

# Digital Publishing Toolkit

## Contents

<b>01 Executive summary</b>	<b>5</b>
Electronic publishing requires deeper changes in your work than you might have expected . . . . .	5
Change is minor for scientific and mass publishers . . . . .	6
Change is major for small, arts-oriented publishers . . . . .	6
Industry promises versus reality . . . . .	6
What this Toolkit provides . . . . .	7
<b>02 Overview</b>	<b>7</b>
Why this toolkit . . . . .	7
Whom is this toolkit written for? . . . . .	8
What culture shock to expect . . . . .	8
How to use this toolkit (check chpt #s) . . . . .	8
Who worked on this toolkit . . . . .	8
<b>03 Introduction</b>	<b>9</b>
For newcomers in the field . . . . .	9
What is a text? . . . . .	9
What is an electronic text? . . . . .	12
What is an electronic publication . . . . .	13
Electronic possibilities . . . . .	13

<b>04 Genres of Publication</b>	<b>14</b>
1) Research publication . . . . .	14
2) Art/design catalogue . . . . .	15
3) Artist/designer book . . . . .	15
4) Art/design periodical . . . . .	16
5) New genres of publication . . . . .	16
Three levels of electronic publishing . . . . .	17
1) One-to-one publications . . . . .	17
2) One-to-many publications . . . . .	19
3) One-to-database . . . . .	19
<b>05 Strengths &amp; limitations of e-reading technologies</b>	<b>21</b>
Reader hardware . . . . .	22
Ereaders . . . . .	22
Tablets . . . . .	22
Smartphones . . . . .	24
PC/Laptops . . . . .	24
Reader software . . . . .	24
Mobile applications . . . . .	24
PC software . . . . .	25
Web platforms . . . . .	26
Browser applications . . . . .	26
File formats . . . . .	27
Reflowable documents (epub and AZW) . . . . .	27
Other formats . . . . .	28
Alternative ways of publishing . . . . .	29
<b>06 Distribution platforms</b>	<b>29</b>
e-book stores: Amazon, Apple, Google, Kobo, Sony... . . . .	29
Online reading platforms: Scribd, Issuu, website / blog . . . . .	30
Issuu . . . . .	30
Scribd . . . . .	31

Third party blogs . . . . .	32
<b>Revenue Model . . . . .</b>	<b>33</b>
Social reading platforms . . . . .	33
Print-on-demand . . . . .	34
Lulu.com . . . . .	34
Espresso Book Machine . . . . .	35
Global options . . . . .	36
Options in the Netherlands . . . . .	36
Pirate platforms . . . . .	36
Examples . . . . .	37
artist-/designer-run e-publishing projects . . . . .	38
Why change your life . . . . .	38
Towards a hybrid workflow . . . . .	40
One flowchart of all the three “flows”: Traditional, XML, Markdown	40
Desktop publishing workflow (from Word to InDesign to digital)	42
XML . . . . .	43
Markdown . . . . .	44
Database publishing / Content Management System . . . . .	48
<b>08 Guide: How to make a simple ePub . . . . .</b>	<b>48</b>
A simple e-book . . . . .	48
Direct conversion to ePub . . . . .	48
EPUB from scratch . . . . .	49
Layout of an epub package . . . . .	49
Creating your own epub . . . . .	50
<b>09 Guide: E-publications per genre . . . . .</b>	<b>54</b>
General questions . . . . .	54
Research publication . . . . .	54
General . . . . .	54
Decision making . . . . .	56
Instructions . . . . .	57

Specific issues for this publication type . . . . .	57
Based on the above, what to choose? . . . . .	57
Art/design catalogue . . . . .	58
General . . . . .	58
Artist/designer book . . . . .	58
General . . . . .	58
Art/design periodical . . . . .	61
General . . . . .	61
<b>10 Futurology</b>	<b>62</b>
Reading technology . . . . .	62
Reading culture . . . . .	62
Publishing culture . . . . .	63
NOTES / IDEAS . . . . .	63
11 Glossary of technical terms . . . . .	64
<b>A</b> . . . . .	64
<b>B</b> . . . . .	66
<b>C</b> . . . . .	67
<b>D</b> . . . . .	68
<b>E</b> . . . . .	69
<b>F</b> . . . . .	70
<b>G</b> . . . . .	71
<b>H</b> . . . . .	72
<b>I</b> . . . . .	72
<b>J</b> . . . . .	74
<b>K</b> . . . . .	74
<b>L</b> . . . . .	75
<b>M</b> . . . . .	76
<b>N</b> . . . . .	77
<b>O</b> . . . . .	77
<b>P</b> . . . . .	78

<b>Q</b> . . . . .	79
<b>R</b> . . . . .	80
<b>S</b> . . . . .	80
<b>T</b> . . . . .	81
<i>U</i> . . . . .	81
<i>V</i> . . . . .	82
<i>W</i> . . . . .	82
<b>X</b> . . . . .	83
<b>Y</b> . . . . .	83
<b>Z</b> . . . . .	83
 <b>12 Software Catalogue</b>	 <b>84</b>
Pre-existing software . . . . .	84
pandoc . . . . .	84
git . . . . .	84
github . . . . .	84
Self-written software . . . . .	84
 <b>Appendix</b>	 <b>84</b>
Keyword index . . . . .	84
Bibliography . . . . .	84
Link list . . . . .	84
Instructions used in the preparation of this document . . . . .	84
<a href="#">Instructions for placing images</a> . . . . .	84

## 01 Executive summary

### Electronic publishing requires deeper changes in your work than you might have expected

“You must change your life” - borrowing from philosopher Peter Sloterdijk, this could be the summary of our message for art and design oriented publishers, writers, editors and designers transitioning from traditional book making to electronic publishing or - more typically - hybrid print and electronic publishing.

Hybrid publishing will sooner or later confront them with the following: (a) rethinking of traditional publication formats (eg a catalog), (b) rethinking of editorial and production workflows and (c) rethinking of distribution.

### **Change is minor for scientific and mass publishers**

That said, there are exceptions. Workflow changes can be minor for publishers who already accomplish all their editorial work in highly structured digital document formats (like XML or databases); but typically, this is only the case in scientific and technology-oriented publishing. Changes can also be minor for larger publishers that can afford outsourcing. Making an electronic (digital) publication next to a printed publication then is simply a matter of paying an external service provider such as a document engineering company or a media design bureau for turning a Microsoft Word or InDesign file into an e-book [gl]. This process can be quick if the book is visually simple - such as a novel or a textbook with little illustrations - and economic if many e-books will be sold.

### **Change is major for small, arts-oriented publishers**

Since this Toolkit is for publishers who (a) publish visually oriented books in (b) mostly small editions (typically up to 500), it proposes an alternative route. Neither a complex internal IT infrastructure, nor costly outsourcing will be viable solutions for them. And, to disappoint a common expectation, there is no magic software button that will turn a print book design into an electronic publication. Since the two media are so different, with their own specific editorial and visual design needs, such a button won't materialize in the future either. Hybrid publishing will ultimately require editorial work to change, both technologically as well as editorial concepts.

For art and design publications, the challenge of going electronic is greater compared to other fields of publishing for a number of reasons: - visually oriented publications are still more difficult to technically realize in the electronic medium when designing for a multitude of different reading devices and e-book platforms; - small publishers are under even higher pressure to keep projects within low costs - but having to publish in one or more media will increase costs unless one accomplishes as much work as possible in medium-independent ways; - small editions make it necessary that electronic publications are sustainable: they should not need constant investment into technical maintenance and version updates.

### **Industry promises versus reality**

We face a major contrast between computer industry promises and the reality of the new medium. On the one hand, publishers, editors, designers and artists tend

to overestimate interactive and multimedia possibilities of electronic publishing. These new possibilities do exist, but in most cases cause higher development costs and remain specific to one particular technical reading platform.

On the other hand, publishers tend to underestimate how even technically simple and seemingly banal types of electronic publications can force a rethinking of traditional publishing formats (such as: the anthology, the catalogue, the periodical). Rethought formats may be the greatest opportunity offered by electronic publishing: for example, splitting up a traditional exhibition catalogue into interrelated micro-monographs that people can download and read as individual e-books.

## **What this Toolkit provides**

Going electronic - or going hybrid - means that you need to change the way you work in the whole process from manuscript to publication. The software tools currently in use, from word processors like Microsoft Word to desktop publishing suites like InDesign, were created for the analog or Desktop-Publishing world. Although it's possible to create electronic publications from those Microsoft Word and InDesign files that are likely the standard in your publishing workflow, it will be painful, slow, inefficient and expensive in most cases. This Toolkit describes technical alternatives because it is primarily aimed at small edition publishers who, in most cases, cannot afford to outsource e-book design to external service providers because this would double design costs.

This Toolkit describes easy-to-use tools and work methods for editing and designing hybrid print and electronic publications. These tools and methods do require some effort, but are for everyone: publishing houses, editors, writers, designers, artists and anyone who is curious.

This Toolkit has a particular (but not exclusive) focus on epub3 as an electronic publication format and on Markdown as a word processing format because of the specific needs of small edition publishers in the art and design field: low cost, ease of use, sustainable and platform-independence. These recommendations result from our practical experience in collaborating with four Dutch art, design and research publishers on electronic publication projects.

## **02 Overview**

### **Why this toolkit**

This toolkit aims to provide a practical guide on how to develop epub3s for publishers as well as for anyone else interested in this subject matter. It will give insight into the preconditions of electronic publishing, providing open-source

tools where possible, and allow publishers of the art and culture sector to navigate the diverse and complex landscape of digital publishing in a more informed way.

## **Whom is this toolkit written for?**

You may be a publisher, an experienced programmer, a designer or someone who is just starting out. You probably identify with or work in the art fields. You may be hoping to gain answers, knowledge, get tips, and consult various resources. This toolkit intends to provide them all. The overall focus is on pragmatic solutions for digital publishers within the art and cultural sector. No prior knowledge of creating an epub is required – with that said, a familiarity of various computer software is helpful, as is the willingness to learn and the inquisitive curiosity to look beyond this book for certain answers. This publication is not a tutorial on how to make an electronic publication, it just gives you some guidance on how to make a first attempt in creating an electronic publication.

## **What culture shock to expect**

As mentioned before “you must change your life”, so there might a little culture shock reading through this document. If electronic publishing will merely remain an afterthought in the production chain and product portfolio, the change might be light. If electronic publishing will become just as important as print, the change may be radical. What this means is that if you want to make visual electronic publications, you should be willing to change your current way of working and gradually get used to the offered but less common alternatives.

## **How to use this toolkit (check chpt #s)**

This toolkit will help you to develop epubs, specifically art and design publications, that overall make more use of different fonts, graphics and layout than plain text, for example novels. Chapter 3 will give a general introduction of these publications and sketches out various scenarios to develop epubs. Issues and opportunities of going electronic will be discussed, as well as strengths and limitations of particular reader hardwares and softwares, file formats and lastly distribution platforms. Chapters 4 offers a practical, how-to guidance for workflows (structured and per scenario) and designing epubs for the various scenarios addressed earlier in chapter 3. Chapter 5 illustrates the concepts with concrete examples from the Digital Publishing Toolkit program.

## **Who worked on this toolkit**

This publication is part of the Digital Publishing Toolkit research program, based on recommendations from practical experiences throughout the collaboration



with four Dutch art, design and research publishers. Thus, all disciplines were represented from publisher to developer to designer.

## 03 Introduction

### For newcomers in the field

Although there are no such things as stupid questions, there may be times when you're hesitant to ask questions in situations where people around you seem to know already. The function of this section is to solve your shyness. In this section we try to explain the fundamentals of electronic publishing in order to help the reader to formulate questions more precisely.

#### What is a text?

A text is a collection of words and words are compositions of letters. In order to read a text we have all kinds of layout helpers. Keep in mind that in the Roman times (when texts were cut in stone) we had continuing writing: no spaces between the words. This was not considered a problem as reading was a craft only a few people mastered. These people knew the words and hence were able to read aloud, just try: *Icanreadthiseasilyaloud*. In time, the craft of reading became a common good and many design helpers were introduced, such as spaces between words, capitals at the beginning of new sentences, commas, semicolons, colons and line breaks. Also the notion of paragraphs, chapters, etc., developed into a standardized culture that allowed a smooth transmission between the structure the authors endowed their text with and the readers who became familiar to these standards. Thus enabling the ease of reading and furthermore to read in silence.

This whole structure of stratifying words into sentences, sentences into paragraphs and so on, including reading aids such as exclamation marks, bold and under-scored text is called **markup**. It goes without saying that all these **markup** elements demand stable definitions and clear relationships. Everybody is free to invent their own rules (e.g., every first letter of a new chapter is a well-decorated small picture). For example, in the time of handwritten manuscripts many “free style” inventions were made. Some of them remained in our time and became part of the expended alphabet. Think of the ampersand “&”, it originated from the conflation of the letter e and the letter t – we call this conflation, *ligatures*. Hence, to be able to handle all vanities and define what we allow and what not, **Markup languages** emerged.

Markup languages are grammars that define the markup and the relations between markup elements. With the emergence of computer networks and the increasing need to standardize texts for multiple usages (think

about manuals), an international ISO standard established in 1982 is called: Standardised General Markup Language (SGML). This logically structured markup language was a big step forward, as it made a fundamental split between the text structure as such and the final representation of that structure. For example, contrary to languages used in word processors such as Tex, LaTeX, Word, Microsoft Word, Wordperfect or ODF (open office document format), where presentation and text structuring are mixed, SGML only defines functions or roles. When we type a **Bold** word in the text using a word processor we in fact type “start bold” -> type the word -> “end bold”.

