New Avenues for Electronic Publishing in the Age of Infinite Collections and Citizen Science 37
B. Schmidt and M. Dobreva (Eds.)

doi:10.3233/978-1-61499-562-3-37

Using EPUB 3 and the Open Web Platform for Enhanced Presentation and Machine-Understandable Metadata for Digital Comics

Pieter HEYVAERT [a,1], Tom DE NIES [a], Joachim VAN HERWEGEN [a], Miel VANDER SANDE [a], Ruben VERBORGH [a], Wesley DE NEVE [a,b], Erik MANNENS [a] and Rik VAN DE WALLE [a]
[a] *Multimedia Lab, Ghent University – iMinds*
[b] *IVY Lab, Korea Advanced Institute of Science and Technology (KAIST)*

Abstract. Various methods are needed to extract information from current (digital) comics. Furthermore, the use of different (proprietary) formats by comic distribution platforms causes an overhead for authors. To overcome these issues, we propose a solution that makes use of the EPUB 3 specification, additionally leveraging the Open Web Platform to support animations, reading assistance, audio and multiple languages in a single format, by using our JavaScript library comicreader.js. We also provide administrative and descriptive metadata in the same format by introducing a new ontology: Dicera. Our solution is complementary to the current extraction methods, on the one hand because they can help with metadata creation, and on the other hand because the machine-understandable metadata alleviates their use. While the reading system support for our solution is currently limited, it can offer all features needed by current comic distribution platforms. When comparing comics generated by our solution to EPUB 3 textbooks, we observed an increase in file size, mainly due to the use of images. In future work, our solution can be further improved by extending the presentation features, investigating different types of comics, studying the use of new EPUB 3 extensions, and by incorporating it in digital book authoring environments.

Keywords. Dicera, Digital comics, EPUB 3, Enhanced presentation, Linked machine-understandable metadata, Open Web Platform

1. Introduction

The market for digital comics is growing, consisting of both people familiar with print comics and newcomers. The revenue numbers of the past years, as obtained for the North American market, testify to this statement [1]: (1) a growth between $640 million in 2011 and $680 million in 2012 for print comics and (2) a growth between $25 million in 2011 and $70 million in 2012 for digital comics. After a decline in 2010 for print comics, the revenue for these comics is increasing again, together with the revenue for digital comics, as shown in Figure 1. This figure also proves that the growth of the digital market does not make the print market shrink.

[1]Corresponding author. E-mail: pheyvaer.heyvaert@ugent.be

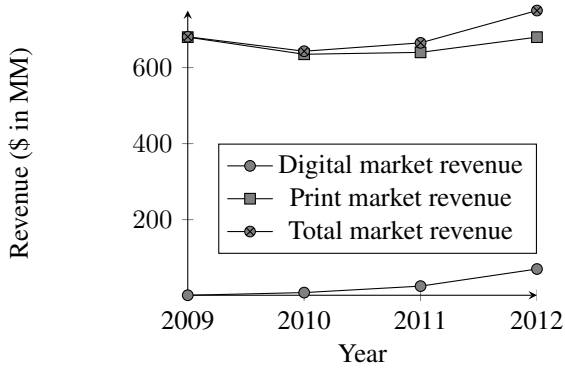


Figure 1. After a decline in 2010 for print comics, the revenue for print comics is increasing again. The revenue for digital comics is increasing as well (data gathered from [1]). The growth of the market for digital comics does not make the market for print comics shrink.

Authors have a number of distribution platforms at their disposal, including comiXology [2] and the iBooks Store [3]. However, these platforms use different (proprietary) formats to distribute comics, thus storing the same information in different packages. In addition, several platforms assume that all publications are textbooks, which can result in bad on-screen formatting. Furthermore, several platforms also require authors to add additional information (e.g., panel annotations), so that comics enable every feature of the dedicated reader application (e.g., for tablets and smartphones). All these factors contribute to an increased *production overhead*, while authors primarily want to focus on telling stories.

