Accessible EPUB Standard

# About Accessible EPUB

Accessible EPUB started as bachelor thesis project to create an EPUB standard which allows people with regular vision, visually impaired and blind to use the same document. At the same time, the document has to support the various requirements each group has, such as sans-serif text for visually impaired, alternative text to be easily accessible and LaTeX code to appear instead of mathematic formulas. Furthermore, math should not appear as images, but as MathML, which can be read aloud by the screen reader and can be zoomed in without quality loss. Text must use the preset formats, such as Heading 1 to Heading 6 in HTML.

As a result two standards were created, one which uses JavaScript to do the switching, and one which uses CSS. JavaScript works better, but is not officially supported in the EPUB 3 standard. It can be supported, but that is optional and up to the EPUB reading system developer. CSS 3 is officially supported by the EPUB 3 standard, but many reading systems do not support it yet. Nevertheless, a standard was created with CSS in case support for CSS 3 becomes widespread. The CSS standard has less features than the JavaScript standard, as it does not support table of contents and multiple HTML files.

EPUB 3 uses XHTML and therefore should be written in XHTML, which differs slightly from proper HTML. Tags should be self closing, such as <br> in HTML becoming <br/> in XHTML. Henceforth all HTML elements will be called XHTML elements so that there no constant changing between the terms.

For ease of use, the standard uses normal XHTML links to switch between the versions, which appear at the beginning of the document and persist throughout the document. There should no font set in the XHTML document, as that would not allow the font to be changed with CSS. The font for each target group is already set in the CSS files.

All images and formulas are enclosed in figures. This is done so that captions in the form of the figcaption element can exist. Furthermore, the Accessible EPUB editor needs to have figures to allow grouping of the element for all three display styles, visually impaired, blind and normal sighted.

# JavaScript

The Accessible EPUB JavaScript standard works with using local and session storage. Once one of the style changing links are clicked, the it first attempts to save the choice in local storage first. If this does not work, it tries session storage next. It then loads the corresponding CSS file immediately, but also after each time the XHTML get loaded again.

The JavaScript standard has an additional XHTML page, named VersionChanger.xhtml. It contains the links to change versions.

## Content.xhtml

The head of the document has a reference to style.css, which imports each of the specific CSS documents. The script also is referenced.

It is important that the body of XHTML document has the property, onload="storageCSS();", which loads the style written in local or session storage.

# CSS

The Accessible EPUB CSS standard is much more complicated. It uses the :target CSS selector. Using only CSS means that the style selection (normal, visually impaired and blind) is not persistent across XHTML files, so all of the content is saved in one XHTML page by hiding the other style sections. As a result, table of contents can not be created for the CSS standard, as they would have each element three times.

## Content.xhtml

The head of the document has a reference to style.css.

The first three lines of the body are links to the appropriate div element, such “impaired” being the title for the visually impaired div element. There are three major div elements, “visible”, “impaired” and “blind”. Each div has its own CSS class, which are display styles for group. If the correct link is clicked on, it hides the other two sections and makes the one section appear. The actual content of the document, like text, images, etc., has to be inserted in each section. The content does not have to differ for each group, as each element will have a CSS class which does not appear if they are within a class. For example, the class .mathImpaired does not appear in the .visible and .blind class, but does appear in .impaired.

# Files changes

For both standards the files which change during the document creation process are the content files, Content.xhtml. The table of contents file, nav.xhtml also changes for the JavaScript standard. Referenced image files have to be copied in the images folder. The package document, content.opf, has to reflect these changes. Each new image should be an item with the correct media-type. All other files can be kept from the template EPUB file.

# Images

Images need to have alternative text and a title.

The following is part of the code used to insert an image in the Accessible EPUB editor, edited only so that part of it makes more sense (with extra quotations marks to escape other quotations marks):

@”<figure" + styleTag + heightTag + widthTag + @"><img class=""imageOthers""" + styleTag + @"title =""" + title + @"""src =""" + imagePath + @""" alt =""" + altText + heightTag + widthTag + @"""><img class=""imageImpaired""" + styleTag + @"title =""" + title + @"""src =""" + altImagePath + @""" alt =""" + altText + heightTag + widthTag + @"""><p class=""transparent"">" +

tag + altText + tagEnd + @"</p><figcaption style = ""text -align:center"" >" + caption + @"</figcaption></figure>

All elements are given in the insert image dialog box.