## INTRO: MIXTURE LAYOUT

Markup languages are grammars that define the markup and the relations between elements. With the emergence of computer networks and the increasing need to store documents for multiple usages (think about manuals), an international ISO standard established, called: Standardised General Markup Language (SGML). This logically structured markup language was a big step forward, as it made a fundamental split between the text structure and the final representation of that structure. For example, contrary to languages used in word processors such as Tex, LaTeX, Word, Microsoft Word, Wordperfect or ODF (open office document) where presentation and text structuring are mixed, SGML only defines functions or actions. To type a Bold word in the text using a word processor we in fact type "start bold" -> "end bold". <!--Kimmy and Loes, perhaps this can be made clearer with an image--> What is happening here is a mixture of Layout and structure. Layout is the activity of presenting text onto a medium, such as a paper page. SGML and its derivatives, the easier HTML (HyperText Markup Language) and the expanded XML (Extensible Markup Language) make a strict distinction between structure and representation. A markup language knows notions such as "highlighted word or phrase" and then type 1,2,3...

What is happening here is a mixture of **Layout** and structure. Layout is the activity of presenting a text onto a medium, such as a paper page. SGML and its derivatives, the easier HTML (HyperText Markup Language) and the expanded XML (Extensible Markup Language) make a strict distinction between structure and representation. A markup language knows notions such as "highlighted

word or phrase” and then type 1,2,3... This allows you to equate e.g.: chapter heading with type1 and quotation with type 4. Depending on the output substrate you can then define in the layout phase how this will look like. For instance, a chapter heading is in a certain type font and font size and centred on the page, whilst a quotation is represented in the same font size and font of the running text, but now in italics. On a screen we can have things like chapter headings in pink and quotations in yellow. This freedom in the layout is then explicated in a so-called **style sheet**, which is a table that connects a layout schema with the markup schema. This way of working is imperative if we want to allow a source text to be represented in many different ways on various media of various size. Note, that in many programs these translations are done fairly invisibly to the user. If we translate a .odt file into a .docx file, in principle all coding is one-to-one translated. As we will see in the following, translation between one file type into another is not always symmetrical. Hence, the golden rule is to always make sure that the source text is as systemically structured as possible.

An important notion is that all text and accompanying coding are all written in simple letters and numerals, this is called **flat text**. In short, a stripped down text without layout — the most elementary token. It goes without saying that the flexibility of this process is limited to the character set for flat text we use. In the “old” days this was ASCII based and limited to the possible number of distinct signs (letter, numeral, comma, point, ..) of an 8 bit binary computer word. Slowly but firmly a new list of allowed signs is making inroads called: Unicode. It aims to include all alphabets and letter systems including common signs and ligatures. Again, this is a step forward to guarantee a clean source file for all kind of usages, even those we don’t consider or even imaging today.

### What is an electronic text?

An electronic text is normally understood as a text which is represented on a screen of some sort. Of course this is sloppy language. The key issue is that an electronic text became a well-structured file in which the emotions and intentions of the author are translated into notions like highlighted text of a certain type. Due to this markup, we become able to make different layouts, expressions, onto different media. It is of great importance to realise that electronic publishing introduces a big shift away from our page centred culture of book printing. In book printing we allow for various sizes depending on the original works of the author. For example, if we make an art book based on a collection of paintings or drawings, we can decide what the ideal book size is and whether or not it will be printed in oblong form. In the world of screens this becomes very difficult as we have very different sizes of screens, but no screen can be cut to the demanded size as with paper book publishing. In electronic books we have to work around things in a completely different way to the paper world. This means that the transposition from an existing work to an electronic representation is rife with difficulties if the structure of the texts and in particular if the relation between

illustrations and running text is important. In the world of novels we normally have only running text. Here the page size is less important and this is the reason why e-readers are becoming so popular. The text can always be made fitted to the size of the device, this is called **reflowable** text. In all other cases, we have to think deeper into how to design the work and under what conditions do we represent content and meaning. This will be discussed in the following chapters for various outlets, as we might opt for various versions of the original work. It goes without saying that the coming years, authors and designers will try and develop digitally born works that intrinsically allow for a variety of representations depending on the reading device whether it is electronic or not. Note also that the term digital just means that information is stored in unique entities. Only when we decide to deal with the smallest entity possible: a zero and a one, we speak about binary.

## **What is an electronic publication**

### **Electronic possibilities**

Novel electronic capabilities enable a great number of possible publication outlets. Before entering into the problematic aspects of such a plurality of presentations of the same message, we investigate the possibilities. As with all technological possibilities; the coin has two sides (leaving aside the unstable rim) which exclude each other. The printing press introduced pagination and indexing, allowing many thousands of identical texts to be read and compared by even more readers over the years, independently of location. In an electronic world with non-fixed screen sizes, this fact becomes problematic. On paper, text is fixed and this allows comparisons and interactions between different readers, separated in space and time. In an electronic version the fixity of the text remains, as the text file is independent from its final substrate (e-ink, lcd, paper), but its presentation onto the final substrate can vary substantially. Think about the introduction of numbering phrases in the Bible, which came along during a ceremony when the priest wanted everybody in the audience to read at once, however as the formats of the Bibles were non standardised, page numbers were of no help.

The challenge even becomes greater if we widen our ambition to pictures, audio and video, hyperlinks, etc.

In the Toolkit project, we deal with the field of books in the arts. This category of books integrates all aspects of text-only publications but expands it with visual information that can be explanatory of the text and -even more- importantly to visual information which is a 'stand alone' statement. In such cases, text, helps the 'reader' (viewer) to understand and have a deeper appreciation of the object. It is clear that various forms of art books demand and enable different electronic representations. In these books we witness a crossover between the primary importance of text and picture.

Nevertheless, as cultural standardization over the centuries rendered, for example, a detective, religious or an educational book instantly distinguishable by its typography and layout, - this too will also become a fact for electronic books. Apart from the electronic (or paper) book as a ‘object d’art’, electronic art book publishing will have many commonalities despite its difference in genre. A major issue is that the electronic sub-structure makes all files just bitstreams (streams of binary code: zeros and ones). The digital files containing the various kinds of information are all equal on this level of digital bitstreams in the computer memory: merely standardized code. The great new thing in the world of electronic art books is that based on standardized, though well tailored structures, the creative message can be published in a great variety of ways. This not only depends on the capabilities of the output (reading/viewing/listening) device, but also on the function of the book for the author, in a present context, such as a dictionary, a study, a reference, a coffee table book, or a leisure book. The same source can and will be represented differently under different circumstances. All these vistas demand a thorough and more labour-intensive editorial and production strategy. Not only because the same source can express itself in various output forms but even more so because once properly edited and stored electronically, information and its constituting parts can be reused and used in different ways, to be decided upon given a specific environment of goal, now and in the future.

## 04 Genres of Publication

In the present project we deal with various kinds of products which we combine into five genres:

- 1) Research publication;
- 2) Art/design catalogue;
- 3) Artist/designer book;
- 4) Art/design periodical;
- 5) New genres of publication.

### 1) Research publication

- Heavy in use of text
- Texts take a central role
- Visuals are secondary (images refer to texts)
- Often longer texts
- Often with a standardised structure

### Issues and advantages of going electronic:

One of the issues with going electronic with a research-like publication is setting parameters for the use of and interaction between text and images. Advantages of going electronic with this type of publication have to do with the wider reach and availability of the content. Making the text available digitally could increase its spreading. This is often a goal of research-like publications. Another advantage is that, due to the high level of standardised structure, a ‘modularisation’ of the content becomes possible thereby allowing to go beyond the traditional way of essay writing. In particular when authors refer to each other, they often share data, quotations, tables, figures, etc. Multiple use and reuse of ‘modules’ will enormously enhance the communication within a community.<sup>1</sup>

## 2) Art/design catalogue

- Heavy in use of images
- Visuals take central role over texts
- Visuals are primary (text refers to visuals)
- Descriptive texts of changing lengths

### Issues and advantages of going electronic:

One of the main issues which arise when trying to go electronic with Art/design catalogue-like publications lies in the use of images. The quality of the images is important. They must often be high in quality with as little distortion of colours as possible. An issue of concern is that often a fixed page layout determines the message. Changes in size might influence the flow of content. An advantage of going electronic with this type of publication could lie in the use of other (interactive) materials and easier distribution.

## 3) Artist/designer book

- Book *by* an artist, rather than *about* an artist
- Complex use of images and typography, often pushing the possibilities of the medium
- In print: making use of the book as a material, visual and haptic object;
- Difficult or impossible translation into other media;
- Text is often used in non-descriptive or non-narrative ways.

---

<sup>1</sup>lit ref: see for scientific article: de Waard, Anita; Kircz, Joost (2008) Modeling scientific discourse - shifting perspectives and persistent issues, ELPUB2008. Open Scholarship: Authority, Community, and Sustainability in the Age of Web 2.0 - Proceedings of the 12th International Conference on Electronic Publishing held in Toronto, Canada 25-27 June 2008 / Edited by: Leslie Chan and Susanna Mornati. ISBN 978-0-7727-6315-0, 2008, pp. 234-245. [http://elpub.scix.net/cgi-bin/works/Show?234\\_elpub2008](http://elpub.scix.net/cgi-bin/works/Show?234_elpub2008)

**Issues and advantages of going electronic:** An issue with making a digital publication out of an artist's/experimental book is the choice of material. The choice to go digital in this case is the most fundamental. Furthermore, here too, the quality of images and the relation between text and images is important. The choices which are being made by the creator influence the final outcome immensely. Advantages of going digital lie in the more extensive possibilities digitalization offers (e.g. interactive material etc.).

#### 4) Art/design periodical

- Heavy in use of both texts and images
- Both images and texts take central role
- Use of images is both primary and secondary
- Texts of changing lengths
- A more-or-less fixed page-layout format
- Recurring publication format (e.g. magazines, series etc.)

**Issues and advantages of going electronic:**

One of the issues with going digital with an art/design periodical lies in the limitations of the digital medium. The periodical thrives well by its physical presence. It can be tossed in a bag and just as easily be disposed of. In a digital form, the physical presence becomes limited. Another issue with going digital with an art/design periodical lies in the archival aspect. New articles can refer to older ones and even integrate parts. The periodical as such becomes a repository and therefore a playground for new work. An advantage of going electronic with this kind of periodical lies in the increased possibilities in extending content (interactivity, video etc.).

Another specific advantage of the periodical lies in the decreasing of distribution costs and the convenience with which a large audience can be reached.

#### 5) New genres of publication

By new types of publications we mean publications which cannot easily be placed in one of the genres mentioned above. These could be publications which fall outside of what we would normally consider a 'book'. Examples could be an app or a series of tweets etc. Setting a predisposed set of parameters for this genre would be difficult. Similar to adventure games, here we can imagine the use of an interactive approach where the reader is able to control the narrative or to build or customize the publication. For example one can think of platforms used to collect tweets in book form or **tumblogs** in an epub as shown during the Off the Press conference.



## Three levels of electronic publishing

The goal of the project is to define a common ground for directions, strategies, and tools that will enhance the transition from paper-only publications to a mixture of paper & electronic publication, given the above listed genres of publications. As the mindset of the partners in the projects are very diverse and in most companies the discussion is only in an early stage, it does not make sense to produce a matrix of all questions and all answers. A better approach is to try and sketch the overall picture, based on commonalities and diversions. This attempt is presented below.

Independently of the different genres listed above, we can stratify from simple conversions from paper publications to electronic ones to full-scale electronic publications which allow one possible representation of its content as a print-on-demand (POD) traditional paper book.

A division into three levels can be made:

1. One-to-one; when a book is considered as a separate product where text-authors, illustrators, artist photographers, and designers work together to produce the book as a one-off team.
2. One-to-many; this publication is not a single object, but has various chameleonic appearances, dependent from the context and available presentation media (type & size of screen/paper).
3. One-to-database; here the various independent components are defined as modules that can be used and re-used multiple times by whomever has access to them.

Below we expand a little more on this and compare it with the mentioned genres:

### 1) One-to-one publications

On the first level, we have the unique book. Here we consider each book as a separate product where authors, illustrators/artists/photographers, and designers work together to produce the book as a one-off team. As there are many of such unique books, the production workflow is traditional. Pictures and/or full layout pages are not yet considered as reusable digital objects. In many cases no archives of the independent ingredients that together create the end-product are available. Take note that there are many common characteristics between books in one genre. Hence, the question is how we can reveal and define these characteristics to enhance insight in the nature of this type of book. Artist's/experimental books are an obvious example of this kind of book.

In light of this, the issue at hand is experimenting with an e-representation of the same book. In some cases files of the final pages/book are being converted

## THREE LEVELS OF ELECTRONIC PUBLISHING

A division into three levels can be made:

One-to-one; when a book is considered as a separate product where text-authors, illustrators, artist photographers, and designers work together to produce the book as a one-off.

One-to-many; this publication is not a single object, but has various chameleonic forms, dependent from the context and available presentation media (type & size of screen).

One-to-database; here the various independent components are defined as modules, which can be used and re-used multiple times by whomever has access to them.