Tools [4, 5] exist to convert a comic to different formats used by the different distribution platforms. However, the use of these tools might still require intervention by the authors, because the *feature sets of the different formats differ*. This is *cumbersome for authors*. Other problems are related to the *extraction of information* from comics (e.g., separate panels, characters, text, and so on), both in the domain of presentation and metadata. While this information is present in a number of comics, it is not available in a machine-understandable way, which can improve discoverability.

We argue that the aforementioned problems are caused by the digital formats in which the comic books are stored. As such, we propose an Open Web Platform-based solution that has the following merits:

- it **circumvents the conversion** of a comic to the different formats required by different distribution platforms;
- it **alleviates the use of extraction methods**, which are currently powered by image processing, by providing the descriptive information, i.e. information about characters, pages, panels, and so on, through machine-understandable metadata, facilitated through our newly proposed ontology Dicera;
- it **works towards truly digital comics**, i.e. comics that allow for the use of animations, audio, reading assistance, multiple languages and machine-understandable metadata, facilitated through our newly proposed JavaScript library *comicreader.js*.

The remainder of this paper is organized as follows. In Section 2, we review related work. Next, in Section 3, we provide an outline of the technical requirements identified and the solution proposed. In Sections 4 and 5, we discuss the presentation and metadata elements of the proposed solution in more detail, respectively. We subsequently discuss and pay attention to an evaluation of the proposed solution in Section 6. In Section 7, we give an overview of future work. Finally, we provide concluding remarks in Section 8.

2. Related Work

Arai and Tolle [6] propose a method that allows for the automatic extraction of frames and their content, which includes text balloons and their text, resulting in better accuracy and processing time, compared to other methods.

Hoashi et al. [7] suggest to create thumbnails for each comic to more easily identify the content of a single episode of a comic series. Thumbnails are created by extracting the different frames and selecting the ones with the highest scores (using the frame's features and a linear regression model). Their results state that the proposed method enables users to search comic episodes faster.

The main problem Yamada et al. [8] handle is how to display high-resolution comic pages on low-resolution cellular phones. The authors developed a system that consists of frame detection, text extraction and layout analysis.

Print versions restrict the access for a number of people: the visually impaired, the motor-impaired and the users of mobile devices. To enhance the navigation through a comic, Ponsard and Fries [9] propose a solution that performs the following three steps: sequential ordering of the page-wide files, segmentation of each page into panels and sequential ordering of the panels in the right reading order.

Arai and Tolle [10] propose a new online method for automatically extracting text from text balloons in digital comics. This method has an accuracy of 100% for frame extraction from flat comics and an accuracy of 100% for balloon detection. Text extraction achieves an accuracy of 93.75%. The authors suggest that the new method is useful for the automatic translation of Japanese comics into international comics.

Morozumi et al. [11] summarize the basic requirements for a metadata framework for manga (the Japanese version of comics): different levels of description for manga elements, a clear difference between the intellectual entity and the publication of a manga, and the identification and description of the elements that make up a manga. This metadata framework is implemented by Mihara et al. [12].

With eDBtheque, Guérin et al. [13] want to create the first comics database with a ground truth for descriptive metadata. This database is built by using a visual segmentation protocol with guidelines regarding text lines, balloons and panels, and by annotating these text lines, balloons and panels.

Given the above research efforts, we can identify a number of trends. First, we can conclude that the need for information about the content of a comic steers current research efforts towards the development of extraction methods that make use of image processing techniques. The use of these computationally heavy techniques can be circumvented. Second, we can conclude that the research efforts regarding comics metadata are limited to basic administrative metadata (e.g., title, author and genre) and descriptive metadata (e.g., the characters appearing in a comic and the number of panels on a page). These descriptive metadata are stored outside the comic. However, the inclusion of de-

scriptive metadata in a comic can aid in the presentation of the comic content to different types of users (e.g., impaired users and users of mobile devices).

3. Requirements and Proposed Solution

The proposed solution consists of two major parts: presentation and metadata. The requirements for the first part, as identified through an analysis of related work (see Section 2) and the features of the (comic) applications of Marvel [14] and comiXology, are (1) support for *animations*, (2) support for different types of *devices*, (3) support for *reading assistance*, (4) support for *audio* and (5) support for *multiple languages*.