The styleTag is how the float property is set, so either nothing or style="float:left;" or style="float:right;”.

The heightTag and widthTag are the sizes in pixels, written as height=\"" + height + "px\" and width=\"" + width + "px\". Both don’t have to have a number and can get the dimensions from the CSS img style, where the maximum width is 100%.

The title follows the same pattern, but the user can decide their own title without any limits.

The imagePath should be the relative path, not the absolute path.

The altText follows the same like the title. It also appears as a paragraph element, as that makes it easier to read with a screen reader.

The altImagePath is like the imagePath, but if an alternative image does not exist, it should be identical to the imagePath. It is not allowed to be empty.

The tag should tell the blind reader what kind of element the image is. The default is <Image> and </Image>. The tag element should be "&lt;" + tag + "&gt;" as the greater and lesser symbol in XHTML have to be encoded. tagEnd should be similar and use the same tag, but has to be a closing tag: "&lt;/" + tag + "&gt;".

The caption is like the title and altText.

# Math

Math in Accessible EPUB is inserted with LaTeX code, which then has to be converted to MathML. This was done with Pandoc, but any program which convert LaTeX to MathML is fine. The various code segments of inserting math is below:

“\n<figure>”

@"<img class=""toRemove"" " + titleTag + @"src =""" + imagePath + @""" alt =""" + latexCode+ @""" //>"

@"<div" + styleTag + @" role=""math"" class=""math""><math xmlns=""http://www.w3.org/1998/Math/MathML"" altimg=""" + imagePath + titleTag + @""" alttext=""" + latexCode + @""">" + "" + "<mstyle>"

MathML code

"</mstyle></math>"

"</div>"

@"<div" + styleTag + @" role=""math"" class=""mathImpaired""><math xmlns=""http://www.w3.org/1998/Math/MathML"" altimg=""" + imagePath + titleTag + @""" alttext=""" + latexCode + @""">" + "" + "<mstyle scriptsizemultiplier=\"1\" lspace=\"20%\" rspace=\"20%\" mathvariant=\"sans-serif\">"

MathML code

"</mstyle></math>"

"</div>"

"<p class=\"transparent\">$" + latexCode + "$</p>";

"<figcaption style=\"text-align:left\">" + caption + @"</figcaption>";

"</figure>\n"

The titleTag is either what the user has chosen as title or a random file name. The title is used to save an svg image of the math formula, so it can later be referenced for the altImage property. Furthermore, without the svg image of the formula, it is not compatible the Accessible EPUB editor as it is a requirement for it. A LaTeX to image parser has to be used. In the Accessible EPUB editor the .NET library WPFMath was used, but any LaTeX to svg converter can be used.

The styleTag and imagePath are like they were with images. The latexCode is used for alternative text, so it has to be there even if the MathML was generated without LaTeX.

It is important that the mstyle tag is changed in the visually impaired section, as they have different requirements, such as exponents not being smaller and the font being sans-serif.

The MathML code should only consist of the part between the mstyle opening and closing tags.

# Tables

Tables need to have a title element. That is the only requirements tables need to have.

# Further Information

Further information can be obtained from looking at both an empty and a sample file of both variants of the standard. Many things might seem like they are not required, but they are likely needed due to the Accessible EPUB editor having constraints. For example, the Accessible EPUB editor uses the WebBrowser element in C# and this is based on Internet Explorer. Internet Explorer can not display MathML so that is why image elements still have to be enclosed in figures, even if they are hidden when the EPUB is opened.

The website of the project is <https://github.com/zyxsachin/AccessibleEPUB/>

For any questions or comment, please feel free to contact [sachin.rajgopal@student.kit.edu](mailto:sachin.rajgopal@student.kit.edu)