

QDITA ONLINE COURSE VISUAL EXPERIENCE FORMATION

MIRJANA MILOŠEVIĆ

IRVAS International d.o.o. mira.milosevic@irvas.rs

DRAGAN MITIĆ,

IRVAS International d.o.o. dragan.mitic@irvas.rs

PETAR CVETKOVIĆ

Belgrade Metropolitan University, petar.cvetkovic@metropolitan.ac.rs

Abstract: This paper focuses on the stylization of online courses. The key to DITA is that it separates form and format from the content, and removes the appearance of content from the structure. This paper also offers a preview of the BMU style of online course.

Keywords: Online Course, Stylization, CSS

1. INTRODUCTION

The Darwin Information Typing Architecture (DITA) defines a set of XML elements for creating and organizing content. However, the DITA specification does not offer references on how to transform DITA into user-readable documentation. The DITA Open Toolkit (DITA OT) fills that gap, providing a mechanism for transforming DITA content into multiple output formats, including HTML and PDF. The DITA OT formatting for both of these formats is basic, at best. This paper focuses on the changes that can be made to the DITA OT HTML output to create an attractive output. These modifications include changes to cascading stylesheets (CSS), headers and footers, and more advanced customizations.

The paper ¹ also illustrates an idea of separating content from its presentation.

2. TECHNICAL ASPECTS

The Darwin Information Typing Architecture (DITA) is an XML-based architecture for authoring, producing, and delivering topic-oriented, information-typed content that can be reused and single-sourced in a variety of ways. It is a standard that is applicable to any kind of publication or information that might be presented to readers, including interactive training and educational materials, standards, reports, business documents, and trade books, travel, nature guides, and more.

DITA is designed for creating new document types and describing new information domains based on existing types and domains. The process for creating new types and domains is called specialization. Specialization enables the creation of very specific, targeted document-type definitions that can still share the common output transformations and design rules developed for more

general types and domains; this is similar to how classes in an object-oriented system can inherit the methods of ancestor classes. Because DITA topics are XML conforming, they can be readily viewed, edited, and validated using standard XML tools. Although realizing the full potential of DITA requires using DITA-aware tools.

QDITA Styles

From the technical aspect, the QDITA style is a set of CSS and graphical elements that are implemented in software under the name “Publishing Style”.

One of the benefits of using DITA as a solo technical writer, is the separation of formatting from content development. Once stylesheets are set up, user is free to develop content without worrying about layouts and formatting. This has reduced the time needed to produce new documents.

In order to customize the formatting and styles of outputs, stylesheets need to be modified when developed. Both XHTML and HTML Help outputs are formatted by the same set of XHTML stylesheets. Customizing HTML outputs can be done using CSS styles or by modifying XSL templates:

- The simplest way to customize HTML outputs is by modifying the CSS styles used. This can be done by creating an additional CSS stylesheet and adding the appropriate properties to the ANT build script to associate it with HTML files. Individual DITA content elements can also be associated with a CSS class in custom stylesheet using the outputclass attribute.
- Adding parameters to the ANT build file for HTML output to specify an HTML running header and footer. Some custom HTML can be included in the head area of each page.
- For more advanced customizations, custom XSL transformations can be created. XSL transformations determine how DITA elements will be processed into

¹ This paper provides some results obtained by the project with code III44006 financed by the Ministry of Education, Science and Technological Development of Republic of Serbia

HTML tags. I.e. an XSL template specifies that the first topic title in an HTML page should be wrapped in an h1 tag. These transforms can be overridden so that elements are processed differently, or so that new ones could be added for elements that are specialized.

Output Classes

It is possible to give an HTML element multiple looks with CSS. For example, it could be desired to have large and white font, while at other times it could be preferred to have small and black font. CSS would not be very useful if it did not allow to have many different types of formats for a single HTML tag. CSS allows just that with the use of classes. Using classes is simple. It is required to add an extension to the typical CSS code and make sure to specify this extension in HTML.

DITA defines an attribute named `outputclass` that can be used with any DITA element. When the DITA OT processes a DITA element with the `outputclass` attribute, it automatically adds a class attribute to the corresponding output HTML tag, using the value of the `outputclass` attribute.

CSS

CSS (Cascading Style Sheets) allows web authors to apply styles to their web pages. More importantly, CSS enables them to do this independently of the HTML that makes up each web page. CSS is used to define styles against any HTML element by setting CSS properties.

Advantages of CSS:

- *CSS saves time.*
When learning HTML, one is taught to set the font face, size, colour, style, etc. every time it occurs on a page. This means typing (or copying and pasting) the same thing over and over again. With CSS, these details are specified once for any element. CSS will automatically apply the specified styles whenever that element occurs.
- *Pages load faster.*
Less code means faster download times.
- *Easy maintenance.*
In order to change the style of an element, you only have to make an edit in one place.
- *Superior styles to HTML.*
CSS has a much wider array of attributes than HTML.