The requirements for the second part, as identified through on an analysis of related work, are (1) support for *extended administrative metadata* and (2) support for *extended descriptive metadata*.

To meet the aforementioned requirements, the EPUB 3 format [15] was chosen as the foundation of the proposed solution. Our motivation is as follows: (1) EPUB 3 is widely used [16], (2) EPUB 3 targets digital publishing, hence also digital comics, (3) the Open Web Platform (that is, HTML5, CSS3 and JavaScript) allows for a large range of possibilities, and (4) Ghaem Sigarchian et al. [17] concluded that EPUB 3 supports the most desirable attributes of an enhanced publication (that is, a publication enriched with multimedia and interactivity features). Note that the requirement to support audio is immediately fulfilled by EPUB 3, given that EPUB 3 makes use of HTML5.

In what follows, we discuss the presentation elements (Section 4) and the metadata elements (Section 5) of the proposed solution in more detail[2].

4. Presentation

In Section 4.1, we detail the use of layering. In Sections 4.2 to 4.4, we discuss the support for animations, reading assistance, different types of devices and multiple languages. Next, we explain the use of scripting languages in Section 4.5. In Sections 4.6 and 4.7, we discuss the consequences of scripting and layer unavailability, respectively. Finally, in Section 4.8, we look at the use of additional EPUB 3 specifications.

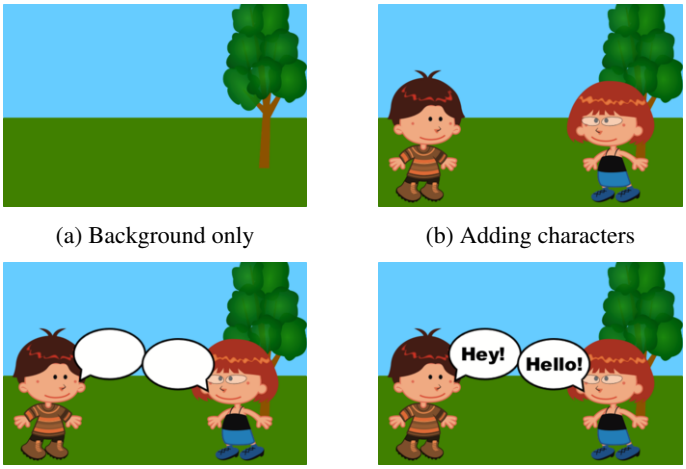
4.1. Layering

Before tackling the requirements, we define the way the graphical information is stored, which will enable the solution to fulfill the requirements. A page is not stored as one image. Instead, each panel is stored separately in a `<div>` element. This is taken even further: by using layering, every panel is stored as a group of images, and thus not as a single image. Every image is called a layer, hence the name layering. A layer can be created for the background, for each character and for each text element (e.g., a speech balloon, a caption or an effect), by representing each layer by a `<div>` element. This makes it possible to do manipulations on separate parts of the panel itself.

4.2. Support for Animations

To support animations in our solution, we rely on the Open Web Platform. JavaScript and the CSS property `display` are used to show the different layers. Figure 2 illustrates the concept of animations through an example. First, the background of a panel is shown, followed by the characters, speech balloons and the text.

[2]A screen cast of the presentation part of the proposed solution can be found at `http://users.ugent.be/~pheyvaer/digital_comic.mp4`.



(a) Background only (b) Adding characters

(c) Adding speech balloons (d) Adding text

Figure 2. A panel is built by adding layers ((a)-(d)) incrementally.

4.3. Support for Reading Assistance & Different Device Types

Reading assistance refers to zooming in on the panel that the user is currently reading and navigating to a new panel once the whole panel has been read (determined by user input). To accomplish this, a jQuery [18] plugin called Zoomooz [19] is used. This JavaScript library makes it possible to zoom in onto elements of Web pages, while taking into consideration the screen size of the device. In our case, these elements are the panels of a comic. An important remark is that our approach towards facilitating reading assistance and support for different types of devices is *independent of the reader application*, because it is achieved through scripting in the EPUB 3 file itself, and can, hence, be *different for every comic*.