Figure 1: Three Levels of Electronic Publishing

into epub3 by an external agent. The final pages are checked on readability and clearness (for example, that figures and captions are on the same page) and changes are made accordingly to create the e-book. On this level the e-book is not more than an exact-as-possible picture of the printed book. In such cases the publisher might use a document management system, which can be a mere system of folders/directories. Archiving files is important for possible reprints. Often publishing houses are forced to scan their own book in order to reprint them because they don't have an archived version of it. Moreover having an archive of the semantically structured contents allows reissuing the publication in ways that were not envisioned before. Workflows are to a large extent unique though follow traditional publication path ways.

## **2) One-to-many publications**

On the next level we see a split between the manifestation of a paper print version and the electronic version. In fact, the electronic version, contrary to the paper version, is not a single object but has various appearances. This is due to the fact that the presentation substrate of an electronic book is not unique. Furthermore, the sizes of reading devices are not standardised and generally allow for both portrait or oblong viewing. On top of this, the popular semi-standard epub3 allows for flowable pages and allows the reader to change the font and fontsize (corps) to avoid the need of reading glasses for a more pleasant consumer experience. The best example of the one-to-many publication are the art/design catalogue-like publications and research-like publications. In this category workflow schemes become very important as the same texts and images are used in various outlets. A workflow and related descriptors (metadata) set is compulsory. Future readers must be able to make choice from the collection of available essays. The most stable approach, but also one the most complex ones, might be a XML based database.

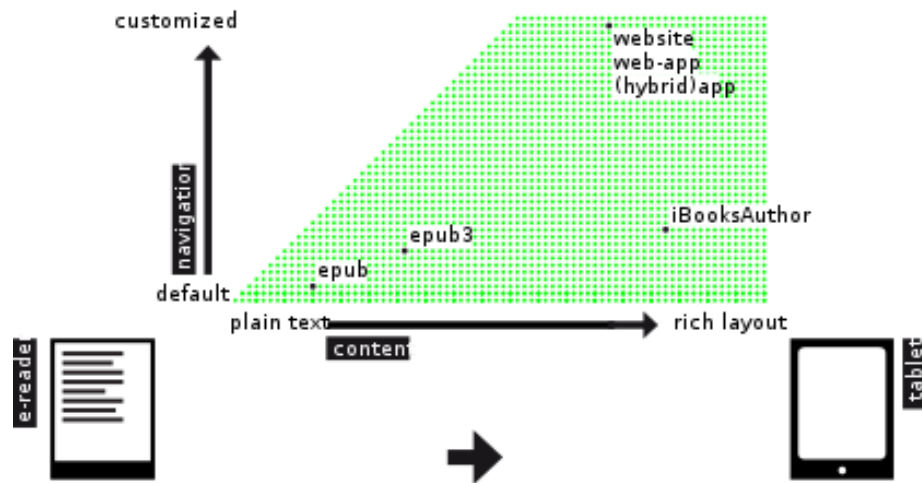
## **3) One-to-database**

The next level is Database Publishing. Database publishing means that all objects or entities can be used independently from each other. This means that they can be used in multiple ways, can be easily reused, are uniquely defined and stored in a database. As a precondition for full scale database publishing, the editorial workflow is important, because it is there that a decision is being made on what items are defined as individual entities and which characteristics and features (metadata) they have. In the case of a collection of essays, the page-layout is of less importance compared to full colour art books; the accidental picture or graph can be easily accommodated. A series of essays (and blogposts) mimics a journal publication with more-or-less fixed lay out. After all, a new volume in a yearly series is only special the year of publication. Later on the essays of that volume are just one item of the collection (though with metadata

indicating the year of publication and related details). However, in all instances the workflow demands for this type of projects might serve as an example for others towards a full database approach. This means that the workflow scheme must have such a coherent structure that choices are explicated. It is important to note that there will never be a one workflow which fits all but if the relevant metadata is available; publications in different forms can be created.

## 05 Strengths & limitations of e-reading technologies

### DIGITAL PUBLISHING RANGE



## Reader hardware

There are several ways to peruse electronic publications. Portable devices like e-readers or tablets are by far the most popular gateway to digital content. Smartphones also offer some of the capabilities of an e-reader, as most of these devices are a miniature version of their tablet counterparts. Lastly not to forget the desktop computer, often used in combination with the physical book, especially in the case of research publications.

### Ereaders

Ereaders became widely known with the introduction of the Amazon Kindle in 2007. Ereaders differ from tablets in the way that they are only suitable for reading books, are relatively cheap and have monochrome displays. An important characteristic of the e-reader is the use of *electronic paper* (e-paper), a so-called electrophoretic screen. Electronic paper is designed to mimic the characteristics of paper as closely as possible and is relatively low in energy consumption. The current generation of e-paper is incapable of rendering complex motion, so animations and videos are impossible to use.

Both Amazon and Barnes & Noble also offer more expensive models, Kindle Fire (and Kindle Fire HD) and NOOK Tablet (and NOOK HD+) respectively. These models are not really e-readers anymore, but fully-fledged, small sized tablets. These models do not use *e-paper*, but instead work with colour LED screens with backlight – a standard component of tablets such as iPad and Galaxy Tab. Often these tablets use a version of the Android operating system as its core, in contrast to the classic monochromatic e-readers using their own operating system built on a version of Linux. The Samsung Galaxy Note and Apple iPad mini can be seen as an answer to the smaller sized tablets.

Because of the lack of a colour screen the current generation of e-readers is best suited for text-based publications, i.e. novels, etc.

### Tablets

The tablet-market is dominated by two main players,<sup>2</sup> Apple and Samsung. Apple utilises their own operating system (iOS) for the iPad, while Samsung with its Galaxy Tab product line has opted for Android. The Android-segment of the market consists of many manufacturers offering similar hardware with a variant of Android as its operating system. Several e-reader applications exist for both iOS and Android, the most important ones will be reviewed in the Reader Software section.

The most important advantage of tablets over traditional e-readers is the fact that the latter do not support colour. The upside of the tablet's colour screen is

---

<sup>2</sup><http://www.gartner.com/newsroom/id/2674215>

## Strengths & Limitations

<http://digitalpublishingtoolkit.org/2013/06/digital-publications-in-practice/>

- devices
- e-paper
- led screens
- iOS/apple
- Andriod
- Nokia/Microsoft
- lezen in de zon...
- battery life
- grootte van het scherm oppervlak

Figure 2: Strengths & Limitations

at the same time also its weakness, as backlit LED screens are generally known to cause eye fatigue. Furthermore, reading in the sun is problematic as well, because direct sunlight obscures most of the visibility of the screen. Battery life is another issue with tablets. Though newer tablet models often claim impressive battery life, e-readers still reign supreme with an average battery life of up to several weeks.

## **Smartphones**

The story for tablets also largely applies to the realm of smartphones.<sup>3</sup> Of course most smartphones have smaller screens than both e-readers and tablets. But the similarities between smartphones and tablets are large enough to justify clustering them with tablet devices.

## **PC/Laptops**

Reading e-books on a laptop or personal computer is a possibility but may be a less natural fit than handheld devices. A full range of software for e-reading is available, but the same downsides of relatively limited battery life and increased chances of eye fatigue exist as with tablets.

## **Reader software**

Standalone applications for desktops, tablets and smartphones offer some advantages to the software present on e-readers. Firstly almost all devices offer a full colour display, as opposed to the monochrome e-ink screens of e-readers. Secondly, a lot of applications support a larger subset of the epub standard, which allows for more features - like custom fonts, layout styling, etc.

## **Mobile applications**

There are many existing applications for mobile devices (tablets and smartphones). Some of the available software constitutes of companion apps to well-known e-readers, like Amazon's Kindle and Kobo's range of reader hardware. The other category contains applications which are not for available e-readers but often offer their own storefront, viz. iBooks and Aldiko. epub 2.0 and 3.0 support is generally available, but some of the applications require the user to convert files into a proprietary file format in order for the e-reader to be able to read the document. Kindle is an example, as it only supports Amazon's proprietary file formats (AZW, KF8, mobi and txt) and PDF (Portable Document Format).

---

<sup>3</sup><http://www.gartner.com/newsroom/id/2665715>



**iBooks** iBooks is Apple's e-reader application for iPhone, iPad and Mac. Books may be bought in the iTunes Store or can be copied from a desktop to the library of iBooks. The latest version of iBooks offers support for many of the features of the epub 3 standard, but also PDF and the closed-source `.ibooks` file format (IBA) - a derivative of epub 3.

**Marvin** Marvin is another application and one of the few paid entries. It does not offer its own store and is mainly geared towards managing existing collections of digital publications compiled from various sources. Marvin only accepts epubs and offers to convert incompatible file formats using third party software package Calibre.

**Aldiko** Aldiko is both a storefront and an e-reader for Android which supports epub and PDF. As such it is comparable to Apple's iBooks. There is a paid version which offers some extra features (annotation, removal of advertisements) which the free version does not offer.

**Kindle** Kindle is Amazon's tablet and smartphone counterpart to their physical e-readers (Kindle Paperwhite, etc.). The application is available for both iOS and Android. Books can be bought from inside the application on Android. On iOS it's less straightforward due to the restrictions Apple places on in-app purchasing. Kindle for mobile devices only has support for Amazon's proprietary file formats (AZW, KF8) and PDF. Epubs have to be converted using software like for example Calibre, in order for Kindle to be able to import the files into its library. The application is able to sync its library so that a collection of e-books is available on both the Kindle e-reader and in the application. It is not possible to copy-paste.

**Kobo** Kobo, like Amazon, produces e-reader hardware and also has a large bookstore. As such the apps they offer for Android and iOS are largely complementary to the e-reader companion devices manufactured by Kobo. Libraries are synced across devices, provided all the content is bought in Kobo's bookstore. Kobo supports both epub and PDF files.

## PC software

A plethora of desktop applications exist that allow users to read epubs. Of the five applications mentioned above only Aldiko is not available on desktops and iBooks is a Mac-only application. Kobo and Kindle both offer Windows and Mac versions of their software. The functionality is similar to that of the mobile versions.

Calibre is an application of note here, because it's an e-book management suite with many features. It offers tools for managing large collections of e-books, but also converts files to many different formats (e-books and other text based formats)<sup>4</sup>. Viewing all major e-books file formats, as well as editing epubs and AZWs is also part of the software package.

**Calibre** Calibre is an extensible application that uses plugins. It is an e-book management suite with many features. It offers tools for managing large collections of e-books, but also converts files to many different formats (both e-book and other text based formats)<sup>5</sup>. Viewing all major e-book file formats, as well as editing epubs and AZWs is also part of the software package.

**Adobe Digital Editions** Adobe Digital Editions (ADE) is an e-book reader and management tool which is able to read epub and PDF. Support for epub3 is incomplete though, though Adobe claims support for all the important features is present - without elaborating what those important features are. ADE also integrates with many e-readers offering syncing possibilities. So a library managed with ADE could in theory be synced with a compatible e-reader.<sup>6</sup>

## Web platforms

There are some web platforms for reading epubs online, their popularity is hard to gauge. A project called [Bookworm](#), developed by Threepress, was closed by O'Reilly Labs in 2012. The reason given for the closure mentions an "interesting experiment" but also due to the dramatic changes in the e-book ecosystem over the past few years. [Booki.sh](#), another project is still online and offers a complete library of e-books which may be view and stored online. It doesn't appear to operate commercially though, the ability to purchase books was removed in June 2013.<sup>7</sup>

## Browser applications

Apart from web applications, several browser extensions exist allowing users to read epubs in their web browser. These extensions are most likely more of a convenience method in order to quickly (pre)view e-books, instead of being fully-fledged solutions for reading e-books comparable to the functionality of e-readers.

---

<sup>4</sup><http://manual.calibre-e-book.com/faq.html#what-formats-does-app-support-conversion-to-from>

<sup>5</sup><http://manual.calibre-e-book.com/faq.html#what-formats-does-app-support-conversion-to-from>

<sup>6</sup><http://blogs.adobe.com/digitalpublishing/supported-devices>

<sup>7</sup><http://blog.booki.sh/blog/post/e-book-sales-via-booki-sh-to-end-on-june-30/>

**Radium** Radium is a project by several publishers and technology companies aiming to provide a reference system, a collection of best practices, for rendering epub 3 publications.<sup>8</sup> The Radium Project offers a range of tools for online and offline use, mainly geared towards software developers. One of the tools is a browser extension<sup>9</sup> for the Google Browser. After installing the extension offers an epub reader inside the browser window.

**epubReader** epubReader is similar to the browser extension offered by Radium, the main difference is the supported browser, as epubReader is only compatible with Mozilla Firefox.