Cascading Style Sheets, level 1 (CSS1) became a W3C Recommendation in December 1996. It describes the CSS language as well as a simple visual formatting model. CSS2, which became a W3C recommendation in May 1998, builds on CSS1 and adds support for media-specific stylesheets (e.g. printers and aural devices), downloadable fonts, element positioning and tables.

There are two basic approaches to applying styles in CSS:

- *Redefine the style of an existing tag.*
E.g. Redefine the styles for the body tag. These styles would then apply to the entire body section of the HTML page and can be used to apply page

backgrounds and set default font styles. Other tags can be redefined in the same way; p for paragraph styles, td for table cell styles, and li for list items etc.

- *Creating named styles and applying them to tags in HTML page individually.*
Named styles can be created in the CSS stylesheet by prefacing the name with a period (.) character. Styles in an HTML tag can be applied by using the class element. I.e. In CSS: `.mystyle {color:red}`; in HTML: `<div class="mystyle">`.

In DITA, a CSS style can be applied in the XML content by specifying the custom style name in the `outputclass` attribute of the DITA element. i.e `<step outputclass="mystyle">`. This will be processed to produce the correct class attribute in XHTML output types. Styles applied to HTML tags will override any style applied to a parent tag. i.e applying a style to a p tag will override the same style on the body tag.

CSS stylesheet files are associated with an HTML page using the `<link>` tag in the `<head>` section of the HTML. In the DITA Open Toolkit, the default spreadsheet `"commonltr.css"` is copied to the output directory and associated with each HTML page when building a XHTML output. If the same style is present in both CSS files, the one in the custom file will override the default, since the custom file is included after the default in the HTML source code.

3. VISUAL EXPERIENCE

After a course is published, it is displayed in a very simple way without stylization. For this reason, there is a need to create custom templates for more visual appealing online courses. A course is always published using only one template, but there may be multiple templates. A published course may be in HTML, PDF or other format. XML structure of the course is detained in the HTML view, in the way that elements have a certain order.

A template is a set of styles for course elements that are specified as CSS. One online course may have lessons, and every lesson has topics. Using navigation in content management system, a user can choose which course or lesson to preview. Every topic has the same design for certain elements such as header, footer, etc. A topic is divided into smaller sections that contain titles and content. Design for subtopics can be the same CSS class or the different, depending on the required representation.

4. GRAPHICAL DESIGN AND COURSE COMPONENTS

With each new technology that emerges, website visitors' attention spans shorten and people expect more speed from their web experiences. If web pages do not load quickly, the visiting surfer will leave the page and move on many designers are not aware of this fact. However, what confuses people is the belief that designing quick-loading pages necessarily leaves them no choice but to design simple, plain looking sites. This is not true. The

use of cascading style sheets (CSS) allows creating attractive pages that load quickly.

Page

The application has three types of pages that can be a part of an online course:

- Index page,
- Info page,
- Assessment.

The Index page has a course content list. The info page is the first page in the course, with an introduction describing the main purpose of the course. Tasks are the main components of the Assessment pages. Every task is a question that has offered answers.

Navigation

Navigation represents a set of options for going through a course, though navigation is not a part of a course. Depending on the content management system, navigation may contain an option for a Previous course, Next course, Continue, Exit and a Tree view of courses.

Header

The Header is the top rectangular shaped area that runs across the top of the web page screen. The primary purpose of a header is to promote the company's brand and to make it instantly recognizable to the audience. The audience should instantly know they have arrived at the correct place, if they are already familiar with company. The header is the most valuable space on the screen. Or, in a designer's term – it is the most expensive piece of screen real estate. The Header should always be used wisely, keeping in mind the goals and visions of website design.

The Header in the course may have a title and a suitable graphical design with logo of the company or institution's logo.

Footer

The Footer, as its name suggests, is the final piece of content located at the bottom of each page of website page. The purpose of the Footer is to signal to the user that they are at the bottom of the web page and to provide links or shortcuts to other areas of your web site.

The Footer in the course may have the Company name, year and also suitable graphical design like in the header.

Title

The Title in the course may have a different graphical representation. Setting different styles in CSS for:

- Text align,
- Color,
- Text transformation,
- Padding,
- Font family,
- Position,
- Margins,
- Background,

- Letter spacing,
- Border.

Paragraph

Similarly to the Title element attributes in a CSS, a Paragraph can have different styles, as well. The text-align property is used to set the horizontal alignment of a text. The text can be centred, or aligned to the left or right, or justified. When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers). The text-decoration property is used to set or remove decorations from text. The text-decoration property is mostly used to remove underlines from links for the purpose of design. The text-transform property is used to specify uppercase and lowercase letters in a text. It can be used to turn every letter into uppercase or lowercase, or capitalize the first letter of each word. The text-indentation property is used to specify the indentation of the first line of a text.