4.4. Support for Multiple Languages

To allow for multiple languages in the same digital comic and to allow for switching between the languages available, the different translations of a piece of text are stored in different 'sublayers' (`<span>` elements instead of `<div>` elements are used, because span elements are used for inline elements and the different translations can be viewed as the inline elements of the text `<div>`). This allows showing the (text) sublayers of the currently selected language and hiding the other languages. Hiding and unhiding the different sublayers is again accomplished by making use of JavaScript and CSS.

In Listing 1, an example can be found of the use of two languages, Dutch and English. One layer represents all the text, which means both languages. Inside this layer, another `<div>` element represents the text object, needed for the addition of metadata. This `<div>` element has two `<span>` elements (the sublayers previously mentioned), one for each language. The `xml:lang` attribute is used to denote the Dutch and English translation. For example, if the user wants to read the comic in English, the CSS property `display` of all the `<span>` elements, with the value of the `xml:lang` attribute set to `nl`, are set to `none`, which makes those elements hidden, leaving only the English translation visible. In addition, using CSS, the font of the text, together with the size and the position on the panel, can be set.

Listing 1: Example of multi-language support

```
<div id='panel_text'>
    <div id='panel_text_obj'>
        <span id='panel_text_nl' xml:lang='nl'>Hoe het begon ...</span>
        <span id='panel_text_en' xml:lang='en'>How it began ...</span>
    </div>
</div>
```

4.5. Use of Scripting Languages

Most of the JavaScript code is not inherent to a single comic. Therefore, it is useful to create a JavaScript library to bundle all the code that is reusable. First, we list the requirements of such a library. Second, we present our implementation of such a library.

First, the library should contain functions that can be used to provide animations. Besides the standard 'appear' animation, it should be possible to add other (basic) animations, such as a slide, a dissolve, and so on. Furthermore, the functions need to be designed in such a way that they can be reused if a developer wishes to create his or her own animations. Second, the library should have support for reading assistance. To that end, we rely on the Zoomooz library (and jQuery). Third, the library should be able to handle different languages. In particular, the library should have support for switching between languages by manipulating the different layers. Given our proof-of-concept EPUB 3 file, we grouped the common functionality of its XHTML pages in an extensible JavaScript library called *comicreader.js*[3].

4.6. Unavailability of Scripting

According to the EPUB 3 [15] specification, it is not allowed to rely on scripting to deliver content to users. Scripting can only be used to enhance the user experience. Our fallback method, in case scripting is not possible or (temporarily) disabled on a device (or in an application), is to display the whole page at once: layering is not used anymore, no support for animations, no support for reading assistance, no support for audio and no support for multiple languages. For every panel, one image is included as a fallback, and this fallback image will be displayed when it is not possible to execute scripts.

4.7. Unavailability of Layering

It is possible that layers are not available, for instance when print comics are converted to our solution. This has the following consequences. Adding animations to a single panel will not be possible, given that the different layers to work with are not available. Support for multiple languages is not possible in the way it is defined in this paper, because the text is hard coded on the image. However, it is possible to define a `<div>` element that covers the original text with the translation needed. Audio can still be added, together with reading assistance and support for different types of devices.

4.8. EPUB Region-based Navigation and Multiple-Rendition Publications

The EPUB Region-based Navigation specification [20] allows adding region-based navigation through a visual rendition of a publication based on Regions of Interest. The major problem with this specification is the lack of layering, hence the inability to deal with animations. Although the specification incorporates metadata about regions of interests, these metadata are limited to the name (in string representation) of one character.

[3] comicreader.js and our proof-of-concept EPUB 3 file are available at https://github.com/mmlab/comicreader.js.