## File formats

### Reflowable documents (epub and AZW)

The history of the modern e-book can be traced back to the late 1990s with the establishment of the *Open eBook Forum* tasked with the creation of the *Open eBook Publication Structure* (OEBPS). Microsoft was heavily involved in the development of what became the *Open eBook* based partly on technology created by a company called SoftBook Press.<sup>10</sup> SoftBook Press developed the format, based on XML and XHTML, as a companion to their e-reader. The *Open eBook* specification had its first release in 1999 and was later renamed to epub when the standard reached the 2.0 version milestone under the auspices of the *International Digital Publishing Forum* (IDPF, formerly the *Open eBook Forum*).<sup>11</sup> The epub 2.0 standard has support for basic styling, custom fonts, etc.<sup>12</sup>

At the beginning of Amazon's foray into the world of digital publishing, the online bookseller chose not to use epub but opted to buy a French company responsible for the development of *Mobipocket* and the MOBI file format. One of the main reasons for this move was the widely supported Digital Rights Management (DRM) engineered by *Mobipocket*.<sup>13</sup> Technically, the MOBI file-format is partly based on *PalmDOC*, an ebook file-format readable on PDAs running the OS developed by Palm Inc., with added support for the guidelines taken from the *Open eBook* standard.<sup>14</sup> Mobilepocket and Amazon then developed the format

---

<sup>8</sup><http://readium.org/readium-project-goals>

<sup>9</sup><https://chrome.google.com/webstore/detail/readium/fepbnnkkadjhjahcafoaglimekefifl>

<sup>10</sup><http://www.businessweek.com/1998/46/b3604010.htm>

<sup>11</sup>Bläsi, C., Rothlauf, F., 'On the Interoperability of eBook Formats', Johannes Gutenberg-Universität Mainz, 2013, p. 12, <http://wi.bwl.uni-mainz.de/publikationen/InteroperabilityReportGutenbergfinal07052013.pdf>

<sup>12</sup>idpf, 'Open Publication Structure (OPS) 2.0.1 v1.0.1, Recommended Specification September 4, 2010', 2010, [http://www.idpf.org/epub/20/spec/OPS\\_2.0\\_latest.htm](http://www.idpf.org/epub/20/spec/OPS_2.0_latest.htm)

<sup>13</sup>Rosenblatt, B., 'Amazon.com Acquires Mobipocket', DRMWatch.com, 2005, <https://web.archive.org/web/20050426003307/http://www.drmwatch.com/drmtech/article.php/3499386>

<sup>14</sup><http://wiki.mobileread.com/wiki/PalmDOC>

further in order for it to be used exclusively with *Mobipocket* and, later, Amazon *Kindle* e-readers. The MOBI format is now called AZW by Amazon, their generic term for all e-books released by the company.<sup>15</sup>

Nowadays, roughly two generations of widely used e-book formats exist. One generation is based on the epub 2.0 standard as introduced in 2007, or an earlier version released under the *Open eBook* moniker. Amazon's MOBI/AZW is an example of this, as it's still widely used and supported by the *Kindle* e-readers. The other generally implements features from the epub 3.0 standard, while often remaining backwardly compatible with older versions and subsets of epub 2.0 and even 1.0 to a certain extent. epub 3.0, released in 2011<sup>16</sup>, brought support for HTML 5 (as opposed to XHTML 1.1 in epub 2.0), more advanced styling using CSS3, scripting (discouraged in epub 2.0) and easy embedding of video and audio, amongst others.<sup>17</sup> epub 3.0 forms the basis of most of the e-book file formats available today. Most of the current file formats, like Amazon's AZW3/KF8 and Apple's iBooks, implement most of the epub standard, mainly to benefit from the HTML5 and CSS3 specifications, while adding their own proprietary extensions.<sup>18 19</sup> These extensions are mainly geared towards extra support for more advanced (fixed) layout options, rich media integration and DRM.<sup>20 21 22</sup> An AZW3/KF8 e-book is basically an epub 3.0 wrapped in Amazon's DRM. For backwards compatibility a MOBI version of the publication is generally also present in the e-book package.<sup>23</sup>

Of course epub 3.0 files without proprietary extensions exist as well and are sold to various commercial channels. In addition to their own iBooks file format, Apple also sells epubs in their online store.<sup>24</sup> Other major players include Kobo, Google Play and Barnes & Nobles' NookPress.<sup>25 26 27</sup>

## Other formats

The aforementioned file formats may be categorised as reflowable documents, although there is support for fixed layouts. Of course there are different ways to disseminate electronic publications, one way is exporting a document as a PDF,

<sup>15</sup>[http://wiki.mobileread.com/wiki/AZW#Internal\\_Formats](http://wiki.mobileread.com/wiki/AZW#Internal_Formats)

<sup>16</sup>idpf, 'epub 3 Overview, Recommended Specification 11 October 2011', 2011, <http://www.idpf.org/epub/30/spec/epub30-overview.html>

<sup>17</sup>idpf, 'epub 3 Changes from epub 2.0.1', 2011, <http://www.idpf.org/epub/30/spec/epub30-changes.html#sec-new-changed-xhtml5>

<sup>18</sup><http://www.amazon.com/gp/feature.html?ie=UTF8&docId=1000729511>

<sup>19</sup><http://www.apple.com/ibooks-author/>

<sup>20</sup>Castro, L., 'Expandable Photo Blocks in iBooks on iPad', Pigs, Gourds and Wikis, <http://www.pigsgourdsandwikis.com/2011/01/expandible-photo-blocks-in-ibooks-on.html>

<sup>21</sup>'Create Multi-Column Article for the iPad', <http://padilicious.com/multicolumn/index.html>

<sup>22</sup><http://www.amazon.com/gp/feature.html?ie=UTF8&docId=1000729511>

<sup>23</sup><http://wiki.mobileread.com/wiki/KF8#Overview>

<sup>24</sup><http://www.apple.com/itunes/working-itunes/sell-content/books/>

<sup>25</sup><http://nl.kobo.com/writinglife>

<sup>26</sup><https://play.google.com/books/publish/signup#settings>

<sup>27</sup><https://www.nookpress.com/support/faq>

but there are also more content specific file formats like the *Comic Book Archive*, support for these file formats by e-readers varies greatly.

### Alternative ways of publishing

A single downloadable package like epub might be a great vehicle to monetise, but there could be other factors to consider when publishing. A consistent design and uniform interactivity across devices could be a requirement, in which case a reflowable document might not be the ideal solution. Some publishers opt to develop their own (mobile) applications, like The Guardian's iOS version of their newspaper, the amplified eBooks by Penguin or the children's book by Purple Carrot.<sup>28 29 30</sup> These solutions offer fine-grained control over user interaction and consistency of design, but come with the extra cost of hiring a development team to engineer the application. Also, publishing for different platforms is not that straightforward. Code written specifically with the iOS SDK (Software Development Kit) in mind will likely require a lot of editing before it's ready to run on the Android platform. Portability is another issue, it's fairly easy to transfer an epub to another device, however application binaries are not that easily passed around, mostly due to the relatively closed nature of mobile operating systems.

Lastly, why not just publish essays, articles or even whole books on a website? The wealth of weblogs and other publishing platforms shows that this is a viable form of publishing. Monetising is less straightforward, most websites generate income by showing advertisements or sponsored articles. Paid membership is also used as a business model by some websites, like *De Correspondent*.<sup>31</sup> One of the major downsides is that content will only be available online and cannot be easily passed around as a single unit of information like an epub or a PDF.

## 06 Distribution platforms

### e-book stores: Amazon, Apple, Google, Kobo, Sony...

#### Strengths Weaknesses

---

<sup>28</sup><http://www.theguardian.com/global/ng-interactive/2014/may/29/-sp-the-guardian-app-for-ios-and-android>

<sup>29</sup>For example: [http://www.penguin.com/static/pages/features/amplified\\_editions/on\\_the\\_road.php](http://www.penguin.com/static/pages/features/amplified_editions/on_the_road.php) and [http://www.penguin.com/static/pages/features/amplified\\_editions/atlas\\_shrugged.php](http://www.penguin.com/static/pages/features/amplified_editions/atlas_shrugged.php)

<sup>30</sup><https://itunes.apple.com/us/app/the-prisoner-of-carrot-castle/id499981407?mt=8&ign-mpt=uo%3D4>

<sup>31</sup><https://decorrespondent.nl>

## Online reading platforms: Scribd, Issuu, website / blog

Online reading platforms are browser-based platforms where digital publications are uploaded and made available. The publication can be presented in different formats. The choice for an online platform and the accompanying publication format also has effect on modes and reach of distribution.

When using an online reading platform, publishers effectively become dependent on the platform provider. If the platform changes its technology, terms of service or goes out of business, the publisher will be out of control. Online reading platforms typically create a higher degree of dependence and vendor-lock-in than classical e-bookstores that distribute downloadable files (and where it is easier for a publisher to make those files on an alternative download/shop platform).

Below several platforms are discussed, including characteristics, statistics, strengths and weaknesses, and how-to-publish.

### Issuu

#### Characteristics

Issuu can be found at <http://www.issuu.com>.

- It offers many options to publish and distribute works: in-screen reading with flippable pages, downloadable files, embedding of the readable format on your own website
- It is used for books, brochures, magazines, etc
- Allows downloading of books to read online
- Because it is possible to upload a designed PDF it is a good way to publish styled and colourful publications
- Mostly used by independent publishers and individual users
- Supports all devices
- Mobile and desktop
- Full embedding possibilities
- Supports most file formats

#### Statistics

*For Issuu* (June 2014) 80 million monthly readers  
15 million publications

*For individual users*

Issuu offers general statistics for users and their publications  
For deeper statistics a paid account is needed

#### Strengths and weaknesses

*Strengths*

Embed on all platforms  
Sharing possibilities  
Professional look and feel  
Audio support

#### *Weaknesses*

Limit on publications of 500 pages  
No selling possibilities, only link to shop

#### **Revenue Model**

Freemium model <sup>32</sup>

#### **How-to-publish**

Create an account  
Easy upload of many different file formats  
Embed on your own website  
Offer downloadable file

### **Scribd**

#### **Characteristics**

Scribd can be found at <http://www.scribd.com/>

- The world's largest collection of e-books and written works, according to their own sources
- Compatible with iPhone, iPad, Android, Kindle Fire, and any web browser
- Allows downloading of books to read offline
- Supported File Types: pdf, txt, ps, rtf, epub, key, odt, odp, ods, odg, odf, sxw, sxc, sxi, sxd, doc, ppt, pps, xls, docx, pptx, ppsx, xlsx
- Lots of free content
- Subscription model
- No limits on genre
- Both works from established publishers and self-published works

#### **Statistics**

*For Scribd* (June 2014)  
80 million monthly readers  
40 million books and documents  
100 countries

#### *For individual users*

Scribd offers general statistics for users and their publications

#### **Strengths and weaknesses**

##### *Strengths*

---

<sup>32</sup>Ulises Carrión, "The new art of making books", Aegean editions, 2001

Sharing possibilities

Selling possibilities

#### *Weaknesses*

Few options for metadata

In-browser not optimal (like Issuu for example)

Embedding not optimal

#### *Revenue Model*

Scribd offers a subscription model for its readers and allows publishers to set a price for their work

#### *How-to-publish*

Create an account

Easy upload of many different file formats

Offer in-browser reading and downloadable file

### **Third party blogs**

**Medium** <https://medium.com>

#### **Characteristics**

External party

Text-based

Not for publishing whole books, but for example a summary or chapter Strong connection with eg Twitter

#### **Statistics**

Medium offers stats for individual users, including referrers

#### **Strengths and weaknesses**

##### *Strengths*

New audiences and communities

Easy-to-use

Designed specially for reading

Responsive website

##### *Weaknesses*

Third party website

Suitable for shorter texts only

Login with Twitter or Facebook only

#### **Revenue Model**

Medium has an editorial staff, getting paid per click. There is no revenue model for individual, external users

#### **How-to-publish**

Create an account and upload text for blog

**WordPress.com** [<https://wordpress.com/>] (<https://wordpress.com/>)



**Characteristics**

External party

Text-based

Not for publishing whole books, but for example personal blogs

**Statistics**

Medium offers stats for individual users, including referrers

**Strengths and weaknesses***Strengths*

Open Source

Free Easy-to-use

Designed specially for reading

Also for mobile

*Weaknesses*

Third party website

Suitable for shorter texts only

**Revenue Model****How-to-publish**

Create an account and upload text for blog

**Social reading platforms**

Social reading has to do with the experience of reading e-books. It is a field in great development, which lies beyond the scope of this Toolkit. However, it is good for publishers working on e-publications to keep an eye on the future of social reading.

Examples of social reading platforms:

- *Goodreads* ('Goodreads is the world's largest site for readers and book recommendations. Our mission is to help people find and share books they love. Goodreads launched in January 2007.')
- *Social Book* ('Social Book, created by the Institute for the Future of the Book, is a social reading platform that allows reader to add their own commentary to texts, share these ideas with others, follow others' comments, and create communities of interactive reader/writers.')
- *Wattpad* ('Wattpad stories are free. Whether you're online or off, use the devices you already own to carry an entire library wherever you go.' 'Join the conversation about the stories you read: message the writer and interact with other people who love the story as much as you.')

- *Hebban* (A Dutch social reading platform that released in beta version in 2014)
- *Social media discussions* are also part of the social reading experience. One can think here of Twitter interviews or when a book is being discussed online by a publisher.

## Print-on-demand

There are a lot of options for print-on-demand publishing. A simple Google query will return a myriad of services. Below are discussed the most important ones, international and Dutch. Of course, ‘regular’ printers also offer print-on-demand services. However, they usually ask for a minimum of copies ordered.