The color property is used to set the color of the text. With CSS, a color is most often specified by:

- a HEX value - like "#ff0000",
- an RGB value - like "rgb(255,0,0)",
- a color name - like "red".

CSS font properties define the font family, boldness, size, and style of text. The font family of text is set with the font-family property. The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font. The font-style property has three values:

- normal - The text is shown normally,
- italic - The text is shown in italics,
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported).

The font-size property sets the size of the text.

Being able to manage the text size is important in web design. Font size adjustments should not be used to make paragraphs look like headings, or headings look like paragraphs. Proper HTML tags should be used, like <h1> - <h6> for headings and <p> for paragraphs. The font-size value can be an absolute, or relative size. Absolute size:

- Sets the text to a specified size.
- Does not allow a user to change the text size in all browsers (bad for accessibility reasons).
- Absolute size is useful when the physical size of the output is known.

Relative size:

- Sets the size relative to the surrounding elements.
- Allows a user to change the text size in browsers.

Links

HTML offers many of the conventional publishing idioms for rich text and structured documents, but what separates

it from most other markup languages is its features for hypertext and interactive documents.

Links are found in nearly all Web pages. Links allow users to click their way from page to page. A link has two ends - called anchors - and a direction. The link starts at the "source" anchor and points to the "destination" anchor, which may be any Web resource (e.g., an image, a video clip, a sound bite, a program, an HTML document, an element within an HTML document, etc.).

A hyperlink (or link) is a word, a group of words, or image that can be clicked on to jump to a new document or a new section within the current document. By activating these links (by clicking with the mouse, through keyboard input, voice commands, etc.), users may visit these resources. By far the most common use of a link is to retrieve another Web resource. However, authors may insert links in their documents that express other relationships between resources than just simply "activate this link to visit that related resource". Links that express other types of relationships have one or more link types specified in their source anchor.

When a user moves the cursor over a link in a Web page, the arrow will turn into a little hand. Links are specified in HTML using the <a> tag. The <a> tag can be used in two ways:

- To create a link to another document, by using the href attribute.
- To create a bookmark inside a document, by using the name attribute.

The <a> element is usually referred to as a link or a hyperlink. The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue,
- A visited link is underlined and purple,
- An active link is underlined and red.

The visual representation of links can be modified using CSS. Setting styles of anchor elements will visually change color and text-decoration depending on state and mouse behavior:

- Active,
- Link,
- Visited,
- Focus,
- Hover.

Notes

The purpose of information notes is to inform the user about something relevant. This can be presented in blue because people associate this colour with information, regardless of the content. This could be any information relevant to a user action. For example, a note can show

some helpful information regarding the current user action or provide tips.

Warning

Warning messages should be displayed to a user when an operation could not be completed in a whole. For example "Your profile has been saved successfully, but confirmation mail could not be sent to the email address you provided". A usual warning color will be yellow and will have an exclamation icon.

5. PRESENTING A SAMPLE BMU STYLE

Here is an example of the graphical representation of a course using a Publishing Style. Lesson 5 has a list of topics on the Info page (Figure 1). After selecting the Lesson 5 Overview, a new window will appear (Figure 2). Every title (Introduction, Audience, Lesson Prerequisites and Objectives) has a specific style containing background image.



Figure 1 1: Lesson 5 Info page

Examples for stylization of course elements in CSS:

Body

```
body {
    color:#303032;
    font-family: Arial,Helvetica,sans-serif;
    font-size:13px;
}
```

Header

```
.topictitle1 {
    background-image:
url("../CSSImages/pozadinaMain3.jpg");
    font-size: 24px;
    text-align: right;
    text-shadow: 1px 1px 1px rgba(0, 0, 0, 0.9);
    color: #FFFFFFF;
    font-family: Arial,Helvetica,sans-serif;
    font-weight: bold;
    margin-bottom: 0.1em;
    margin-top: 0;
    padding: 68px 5px 0 10px;
}
```