The EPUB Multiple-Rendition Publications specification [21] defines the creation and rendering of publications consisting of multiple renditions. In what follows, we discuss the effect of this specification on the solution proposed. Considering the use of multiple languages, the difference between our approach and the approach proposed in the specification is that every language should point to a different rendition. Our solution uses one file for each page that includes all the languages. Splitting those languages in different files would force a developer to duplicate all the other code inside the files, with the exception of the actual text of the comic. Hence, redundancy is created and the file size of the EPUB publication will increase (unnecessarily). Studying different approaches is proposed, such as using renditions with multiple languages using one file that includes all the languages, together with the use of JavaScript. Each rendition uses a different script that displays the correct languages and hides the other languages (cf. the approach previously discussed in Section 4.4). The downside is that problems may arise when scripting is disabled (see Section 4.6). This is not an issue when using the approach described in the specification.

5. Metadata

In Section 5.1, we discuss the different types of metadata. In Section 5.2, we discuss how the metadata is modeled, paying attention to the support for extended administrative and descriptive metadata in Sections 5.2.1 and 5.2.2, respectively. Finally, we look at the consequences of layer unavailability in Section 5.3.

5.1. Different Types of Metadata

All metadata can be divided into two groups based on the location where they are stored: *local metadata* or *remote metadata*. With remote metadata, we denote all the metadata that are not necessarily stored inside the EPUB file itself. A publisher can store all the metadata of a certain comic character on a server (e.g., biography, latest comics, and so on). We leave the decision about where to store what metadata to the user.

Local Metadata The metadata stored locally enables the user to gain a number of advantages of linked data, such as discoverability, shareability and re-usability.

Remote Metadata The motivation for also storing metadata remotely is twofold. First, storing all metadata locally would increase the size of the EPUB file. Second, storing all metadata locally would create redundancy, making it more difficult to keep the metadata up to date.

5.2. Data Modeling

To structure the metadata, we developed the ontology Dicera[4] (DIgital Comic book ERA vocabulary). Classes and properties of Dublin Core Metadata Initiative Metadata Terms [22] and the DBpedia Ontology [23] that could be reused are not redefined in Dicera (e.g., characters and locations). The use of the schemas provided by schema.org [24] and Friend of a Friend [25] have been considered, however, they are lacking the necessary concepts.

To add metadata to a digital comic, we use RDFa [26]. This tool, which was recently added to the EPUB 3 specification, allows adding metadata to Web pages.

[4]Dicera can be found at `http://semweb.mmlab.be/ns/dicera`.

5.2.1. Support for Extended Administrative Metadata

As part of the local metadata, we extended the current metadata (e.g., title, author, genre, and so on) available in EPUB 3 with (more detailed) information regarding story arcs, issues, genres and content ratings, through Dicera. Indeed, as an example, every comic can be associated with a certain story arc. This happens through an issue entity, which also includes the issue number. For each genre (a single comic can have more than one genre), the percentage it matches to the story can be denoted, given that genres can be quantified, i.e. a story can be only 25% drama (not necessarily 100%).

5.2.2. Support for Extended Descriptive Metadata

We added the following descriptive metadata, through Dicera: covers, pages, panels and text elements. The pages are connected to the contained panels. These panels are connected to their characters, locations, objects and text elements. Every character, location and object can also be connected to the cover if they appear on it.

5.3. Layer Unavailability

Layer unavailability puts limitations on the presentation part of the solution. The limitations on the metadata part are less severe. Each panel can still be annotated with corresponding characters, however, all characters will be included in the same layer. This is also valid for the locations, objects and text elements. Even if the text is hard coded on the image, text objects can still be used. These objects will solely be used for metadata purposes and not for both presentation and metadata as is normally the case. The most important limitation is the following: if the metadata are used to determine all the panels where, e.g., a character is present, it is only possible to get the complete panel/image. When layers are present, it is possible to retrieve images containing only the requested character, without the background, the other characters, the objects and text elements.

6. Discussion & Preliminary Evaluation

In this section, we discuss the need for extraction methods, the change in file size, the support for different publication platforms and the support for different reading systems.

6.1. Extraction Methods

The need for extraction methods, as mentioned in Section 1, can become superfluous if all the required information is already available in the solution. However, extraction methods and metadata information can be used together. Initial metadata can be added by using the information acquired from the extraction methods, and can subsequently be enhanced by manually correcting or adding information.