### Lulu.com

Lulu is the biggest print-on-demand service that offers publishing free of costs, based on a model of shared profit. <https://lulu.com>

#### Characteristics

- Available in six languages: English, French, Spanish, German, Italian and Dutch
- One of the main independent actors
- Used in 225 countries and territories
- 1.8 million publications
- 20,000 new publications each month
- 1.1 million authors
- Mostly for self-publishing
- Mostly for publishing books
- Lulu also offers e-book publishing service

#### Strengths and weaknesses

##### *Strengths*

Ability to set your own price

Distribution through Amazon and iBookstore (eBooks only) is possible Free ISBN

##### *Weaknesses*

Prices can rise quickly per publication, especially because of postal services

Minimum and maximum on number of pages

Not a very user friendly interface and help section

#### Revenue Model

Authors/publishers get 80% of the profit on a publication, Lulu gets 20%

**How-to-print**

Create an account

You will need two PDFs: a single page PDF for the inside and front, back, and spine in one PDF for the outside

Choose your format, paper, etc. It's very important to have precise formats for the PDFs, otherwise Lulu can't make a printable book You can add your own ISBN or have Lulu assign an ISBN to the publication One can use a browser based editor to design a cover Upload the PDF's and publish your work Choose your preferred ways of distribution

**Espresso Book Machine**

The [Espresso Book Machine](#) can be found on different locations, for example in a physical bookstore. It allows to print a (digital) book on the spot in a direct-to-consumer model.

**Characteristics**

Input is PDF Print as many copy as needed

A book can be stored in a database so other customers on other locations can buy a copy

Print copyright-free books from Google

Limitations in format, paper, and color Price is almost equivalent to a general book

**Strengths and weaknesses***Strengths*

The book is printed while you wait for it

Green technology

*Weaknesses*

The Espresso Book Machine (EBM) is tied to location and for now mostly available across the United States, two locations in the Netherlands (American Book Store in Amsterdam and The Hague), and scattered worldwide locations Options vary per location

Limitations in paper, color, size

**Revenue Model**

The EBM location gets a small consignment fee for each printed book, self-publishers can set their price according to their wishes

**How-to-print**

You will need two PDFs: a single page PDF for the inside and front, back, and spine in one PDF for the outside

Sign the affidavit stating you own the rights to the book

Choose your preferred ways of distribution

## Global options

*Kobo Writing Life* (<http://www.kobobooks.com/kobowritinglife>) lets authors and publishers self-publish digital content in 160+ countries. One can use Kobo Writing Life to publish eBooks and track sales.

*PubIt!* (<http://pubit.barnesandnoble.com/>) automatically converts your digital files for viewing on NOOK, mobile, and computing devices. It helps you distribute your ebooks to all kind of readers.

*Smashwords* ([http://www.smashwords.com/about/how\\_to\\_publish\\_on\\_smashwords](http://www.smashwords.com/about/how_to_publish_on_smashwords)) makes it free and easy to publish, distribute and sell ebooks globally at the largest ebook retailers, including the Apple iPad iBookstore, Barnes & Noble, Sony, Kobo, Baker & Taylor, Diesel eBook Store and more. There are no setup fees and no cost to update or revise your book.

*XinXii* (<http://www.xinxii.com/>) facilitate authors to upload and sell their work online on their XinXii author page: short works, documents and books (as eBook or audiobook) - in multiple formats including PDF, ePub and mobi. As an aggregator, XinXii distributes to major international eBook retailers.

## Options in the Netherlands

*CB Print on demand* (<http://www.cb-logistics.nl/markten/media/uitgeverijen/logistieke-diensten/print-on-demand/>) is the largest Dutch supplier of non-specialist printed books to bookshops and consumers. In addition to selling e-books through Dutch and Belgian retailers, CB also facilitates international sales. One of the services is print-on-demand. This is directed primarily at publishers already working with the CB distribution center. The book will stay available through the book databases used by book stores and publishers in the same way when it is in stock as a paper edition.

*Boekscout* (<http://www.boekscout.nl/>) is one of the biggest print-on-demand publishers in The Netherlands, aimed at self-publishing.

## Pirate platforms

As with music and films, ‘pirate’ platforms haven often been the avant-garde of electronic publishing. They provide music, films, games and electronic books as free downloads - either in outright copyright violation or in gray zones (when, for example, providing obscure and out-of-print books, films or recordings whose rights owners are unknown). Whether one likes it or not, pirate platforms have best met customer demand. Often, they are user friendlier than many Internet shops. The mere existence and popularity of pirate sites for digitized book shows, by the way, how large the market for electronic reading really is.

Commercial providers of digital content can learn a lot from the pirates. The best ones are already doing that. [Netflix](#), the most successful web service for streaming films and growing competitor for pay tv, is known for buying films based on their popularity on [The Pirate Bay](#).

## Examples

Pirate platforms are, by their nature, even more volatile than online bookstores and e-reading platforms. They get easily cracked down, and they can easily reshape under new names or modified concepts.

**All-purpose download sites** Sites like [The Pirate Bay](#) offer all kinds of media for download: films, music, computer games and software, and electronic books. Since they are widely used, display current download statistics and since their technology (bittorrent ) depend on user participation for their downloads to stay alive, they can provide great insight into what is really popular among readers.

The content listed on The Pirate Bay and other sites for the bittorrent protocol tends to be mainstream: Hollywood films, pop music, and a tendency towards non-fiction and technical handbook literature alongside fantasy and Science Fiction among the e-book downloads. For example, on a particular day and minute in June 2014, the book “Atlant of Ancient Worlds” had more than 2000 downloads in 48 hours. In comparison, the most popular film - an episode of the TV series “Game of Thrones” - had more than 70000 downloads.

**General e-book sites** The most simple pirate e-book sites offer any kind of book - very much like a large all-audience book store or Amazon.com’s online bookstores. They are little more than a simple search engine that spits out downloadable e-books. These types of sites have become particularly popular in Russia. At the time of this writing, *library.ru* has been the model for this kind of site and, after its crackdown, be superseded by [bookfi.org](#). Also [libgen.info](#), an online library, is an example to mention here.

What online booksellers could learn from these sites: \* Simplicity combined with encyclopedic scope. With no other site except Amazon’s, it is as easy to find and download an electronic book, no matter which genre, language, whether a bestseller or obscure. \* Simplicity in formats. The sites provide mostly PDFs as well as epub, depending on whatever file format they have available (and obtained from hacker networks).

**Specialized sites** Some websites outside or in gray areas of copyright are artistic projects. In the 1990s, “textz.com” by the Berlin-based artist Sebastian Luetgert was the first site to offer cultural, political and media theory books as

simple, gratis-downloadable text files - among them, theory classics by Theodor W. Adorno. The name “textz” was a pun on “warez”, a slang name for illegally copied software. Later, the web sites [aaaaaarg.org](#) and [Monoskop](#) provided related collections of freely downloadable art, cultural and media studies books, yet with different thematic emphases: to stimulate reading and discussion groups, or to provide a carefully hand-selected library mirroring the taste of the site owner.

Best known, and factually legendary, is [Ubuweb](#), an encyclopedic site providing downloadable sound, video and text file version of avant-garde arts records, films and books. Most of them fall under the category of small edition artists’ books, and are provided with the artists’ permission - since most these works never created revenue when they were commercially released.

All these sites run like small specialty bookstores (although they don’t sell anything). They show that the opposite of Amazon and [bookfi.org](#) can work as well: personal selection and combination of books. If they are as strong as in the case of Ubuweb and Monoskop, they create their own public that will download a work not because of having heard of it or of the author, but because being on such a particular site is recommendation enough.

## artist-/designer-run e-publishing projects

In the last couple of years, there have been first attempts to create only small presses and book download stores for [artists’ and designer’s books](#): writer, artist and publisher, James Bridle (who coined the term “new aesthetics”) created the website “[artistsebooks.org](#)” with freely downloadable epub-files by experimental writers and artists. American contemporary artist and designer Paul Chan initiated the more commercial online press [Badlands Unlimited](#) that sells artist-made visual electronic books that are proprietary to the iPad and Apple’s iBooks-platform. The iPhone/iPad app “KYR8” (slang riff on “curate”) invites artists’ to quickly make their own electronic visual zines using page templates and user’s cell phone photo collections. Other examples are [[http://www.gauss-pdf.com](#)] ([http://www.gauss-pdf.com](#)), a publisher of digital and print works and [[http://trollthread.tumblr.com](#)] ([http://trollthread.tumblr.com](#))

None of these projects compete with classical e-book stores, online reading platforms or even specialist pirate sites. They are, after all, experimental projects and artists’ portfolio pieces.

## Why change your life

From single medium to cross-media/hybrid publishing.

From-scratch development of each publication format vs. single source-multi format publishing: advantages, disadvantages

In-house development versus outsourcing to service providers: advantages, disadvantages

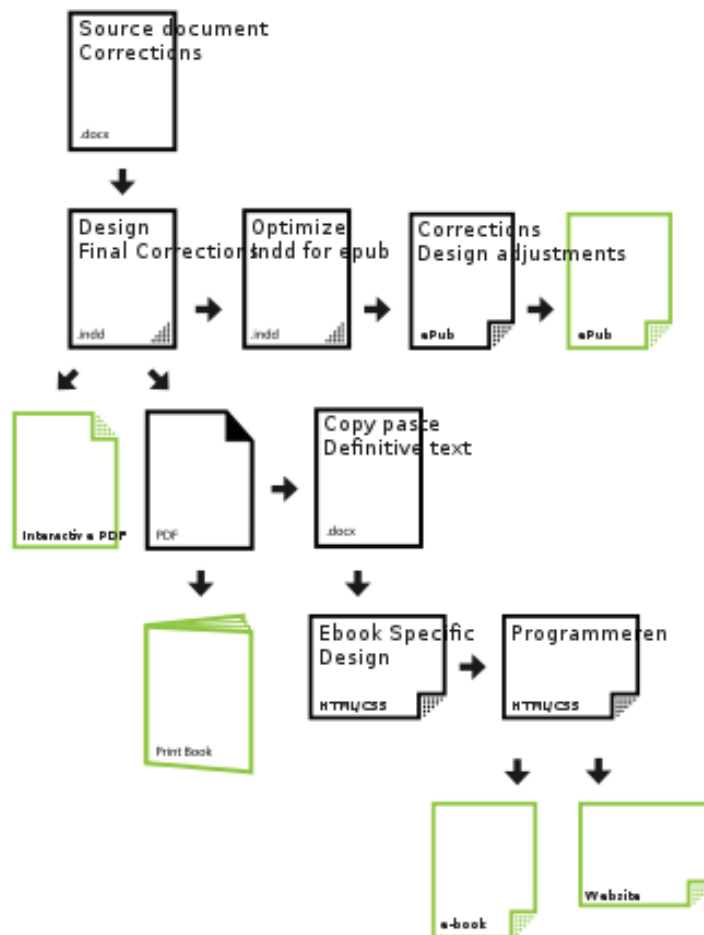
Why new workflows are essential for small edition, low budget publishing.

