.org/TR/css3-multicol/

[6] E. A. Meyer, Cascading Style Sheets: *The Definitive Guide*, Third Edition. O'Reilly Media, 2006.

[7] C. Anderson, *Essential Windows Presentation Foundation (WPF)*, Addison-Wesley, 2007.

[8] A. Nathan, *WPF 4 Unleashed*, Sams, 2010.