6.2. EPUB File Size

Comparing the file size of our comic to the file sizes of textbooks in EBUP format shows that our file size is more than 60 times larger than the average file size of a textbook. The sole reason for this is the presence of pixel-based image files in a comic. A consequence is that applications can no longer assume that an EPUB file comes with a rather small file size.

6.3. Distribution Platforms

From the work conducted here, we conclude that the features offered by different distribution platforms are replicable using the proposed solution.

The use of the proposed solution by distribution platforms eases the publication of content for authors to those platforms, because a single, standardized and actively de-

veloped format is used. Note that publishers might desire their own dedicated application to offer a special and unique experience to their readers. However, this is not valid. The presentation functionality does not depend on the reading system. It is embedded in the EPUB file itself, allowing it to be read on every system supporting the EPUB 3 specification. Based on these benefits, the companies behind these platforms might consider the solution, and eventually replace their current format by an open format such as ours. It also allows them to discontinue the development of their custom application for reading comics or to work towards a reading system supporting the (complete) EPUB 3 specification.

6.4. Reading System Support

Readium [27] (with the use of scripting), developed by IDPF [28], and the Kobo Glo [29], a dedicated e-reader (using the fallback panels) allow reading comics that make use of the proposed format. Unfortunately, iBooks [3], an iOS application, and Calibre [30], a desktop application, are not able to display these comics in a correct way, both with and without scripting. By testing the solution on these four different types of reading systems (that is, a web app, a dedicated e-reader, an iOS application and a desktop application), we have a basic global view of the reading system support for our solution.

7. Future Work

Future work is possible, both in the direction of the enhancement of the proposed solution and the creation of applications, reading systems and frameworks that work with the solution, so to enhance its adoption. Enhancing the solution can be done by investigating the support for different types of comics, by extending and optimizing the presentation features, by studying the use of the new (draft) specifications of EPUB 3, and by studying how the solution can be used to recommend new comics to readers. Furthermore, work needs to be conducted to create a remote server to work with the solution and a framework that links comics to other multimedia content such as movies and games, and to develop a reading system that fully supports the solution. While the advantages of enhanced digital comics are clear, authoring them in an efficient manner remains a major challenge, which prohibits their wide adoption. However, as digital book authoring solutions evolve, opportunities arise to adapt them towards digital comics. For example, part of our future work is to adapt the authoring environment described in [31] to create and manage enhanced digital comics using our proposed approach.

8. Conclusion

In this paper, we tackled problems regarding the extraction of information from (digital) comics by making use of EPUB 3 and adding metadata by making use of RDFa and the ontology Dicera. By offering presentation features using the Open Web Platform the proposed solution, with comicreader.js, circumvents the use of different (proprietary) formats by different publication platforms (causing overhead in the production process), the cumbersomeness of the use of format conversion tools for the different distribution platforms, and the differences in the feature sets of these formats. The increased file size compared to textbooks, and the currently limited EPUB 3 support by reading systems, prevent large-scale deployment of the proposed solution for the time being. However, since a standardized format is used that is being actively developed, reading system support will only increase in the future.

9. Acknowledgements

The research activities were funded by iMinds, Ghent University, the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research Flanders (FWO Flanders), and the European Union.