## Towards a hybrid workflow

One flowchart of all the three “flows”: Traditional, XML, Markdown

### WORKFLOW TRADITIONAL

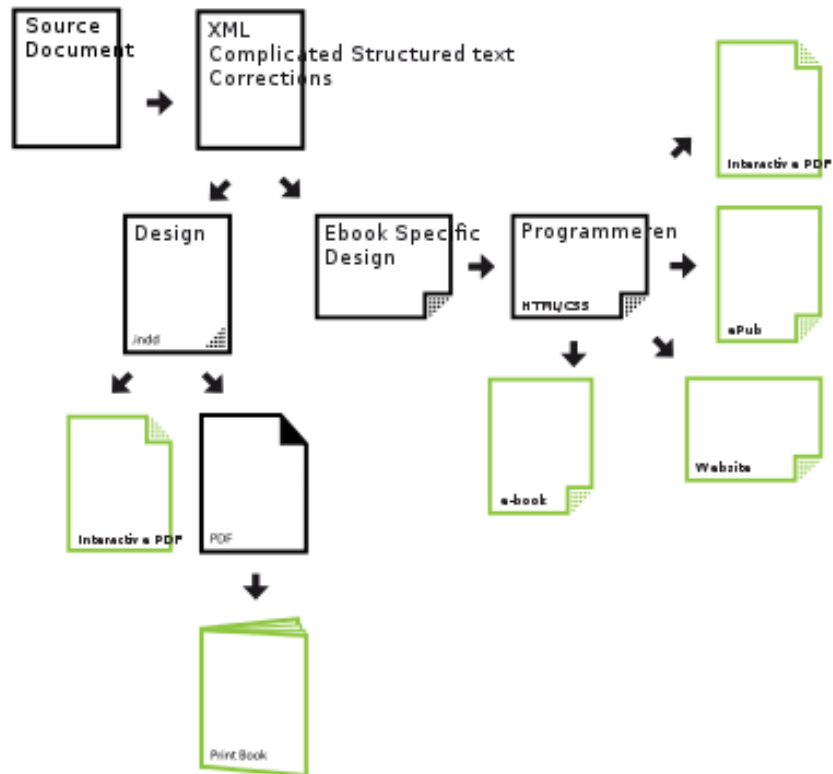
word → indesign → ebook





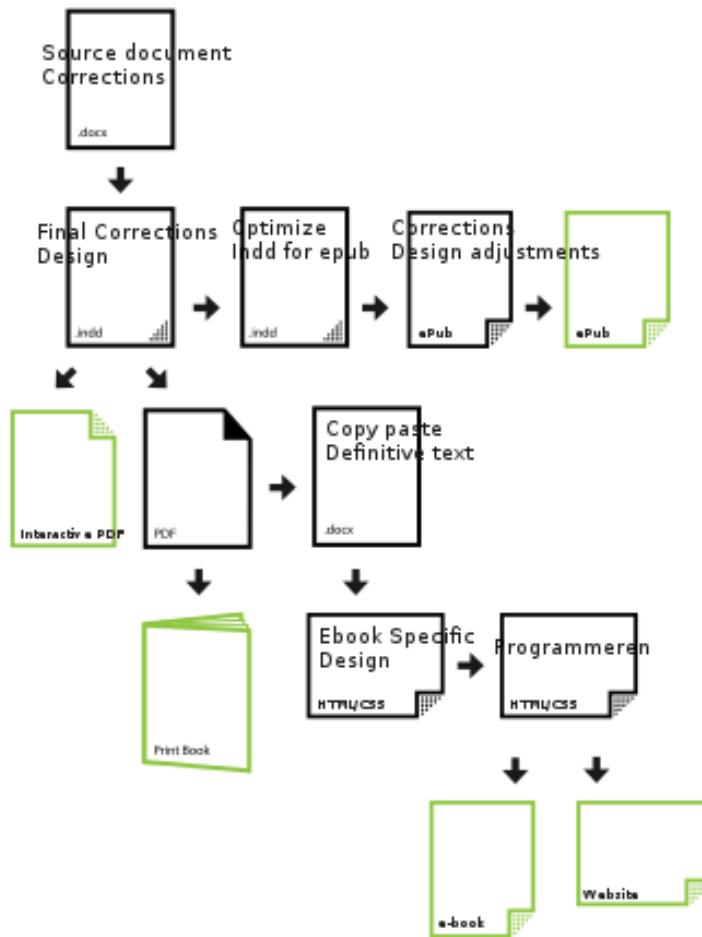
## WORKFLOW XM

? → XML → print  
→ eBook



## WORKFLOW TRADITIONAL

word → indesign → ebook



Desktop publishing workflow (from Word to InDesign to digital)

Advantages, limitations

The current workflow of desktop publishing, from Microsoft Word to InDesign, has certain advantages. It is simple, linear, there are no version branches, you end up with one consolidated manuscript, and What You See Is What You Get in relation to design. For example, in the case of analogue print where the page is fixed, changing hyphenation is both an editing and a typographic issue. However when transitioning into digital publishing, these hyphenations will be subject to change as they'll shift according to the aspect ratio of the device you're using.

The main disadvantage of this workflow is that you only target one medium and the step to go from here to digital are cumbersome, and do not make full use of its potential. It tries to implement the print into the digital, keeping the workflow one-dimensional. Working with the established software tools - Word and InDesign - it does this almost literally by allowing you to create quirky and strange 'interactive' publications in PDF, which in the end is a static inDesign document, upgraded with some interactive layers. This is a very limited vision on what the possibilities are for digital publishing. Next to this it is possible to export your publication as an .ePub from InDesign, however it is not developed as a program to produce e.pubs and the results are very messy and requires extra steps for the designer. (For a full description of InDesign to Epub see Elizabeth Castro's Epub Straight to the Point)

1. **InDesign** will never become a hybrid publishing tool as it is (too) specifically developed for print design.
2. **Word** (or any other text editor that uses this approach, for instance Open Office) will never become a structured text processor. It is a hybrid of a text editor, a desktop program with typographic attributes, and something like a semantic mark-up program. To a certain extent you can work in a structured manner using stylesheets - determining for instance different types of headers. However, the big problem is that these programs don't enforce working in this structured manner. If only one person doesn't use these stylesheets, but uses manual formatting (instead of structuring!), the whole document is messed-up. The problem with these programs is that there is no separation between formatting and structure, as if these two are equal to each-other, but especially in the world of digital publishing this is not the case.

## XML

The most detailed structuring language developed is XML. The advantage is that it is very clean, but on the other hand it is very difficult to work with. It is a purely semantic language - with a clear separation with formatting - allowing you to structure the document in a very detailed and thorough manner. These documents work like a database. The original XML document is your fundament, your manuscript, and from there you can create translations for every medium. Thus, if there is a misspelling or another error - you change this in your XML

document and you make new translation to the different media that you have produced. For formatting you would need to add a second step, like for instance CSS stylesheets. However, because XML is so detailed it becomes too difficult to work with as the tagging systems become very complex.

Advantages, limitations

Does it work for me?

## Markdown

**Advantages and Limitations** Markdown is nothing but plain, unformatted text with human-readable formatting symbols. For example, this is what the beginning of “Alice’s Adventures in Wonderland” would like in Markdown:

```
# Alice's Adventures in Wonderland
```

```
## Chapter I. Down the Rabbit-Hole
```

```
There was nothing so _very_ remarkable in that; nor did Alice think  
it so VERY much out of the way to hear the Rabbit say to itself,
```

```
> Oh dear! Oh dear!
```

```
> I shall be late!
```

```
(when she thought it over afterwards, it occurred to her that she  
ought to have wondered at this, but at the time it all seemed quite  
natural); but when the Rabbit actually **took a watch out of its  
waistcoat-pocket**, and looked at it, and then hurried on, Alice  
started to her feet, for it flashed across her mind that she had  
never before seen a rabbit with either a waistcoat-pocket, or a  
watch to take out of it, and burning with curiosity, she ran across  
the field after it, and fortunately was just in time to see it pop  
down a large rabbit-hole under the hedge.
```

“#” signifies a top-level headline, “##” a second-level headline, “\_” italic text, “\*\*” bold text, “>” a long quote. Beyond that, Markdown provides conventions for marking up bold text, lists, embedded images, links. Its popular extension MultiMarkdown also supports footnotes, tables, mathematical formulas, cross-references, bibliographies and definition lists. With simple Open Source converter programs, Markdown text like the above can be automatically translated into well-structured HTML, ePub, PDF, RTF (for importing into InDesign) and other document formats, with a single mouse click or keyboard command, requiring no manual adjustments.

Markdown is a child of Internet culture. It standardizes ad-hoc formatting signs used in E-Mail and chats, and became popular in blogging software. There

are similar plain text formatting languages to Markdown: the Wiki syntax used in Wikipedia, the language *Textile* that is used for a number of web content management systems and the language *reStructuredText* used for writing technical manuals, and many others.

Markdown & Co. are human-readable, human-friendly well-structured document formats, excellent for long-term storage and as the basis (or source code) for conversions into present and future document formats. While they are simple, they are also strict and unambiguous enough in their formatting syntax that multiple writers and editors can work on a document without introducing weirdness into its formatting. Another advantage: Markdown can be written and edited on any text program, even as E-Mail or in a web browser or on a mobile phone. Unlike Word or other classical word processing programs, it is not bound to specific software (just like in graphics, the JPEG format is not dependent on any particular piece of software, but can be edited in any graphics program).

Why do we recommend Markdown in particular? For particular publishing projects - for example, handbooks or books derived from Wikis -, it can be worth considering one of the alternatives to Markdown such as *reStructuredText*. (It's also quite easy to convert those into Markdown and vice versa.) There are, however, two reasons why we recommend Markdown as a practical tool for electronic and mixed media publishing:

1. Excellent software support. As the most popular human-readable plain text formatting language, there is a plethora of available user-friendly, high quality software for writing and editing documents in Markdown and for converting it into other formats. While Markdown can be written and edited in any computer program that allows to edit text, there are a number of very user-friendly text programs that make it easier to write and view.
2. With Multimarkdown (an extended version of Markdown), it provides all the necessary formatting and document syntax needed in arts- and humanities-oriented text publishing. It is perfectly possible to write, for example, a cultural studies Ph.D. thesis in Multimarkdown, or the essay part (complete with footnotes and bibliographical references) of an exhibition catalog.

Markdown/MultiMarkdown is however not a magic one-size-fits-all solution. It is excellent for text-heavy editorial work, but limited for visual document creation and not really usable for interactive publishing formats. It is not a layout tool, but a pure manuscript format, excellent for keeping manuscripts in a well-structured, readable, durable, software-independent format.

Markdown (and similar formatting/markup language) are meant for workflows in which there is a clear separation between editorial work - involving writers, translators and editors - on the one hand and publication design on the other. For publications where there is a high degree of interaction between writers/editors

and visual designers/artists from the very beginning of the authoring process, these document languages are not the right tool.

**Markdown versus XML** Markdown, and similar human-readable plain text markup languages could be called a “poor man’s XML”. To summarize XML is complexity hell even by the measures of computer science. It’s the holy grail of document processing, and has been deployed by large scale publishers (especially in the academic field) very successfully. For small to medium publishers, it is often overkill. Markdown provides a good middle-of-the-road solution of a format that is easily usable for non-technicians yet much better structured, and a basis for easy document conversion into HTML, epub and many other formats, than Microsoft Word and similar classical word processing programs.

Technically speaking, Markdown provides some of the same features and advantages: separation of content structure from visual layout, painless translation into multiple output formats. Their relative simplicity and human readability comes, however, at the price of extensibility and universality. XML is, strictly speaking, not a document markup language, but a meta-language (or toolkit) for building domain- and application-specific markup languages such as: a document markup language for exhibition catalogues, a document markup language for restaurant menus, a document markup language for flyers, etc.

Markdown & Co. do not provide this flexibility of building one’s own syntax, but provide only their built-in, hard-wired syntax. For example, if one needs syntax for encoding footnotes and endnotes, MultiMarkdown simply doesn’t provide it. One could think up and use one’s own syntax extension (for example <sup>33</sup> for an endnote), but this would not be supported by the word processing and text conversion programs for Markdown. In XML, there are standard methods of declaring and extending markup languages that can automatically be picked up by XML document converters.

However, the declaration of these extensions in the document syntax and conversion rules is highly complex. Even for computer scientists and engineers, XML is often so over-complex that they have resorted to simpler, human-readable language like Markdown, ReStructuredText and ASCIIDoc for software manuals.

They are not as universal and thoroughly structured as XML, but still provide the advantage of separating content structure from visual layout, along with the advantage of painless translation into multiple output formats. And lastly, XML has very complex markup that is hard to read and write for humans. Easy authoring tools for XML and any kind of XML-based document formats do not really exist yet.

## Practical tips and tricks for working with Markdown

---

<sup>33</sup><sub>1</sub>

**Word Processing / editing programs** For Apple’s Mac OS X and iOS, there are nice and very user friendly programs for editing in Markdown. None of them, however, are Open Source:

- [ByWord](#), a very user-friendly, distraction-free text writing program with built-in MultiMarkdown support and export to HTML, RTF, PDF and Microsoft Word. The program runs on Macs, iPhone and iPad.
- [iA Writer](#), a program similar to ByWord. The program runs on Macs, iPhone and iPad.
- [Scrivener](#), a word processing program popular among professional writers, for Mac OS X and Windows. Fully supports MultiMarkdown internally.
- [Mou](#), is a Markdown editor that funds on Mac OS X. It consists of features like live preview, sync scroll, auto save, powerful actions, auto pair, custom themes and CSS, HTML and PDF export, enhanced CJK support and more.

### **Document conversion programs**

- [multimarkdown](#), the original program convert MultiMarkdown files into HTML, PDF, OpenDocument (for later conversion into RTF or Microsoft Word). Open Source, runs on Linux, Mac OS X and Windows.
- [pandoc](#), similar in functionality to multimarkdown, but much more powerful. Pandoc reads more input formats (including HTML and reStructured-Text) and can output HTML5, XHTML, LaTeX, RTF, Word, Epub 2 and 3, PDF and many more. Typographic templates for the conversion can be easily customized.
- [Calibre](#), application suite that allows users to manage e-book collections as well as to create, edit, and read e-books. It supports a variety of formats (including the common Amazon Kindle and EPUB formats), e-book syncing with a variety of e-book readers, and conversion (within DRM restrictions) from different e-book and non-e-book formats. Open Source, runs on Linux, Mac OS X and Windows.

Pandoc is the tool we recommend for working with Multimarkdown, and has also been extensively used in creating this publication.

### **Practical tips** Cleaning up Markdown

Since Markdown is a document format and not a word processing program, it does not offer functions like automatic renumbering of footnotes and list items during text editing. In fact, such numbers don’t matter since everything will be renumbered during the document conversion anyway.

However, to also make the Markdown text source coherent and tidy, pandoc can be used to clean it up. The trick is to tell pandoc to convert a document from Markdown to Markdown:

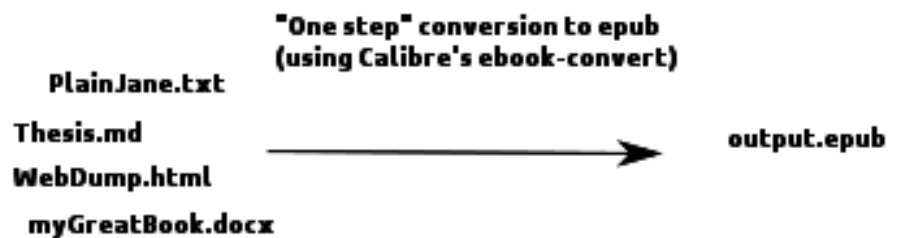
```
pandoc -f markdown -t markdown --output markdown-document-clean.txt markdown-document.txt
```

Database publishing / Content Management System

## 08 Guide: How to make a simple ePub

Focus: epub2/3, HTML5 (app) + PDF

### A simple e-book



Flow chart: workflow

### Direct conversion to ePub

Making an epub doesn't have to be complicated. As the epub standard is open and based on HTML (the same format as web pages), there's a large and growing variety of ways to convert and export to an epub from different kinds of sources. For very simple publications, it may be possible to use a tool that directly converts your document to an epub.