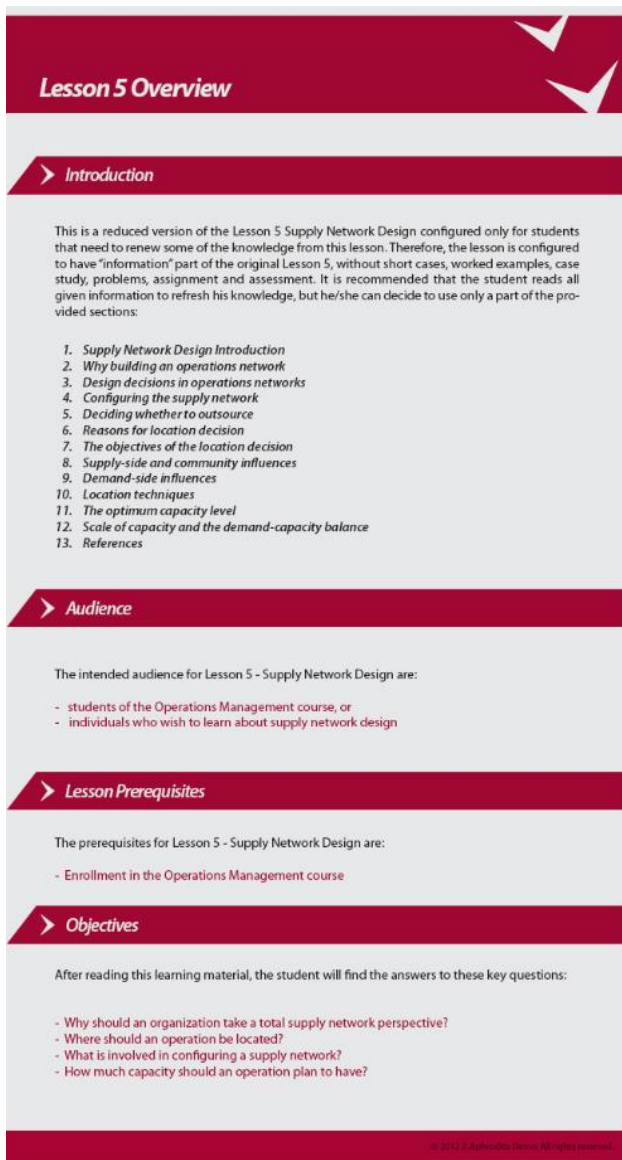


Figure 2: Lesson 5 Overview

Section Title

```
.sectiontitle {
  background: url("../CSSImages/pozadina4.jpg") repeat
  scroll 0 0 transparent;
  color: #FFFFFF;
  font-size: 14px;
  height: 30px;
  padding: 12px 0 0 75px;
  text-align: left;
  font-weight: bold;
  margin-bottom: 0;
  margin-top: 1em;
}
```

Section Body

```
.section {
  background-color: #FDFEE5;
  color: #000000;
  font-family: Arial,Helvetica,sans-serif;
  font-size: 13px;
  line-height: 18px;
  margin-top: 0;
}
```

Paragraph

```
p {
  padding: 10px 10px 10px 10px;
}
```

Links

```
A {
  background-image:url("../CSSImages/link1.png");
  background-repeat:no-repeat;
  height:25px;
  color:#06a9cc;
  padding:0px 0px 0px 18px;
}
```

Footer

```
#Footer {
  font-family: Verdana, san-serif;
  font-size: 9px;
  color: #e2e55a;
  text-align:right;
  background-image:url("../CSSImages/plavo.jpg");
  width: 650px;
  height: 35px;
}
#Footer p {
  font-size: 1.2em;
  padding-right: 5px;
}
```

6. CONCLUSION

DITA allows you to produce documentation in multiple formats from a single set of XML source files. The current open source toolkit provides support for HTML, HTML Help, JAVA Help, Eclipse Help and PDF output. DITA eliminates the need to maintain multiple versions of the same documents; a process which can be both error-prone and time consuming. Content can be reused within and across topics and output formats, making updating documents easier eventually reducing development time. Some of the things that DITA can bring to learning content are:

- Consistency of content, processing, and delivery.
- Can capture instructional design information in a consistent way, such as objectives, audience, and other metadata.
- Can enable reuse between learning-specific content and other existing content.
- Can grow as DITA grows.

LITERATURE

[1] Quackit, *CSS & Design*, <http://www.quackit.com/css/> August 23, 2012.

[2] Stephanie (Sullivan) Rewis, *Simple styling with CSS*, http://www.adobe.com/devnet/dreamweaver/articles/css_simple_styling.html, August 23, 2012.

[3] Brent Kelly, *Web design tips*, <http://www.zeald.com/Blog.html>, August 24, 2012.

- [4] W3 Schools, *CSS Tutorial*, <http://www.w3schools.com/css/default.asp>, August 24, 2012.
- [5] Oasis, *Darwin Information Typing Architecture (DITA)*, <http://docs.oasis-open.org/dita/v1.2/os/spec/archSpec/introduction-to-dita.html#introduction-to-dita>, August 24, 2012.
- [6] Louise Kasemeier, *DITA for Solo Writers*, Pages 41-43, 2007.
- [7] John P. Hunt, *Designing and Delivering Dynamic Training and eLearning with DITA*, 2007.
- [8] Dave Clark, *Content Management and the Separation of Presentation and Content*, <http://rhetcomp.gsu.edu/~bgu/8121/TCQ-CMS%28Clark%29.pdf>, August 24, 2012.