References

[1] Digital comics nearly tripled in 2012. http://www.icv2.com/articles/news/26202.html, Retrieved 2014/05/08.
[2] Comics by comiXology. https://www.comixology.com/, Retrieved 2014/11/12.
[3] iBooks. http://www.apple.com/ibooks/, Retrieved 2014/11/12.
[4] EPUB to MOBI. http://www.epub2mobi.com/, Retrieved 2014/12/05.
[5] EPUB Converter. http://www.epubconverter.com/, Retrieved 2014/12/05.
[6] Kohei Arai and Herman Tolle. Automatic e-comic content adaptation. *International Journal of Ubiquitous Computing*, 1(1):1–11, 2010.
[7] Keiichiro Hoashi, Chihiro Ono, Daisuke Ishii, and Hiroshi Watanabe. Automatic preview generation of comic episodes for digitized comic search. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1489–1492. ACM, 2011.
[8] Masashi Yamada, Rahmat Budiarto, and Shinya Miyazaki. Comic image decomposition for reading comics on cellular phones. *IEICE transactions on information and systems*, 87(6):1370–1376, 2004.
[9] Christophe Ponsard and Vincent Fries. An accessible viewer for digital comic books. In *Computers Helping People with Special Needs*, pages 569–577. Springer, 2008.
[10] Kohei Arai and Herman Tolle. Method for real time text extraction of digital manga comic. *International Journal of Image Processing (IJIP)*, 4(6):669–676, 2011.
[11] Ayako Morozumi, Satomi Nomura, Mitsuharu Nagamori, and Shigeo Sugimoto. Metadata framework for manga: a multi-paradigm metadata description framework for digital comics. In *International Conference on Dublin Core and Metadata Applications*, page 61, 2009.
[12] Tetsuya Mihara, Mitsuharu Nagamori, and Shigeo Sugimoto. A metadata-centric approach to a production and browsing platform of manga. In *The Outreach of Digital Libraries: A Globalized Resource Network*, pages 87–96. Springer, 2012.
[13] Clément Guérin, Christophe Rigaud, Antoine Mercier, Farid Ammar-Boudjelal, Karell Bertet, Alain Bouju, Jean-Christophe Burie, Georges Louis, Jean-Marc Ogier, and Arnaud Revel. ebdtheque: a representative database of comics. In *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013.
[14] Marvel. Downloads and Extras. http://marvel.com/mobile, Retrieved 2014/05/08.
[15] IDPF. EPUB 3 Overview, February 2014. http://www.idpf.org/epub/301/spec/epub-overview.html, Retrieved 2014/05/06.
[16] Matt Garrish. *What is EPUB 3*. O'Reilly Media, September 2011.
[17] Hajar Ghaem Sigarchian, Ben De Meester, Tom De Nies, Ruben Verborgh, Wesley De Neve, Erik Mannens, and Rik Van de Walle. EPUB 3 for integrated and customizable representation of a scientific publication and its associated resources. In *Proceedings of the 4th Workshop on Linked Science*, October 2014. URL http://ceur-ws.org/Vol-1282/lisc2014_submission_3.pdf.
[18] jQuery. http://jquery.com/, Retrieved 2014/11/12.
[19] Zoomooz.js. http://jaukia.github.io/zoomooz/, Retrieved 2014/11/12.
[20] IDPF. EPUB Region-Based Navigation, . http://www.idpf.org/epub/renditions/region-nav/epub-region-nav.html, Retrieved 2014/05/08.
[21] IDPF. EPUB Multiple-Rendition Publications, . http://www.idpf.org/epub/renditions/multiple/epub-multiple-renditions.html, Retrieved 2014/05/08.
[22] DCMI Metadata Terms. http://dublincore.org/documents/dcmi-terms/, Retrieved 2014/11/28.
[23] The DBpedia Ontology. http://wiki.dbpedia.org/Ontology, Retrieved 2014/11/28.
[24] Schema.org. omicvine.com/api/, Retrieved 2014/12/05.
[25] The FOAF Project. http://www.foaf-project.org/, Retrieved 2014/12/05.
[26] W3C. RDFa Core 1.1 - second edition. http://www.w3.org/TR/2013/REC-rdfa-core-20130822/, Retrieved 2014/05/08.
[27] Readium. http://readium.org/, Retrieved 2014/11/12.
[28] IDPF. http://www.idpf.org/, Retrieved 2014/11/12.
[29] Kobo Glo. http://www.kobo.com/koboglo, Retrieved 2014/11/12.
[30] Calibre. http://calibre-ebook.com/, Retrieved 2014/11/12.
[31] Ben De Meester, Tom De Nies, Hajar Ghaem Sigarchian, Miel Vander Sande, Jelle Van Campen, Bram Van Impe, Wesley De Neve, Erik Mannens, and Rik Van de Walle. A digital-first authoring environment for enriched e-books using epub 3. *Information Services and Use*, 34(3):259–268, 2014.