Two popular conversion programs that can convert from a wide variety of input formats and produce epub's are "pandoc"<sup>34</sup> and "ebook-convert"<sup>34</sup> (<http://manual.calibre-ebook.com/cli/ebook-convert.html>).

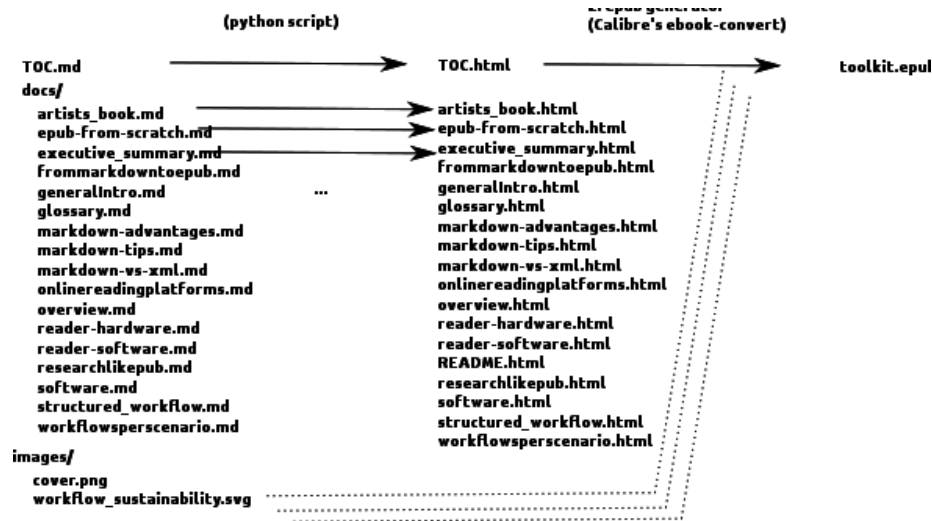
For example, consider Charlotte Bronte's "Jane Eyre," available from Project Gutenberg in a variety of formats (including epub). The ["plain text" version](#), is the complete text of the book in a single file with no styling (no fonts, sizes, or bold etc).

TODO: Give simple example of using markdown + pandoc? to produce a simple epub.

---

<sup>34</sup>Ulises Carrión, "The new art of making books", Aegean editions, 2001





## EPUB from scratch

The process of creating an EPUB from scratch is similar to developing a simple website. The main difference is that while websites can and often link to other websites, an epub is “self-contained”, any pages that are linked to, or images that are displayed must be part of the collection. Creating an epub by hand is useful for creating small personal publications, or for making publications that explore the particularities of the epub format in detail.

An epub is a Zip archive typically named with the extension “.epub” instead of “.zip”. The epub is a compressed collection of HTML files, stylesheets, and images, like the files found on a website, compiled together with some extra files that mark and structure the files so that an ereader can display them. Any file archiver that works with zip files (Archive Utility, The Unarchiver, WinZip, etc.) can open and decompress an epub file. In some cases, it might simply be necessary to rename the “.epub” with “.zip”. More information about how to automate the epub zipping process can be found [here](#).

A rudimentary epub is used as an example to explain several of its concepts and may be downloaded [here](#).

## Layout of an epub package

Decompressing an epub will reveal its directory layout and in that way make clear how an epub is set up. As explained above, the epub can be seen as a compressed Zip archive, looking as follows after unzipping it:

The *META-INF* and *OEBPS* directories and *mimetype* should always be present in an epub file and form a large part of what constitutes as an epub.



Figure 3: Figure: EPUB layout

*META-INF* contains an XML file (*container.xml*) which directs e-readers to an inventory (an .opf file) of all the files present in the publication.

*OEBPS* is the location where all the content (HTML files, images, audio, video, etc.) of the publication is stored, (nested) subcategories are possible but not mandatory. The .opf file (traditionally named *content.opf*) is important; this contains the metadata for the epub and is in turn referenced by the aforementioned *container.xml*. You might see another file with a .ncx extension (traditionally *toc.ncx*), it holds the hierarchical table of contents for the epub and is entirely optional as it isn't part of the epub specification.

The file *mimetype* contains a single line describing the epub file as **application/epub+zip**, this file allows e-readers to check whether the file is actually an epub and thus if they can read it.

These three components form the basic structure of an epub and are required in order for the file to be a valid epub.

### Creating your own epub

Most of the elements of an epub can be produced by hand in a text editor - not to be confused by a word processor like Microsoft Word or Apple's Pages. Popular text editors include BBEdit, TextWrangler or TextMate for Mac or NotePad++ and PSPad for Windows. Below follows a step by step process of creating a very simple epub.

## Creating the required files & directories

- 1- Create a directory to store the files and subdirectories for your epub in, e.g. *Example*;
- 2- Create two more directories inside the one you've just created, one called *META-INF* and the other *OEBPS*;
- 3- Using a text editor create a plain text file and add the line `application/epub+zip` to the file;
- 4- Save the plain text file, without a file extension, and name it *mimetype* alongside the two directories you created in step 2.

### Creating the required files and directories

1. Create a directory to store the files and subdirectories for your epub in, e.g. *Example*;
2. Create two more directories inside the one you've just created, one called *META-INF* and the other *OEBPS*;
3. Using a text editor create a plain text file and add the line `application/epub+zip` to the file;
4. Save the plain text file, without a file extension, and name it *mimetype* alongside the two

directories you created in step 2.

Now there are the two directories and one text file, like we saw when we decompressed the EPUB used as an example.

### container.xml

1. Again using a text editor, create a new file and save it to the *META-INF* directory with the name *container.xml*;
2. *container.xml* contains a simple structure written in XML. Below is a complete version of the document followed by an explanation of its separate parts. You may ignore the explanation without much consequence if its too technical in nature. The important part of this document is what's contained in between the quotes of the attribute **full-path** (*OEBPS/content.opf*). This attribute should point to an .opf file we'll create later on and will be stored in the *OEBPS* directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<container version="1.0" xmlns="urn:oasis:names:tc:opendocument:xmlns:container">
  <rootfiles>
    <rootfile full-path="OEBPS/content.opf" media-type="application/oebps-package+xml"/>
  </rootfiles>
</container>
```

The first line is a **declaration statement** which should always be present in XML files. This is followed by the **container** which denotes that the publication is based on the *Open Container Format* as specified by the epub standard<sup>35</sup>. The **rootfiles** tag denotes a collection of rootfiles, viz. the possible starting point(s) for e-readers to begin processing and parsing the content. In this case the rootfiles contains only one entry, called **rootfile**, this tag has two attributes - **full-path** and **media-type**. The first attribute holds the path to an inventory file (in this case *Example.opf*) containing metadata regarding the publication and its content. Finally, **media-type** is reaffirmation of the epubs mimetype.

3. Save and close *container.xml*.

**The OPF file** The OPF file is an important part of the structure of an epub. It is located in the *OEBPS* directory and contains the necessary metadata to accurately describe the publication, but also because it can contain the linear reading order which, in combination with the contents of *toc.ncx*, may be used by e-readers to build navigation menus or a table of contents. The OPF file is too long to be included verbatim in this document, but the most important sections are referenced below.

---

<sup>35</sup><http://www.idpf.org/epub/30/spec/epub30-ocf.html>

```

<metadata xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:opf="http://www.idpf.org/2007/opf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <dc:title>Example EPUB</dc:title>
  <dc:language>en</dc:language>
  <dc:identifier id="BookId">http://digitalpublishingtoolkit.org/ExampleEPUB.html</dc:
  <meta property="dcterms:modified">2014-03-28T14:11:50Z</meta>
  ...
</metadata>

```

The metadata section describes the current publication. It lists information such as the title, author, publisher, etc. Most of these entries are identical to what librarians use to catalogue publications. Parts of the metadata section are used by e-readers to organize collections.

```

<manifest>
  <item href="styles.css" id="css1" media-type="text/css"/>
  <item href="cover.png" id="cover" media-type="image/png" properties="cover-image"/>
  <item id="chapter1" href="Cover.html" media-type="application/xhtml+xml" />
  <item id="chapter2" href="Page-01.html" media-type="application/xhtml+xml" />
  <item properties="nav" id="toc" href="toc.html" media-type="application/xhtml+xml" />
  <item href="toc.ncx" id="ncx" media-type="application/x-dtbncx+xml"/>
</manifest>

```

A manifest lists all the resources available in the EPUB package, with the exception of directories, the *mimetype* file, the contents of *META-INF* and the .opf file itself. A manifest file can be a pain to produce by hand for large epubs, as the list of resources utilised in the publication is bound to be long. Every resource has a unique **id** and should be referenced by a relative path in the **href** tag and described in the **media-type** attribute. The entry with the *cover.png* is of special interest, as the attribute **properties** describes the fact that the image may be used by e-readers as the cover image for the publication - e.g. for use in collection overviews.

```

<spine toc="ncx">
  <itemref idref="chapter1" />
  <itemref idref="chapter2" />
</spine>

```

Lastly, the **spine** lists all the pages present in the publication and it's listing arrangement tells an e-reader the linear reading order of the publication. The **spine** may only contain (X)HTML pages, not images or other content. The **toc** attribute refers to the **id** of the *toc.ncx* in the manifest.

**The Content** As stated in the introduction of this section a large part of an epub is a collection of HTML files which are often interlinked. The process of creating the pages of an epub is similar to building a website, but with the particular limitations of e-readers in mind - limited support for rich media, colour, etc. An overview of the limitations can be found [here](#). Pages should be written in XHTML, a variant of HTML which was created to make HTML more extensible and increase the interoperability with other data formats. Cascading Style Sheets (CSS) may freely be used, though a lot of e-readers ignore or do not parse many of the style definitions.

**Packaging** Creating an .epub file is as simple as selecting both the *META-INF* and *OEBPS* directories and the *mimetype* file and creating a ZIP archive. This may be done by using the built-in archive utility of the operating system, or an external program like *The Unarchiver* (Mac) or WinZip (Windows) or a special purpose utility.<sup>36</sup> Some of these programs create unnecessary (hidden) files inside the archive which might invalidate your epub. Most e-readers will safely ignore extraneous files, though parse the document properly. Validation of epubs can be done online using the [EPUB Validator](#) or [a desktop application](#). The .zip extension of the archive may then be renamed to .epub. This file can then be opened in an e-reader like Calibre, iBooks or similar applications.

## 09 Guide: E-publications per genre

### General questions

### Research publication

#### General

Traditional, print oriented workflow for text-centred works, with additional illustrations / videos / resources online, and extended referencing and/or indexing See Kimmy's visualized workflow for the INC, which will be generalized for [research-type publications](#)

- Editor works with author(s) on manuscript
- Several versions going back and forth between different people in different roles (editor, author, copyeditor, designer)
- Final version of the text will be in a text editor or Word format

---

<sup>36</sup><http://www.mobileread.com/forums/showthread.php?t=55681>

- ### New workflow

So instead of having a final text file (e.g. Word) which is changed in InDesign - both of which are not good formats to convert to an ePub, the storage / archive file has to be in a format that can translate into InDesign for the print edition and ePub / mobi / web for the e-publication. The most suitable format for this is **html** - because it allows footnotes (which Markdown does not). The html can easily be converted to ePub, though not so easily to InDesign... (working on this) Flow chart: workflow -







## Instructions

### How to adjust the style guide?

The decision about the desired output leads to certain steps to get the right input. First of all it is therefore necessary to communicate this to the authors and others involved in the publishing process. Adjust your style guide which is leading in the communication with authors/designers/programmers etc.

Specify for example:

file format - for example, if you are following the proposed workflow, request the file to be sent in either html directly (which can be converted into ePub) or docx (which can be easily converted into html)

image style and format - while the print book has b/w pictures, the e-book enables colour images; print requires large quality 300 dpi .tiff images, while these may be too heavy for a digital publication, skyrocketing the size of the book, which is bad for download / and or storage space on an e-reader

formatting styles - use Header 1 style for title and author, Header 2 for article sections and Header 3 for subsections \* URL style - full URLs should be clickable but not stylized as links (color or underlining)

which metadata the author has to provide - e.g. keywords, contact information, abstract, biography

an extended example of a style guide adjusted for a hybrid workflow can be found in the appendix

**What metadata to add?** Make a list of metadata to be added to the texts metadata provided by the author(s)

metadata added by the editor/publisher \* an extended example of a metadata list adjusted for a hybrid workflow can be found in the appendix

### Do editorial criteria regarding the content change?

Editorial criteria for the content can also differ in a hybrid workflow and depending on the desired output and the answer to the question of the relationship between the print and electronic edition:

(not discussed in this guide) consider another style of writing, ‘writing for the web’

is interactive material needed, more or other images than can be published in print, video, audio, etc.

## Specific issues for this publication type

Important issues in the hybrid workflow for research publications are 1. the editorial process, with comments and copyediting 2. footnotes and referencing

### Based on the above, what to choose?

How to make a research publication in ePub2/3, HTML5 (/app), PDF

A step-by-step guide

Converting a Docx directly to EPUB using Calibre (post Silvio)

Pressbooks as an example of available open source tools? (post Silvio)

From InDesign - when it's 1:1 use the certified PDF that goes to the printer, downsized when needed.

Possibilities for reading online: Scribd, Issuu, in-browser reading

Advantages, disadvantages

## **Art/design catalogue**

traditional workflow vs. new workflow

### **General**

traditional vs new workflow ### Decision making Should it be a 1:1 transfer from paper book to e-book?, What are alternative possibilities? ### Instructions How to adjust the style guide? What metadata to add? Do editorial criteria regarding the content change? ### Specific issues for this publication type ### Based on the above, what to choose?

How to make an art/design catalogue in epub2/3, HTML5 (/app), PDF

A step-by-step guide

Advantages, disadvantages

## **Artist/designer book**

Flow chart: workflow

### **General**

traditional vs new workflow ### Decision making Should it be a 1:1 transfer from paper book to e-book?, What are alternative possibilities? ### Instructions How to adjust the style guide? What metadata to add? Do editorial criteria regarding the content change? ### Specific issues for this publication type ### Based on the above, what to choose?

How to make an arts/design periodical in epub2/3, HTML5 (/app), PDF

A step-by-step guide

Advantages, disadvantages

Lately, artists' and designer's books have grown from a niche phenomenon to a major genre within arts and design publishing. It could be argued that the book as an art or design object in its own right has become more prominent now that the function of the book as a generic carrier of information (such as in a telephone directory, a dictionary or a run-of-the-mill novel or non-fiction paperback) is increasingly taken over by the World Wide Web and by e-readers.

The history of artists' books - or "bookworks", as the Mexican-Dutch writer and visual artist Ulises Carrión called them<sup>37</sup> - is often traced to the beginnings of concrete poetry, Fluxus and conceptual art, and the artists' books (and book-like objects) made among others by Dieter Roth and Ed Ruscha. But one could just as well include the Bauhaus book series ("Bauhausbücher") of the 1920s, earlier Dadaist and Futurist pamphlets and self-published books, the books made in the 19th century Arts and Crafts movement and even 16th century illustrated books like Sebastian Brant's and Albrecht Dürer's "Ship of Fools" that were crafted in workshop collaborations between writers, visual artists and printmakers.

The common denominator of these books is that they defy easy categorization and genre characteristics, making it difficult to give standard recommendations for their design as electronic books.

**Transfer from paper to electronic?** The more medium-specific an artist's book is, the less sense it makes to translate it 1:1 into an electronic book. Dieter Roth's sculptural book objects, for example, would change from book art works to mere depictions of book art works when reproduced electronically.

On the other hand, many contemporary artists, designers and media activists experimented with electronic publishing as a "poor" (i.e. simple low tech) medium of social sharing rather than a "rich" visual and tactile medium: - In the 1980s and early 1990s, many makers of zines (do-it-yourself small press periodicals) became makers of *e-zines*, electronic zines that used dial-up computer bulletin boards and the Internet as noncommercial samizdat media. [Their typical publication] (<http://www.textfiles.com/directory.html>) format were simple plain text files. Many e-zines worked around their technical limitations by using ASCII art, typograms as they had also been produced in 1960s and 70s concrete poetry on typewriters, and with homebrew formatting codes ("\_" for underlines, "#" for headlines etc.), the immediate precursors of the Markdown format described here in this toolkit. - In the early 1990s, there also existed a hacker culture of "disk mags" for the Commodore Amiga and Atari ST home computers. These electronic magazines were anonymously published on floppy disks and were based on self-written computer programs (factually, early "apps") that displayed their animated text and visual contents. - Since the late 1990s, there have been a number of artist-run sites - [textz.com](http://textz.com), [ubu.com](http://ubu.com), [aaaaarg.org](http://aaaaarg.org), [monoskop.org](http://monoskop.org) - that offer free downloads of cultural theory and arts-related books, typically in simple formats such as plain text, PDF and epub. In these examples, the focus

---

<sup>37</sup>Ulises Carrión, "The new art of making books", Aegean editions, 2001

is not on the book as a visual object, but on concepts and politics of its sharing and dissemination; in other words, artists' publishing as electronic samizdat. In such scenarios, artists' publishing as simple 1:1 transfers from print originals to electronic reproductions can make sense.

**Technical solutions for samizdat publications** For such activist or minimalist projects, the lowest technical denominator and most easily readable file formats are advisable: - plain text (ASCII) as the most simple, compatible and minimalist solution - single-file HTML. It is possible to directly embed images into an HTML file (without providing them as separate files); technical instructions can be found [here](#). - PDF. This format is widely readable and best suited for faithfully reproducing print books, but limited in its readability on different devices and hardly editable (more explanation [here](#)). - epub. This format is factually just HTML for offline reading, with improved publication meta data and improved compatibility to e-readers. Projects can very easily be made and provided both in HTML and epub. Since the design of most samizdat books does not differ from that of [research publications](#) or other visually simple publishing formats, no special design advice is necessary here, except the principle of “worse is better”: the smaller the file size, the most compatible and universally readable the file format, the better, even if this comes at the expense of typographic and visual quality. An plain text files, then, might be preferable to a nicely designed PDF file.

## How to make visually oriented artists' books

**Pre-history and general issues** There is, factually, a rich tradition of artists' audiovisual electronic books: It began in the 1990s with hypertext and interactive multimedia literary experiments on floppy disk, CD-ROMs, later web sites and mobile apps. The [Electronic Literature Organization](#) and the international research project [ELMCIP](#) document it extensively. Much of this documentation has become media archeology since multimedia formats have become obsolete: CD-ROM applications that no longer run on contemporary computers, websites whose links or plug-ins are no longer working or not compatible to today's browsers anymore.

This problem was greater in the 1990s and early 2000s when working open, cross-platform multimedia standards barely existed. But even today, the rule explained in chapter ... still applies: The more complex the audiovisuality of an electronic book, the less compatible it will be to all kinds of different electronic reading devices, and the more technical updates it will likely need over the course of time.

**Simple solutions** Electronic visual books can be made in very simple ways:  
\* As a sequence of images, embedded into an epub file , a self-contained HTML

file or a PDF file. \* As a PDF file. PDF generally is the most easy-to-use straightforward format for visual publications in a universally working format.<sup>38</sup> PDF documents can be graphically designed to work on different screen sizes, and resolution of embedded visuals can be decreased to keep the file size friendly for downloading. Still, the format is essentially limited to fixed document sizes and remains an electronic representation of printed matter. \* Other standard file formats creatively (ab)used as document formats for visual books: animated GIF graphics files for the digital equivalent of flip books, for example, mp4 video files displaying a real-time book, mp3 audio files triggering abstract art on the volume meter display of an audio player, JPEG files with encoding artefacts of corrupted bits.<sup>39</sup> \* Self-contained HTML.

(- iBooks author: medium solution between epub and PDF: example Badlands Unlimited) (- self-contained HTML5) (- epub 2, epub 3) (- plain text: e-zines, BBS era e-books, typograms/typoscripts) (- Problem with it: distribution, books only available in the Apple ecosystem)

## Art/design periodical

- Art OPEN Magazine of e-Flux journal, advise to use the web, instead of e-pub.
- Niet commercieel website / PDF
- Academic magazine (jStor)
- Public Magazine
- Commercial / app store model.

## General

traditional vs new workflow ### Decision making Should it be a 1:1 transfer from paper book to e-book?, What are alternative possibilities? ### Instructions How to adjust the style guide? What metadata to add? Do editorial criteria regarding the content change? ### Specific issues for this publication type ### Based on the above, what to choose?

How to make an arts/design periodical in epub2/3, HTML5 (/app), PDF

A step-by-step guide

Advantages, disadvantages

---

<sup>38</sup>For long-time durability, the “PDF/A” format is preferable to run-of-the mill PDF. [PDF/A](#) stands for *archival PDF* and is an ISO standard originally crafted by Adobe in collaboration with non-profit organizations for information management. As opposed to generic PDF, PDF/A requires that all fonts, references and color profiles are fully embedded into a document.

<sup>39</sup>Used as a medium of artistic experimentation among others by net artists since the 1990s and by conceptual poet and Ubuweb founder Kenneth Goldsmith.

## 10 Futurology

Where will e-publishing be in 2020?

### Reading technology

Both hardware / software

There is an increase in reading on the smartphone. Some speculate that e-readers will be obsolete. (<http://nymag.com/daily/intelligencer/2014/06/heres-what-the-future-of-reading-looks-like.html>)

Books as apps - blurred lines between books / websites / apps

Advantages and disadvantages will remain the same: affordability and portability of whole libraries vs. independence from electricity and visual-tactile advantages of paper books.

Improvement of e-paper: color e-paper, flexible e-paper

4K displays for mobile devices will push display resolution

possible anti-cloud/anti-obsolescence backlash - for example in the creation and maintenance of personal

### Reading culture

Mainstream of publishing will remain in traditional textual formats. Reader technology is now good enough to allow pleasant on-screen reading.

Foreseeable: generic forms of publication (such as the paperback) will slowly migrate to electronic, paper publishing will become more visually and tactile oriented.

Something about the much discussed 'change' in reading - deep versus shallow reading (Nicholas Carr). But is it so? See for example: <http://www.ft.com/cms/s/2/53d3096a-f792-11e3-90fa-00144feabdc0.html#axzz374aeWjXN> Two things are notable in this. First, paper might simply be a cultural preference, and e-readers something we just have to get used to using in a good way. Second, the 'preference' for deep reading might as well be relative. For an active working with the text (for interpretation and learning for example) skimming, adding notes, looking up background information et cetera might work better than immersive, deep reading. And e-readers, tablets, and phones might be better suited for that purpose.

Other ways of dealing with texts open up, allowing a more interactive relationship between reader and text. Not only in the sense of adding audio and video, but also leaving the reader the choice of entry into the text, setting out an individual

reader path per person. See for example [http://futureofthebook.org/blog/wp-content/uploads/2014/05/Tether\\_Mise-en-Page\\_FINAL.pdf](http://futureofthebook.org/blog/wp-content/uploads/2014/05/Tether_Mise-en-Page_FINAL.pdf)

## **Publishing culture**

Likely: switch from individual product retail to rental/subscription model. Subscription to complete libraries with unlimited access to media within the rental period. (Examples: Netflix, Spotify, Adobe Creative Cloud, JSTOR).

Writers / artists / designers who self-publish will increase

Amazon spreading, also causing chances for independent publishers?

Blurring lines between media company / publisher / writer. What is the added value of a publisher? What distinguishes the publisher from those other companies or independent one-man-publishers? How will publishers redefine their expertise? (<http://www.litragger.com/literature-news/commentary/5-myths-about-the-new-era-of-publishing/>)

Publishing becoming a function rather than an industry: <http://www.idealogue.com/blog/atomization-publishing-as-a-function-rather-than-an-industry/>

## **NOTES / IDEAS**

Que Sera, Sera (Whatever Will Be, Will Be), is the never ending question for electronic publishing of art books as well. In art books we try to describe, show, analyse and collect works of art. These works can be made of everything, but more often than not they deal with tactile objects, paintings, drawings, sculptures, etc, etc. In many books these works are reproduced and discussed. But writing itself is also an art form and the mutual interaction between texts and tactile objects becomes more and more integrated. In the 20s and 30s of the previous century collages became very popular. Here texts, newspaper clippings, photographs and any other expressions, mostly on a paper substrate were put together to obtain a new artwork. In an electronic environment we transcend this collage technique into a new substrate. But now enormous range of novel opportunities for creative expression becomes possible. As we showed above in this booklet, gluing, dissecting and endless reorganising enables us to create new works that can express themselves differently in different technological environments as well as tailored to the readers's desire of the moment of consumption. Electronic publishing becomes like a gigantic fruit salad bar, where the end user is able to or fill her own plate or to make a choice between prepared combinations. This not the same as blending all ingredients into one smoothie, which is an end phase from where no way back is possible.

## 11 Glossary of technical terms

### A

#### *Acronym*

An acronym is a shortened version of a phrase, often taking the first letters. For example, GIGO is the acronym for ‘Garbage in Garbage out’ or WYSIWYG, otherwise known as ‘What you see is what you get’.

#### *Adware*

Any software which serves banner ads or pop-up ads to you while in use. It is sometimes installed in freeware or shareware which you download from the nets, and provides one more channel for advertisers to reach you. Some adware will also track your files, net usage, and software and report it back to advertisers to help them channel relevant ads to you.

#### *Aliasing, anti-aliasing*

Pictures on your computer monitor are made up of square pixel. When the edge of a solid colored object in a GIF image is a diagonal or curved line, and it is displayed against a contrasting color, the edges appear jagged, like stair steps. This jagged appearance is called aliasing. The jagged appearance can be softened by filling in adjacent pixels with intermediate colors between the object and the background. This softening of the edges is called anti-aliasing. Software like PhotoShop can apply anti-aliasing for you automatically. One problem with anti-aliasing GIF images is that it increases the number of colors used, necessarily increasing the file size. You must decide whether quick loading or smoothed edges serves your needs better.

#### *Algorithm*

This is the name given to a “defined set of steps that can be used to complete a task”. For example there are ‘algorithms’ worked out for sorting a list efficiently. A computer programmer can buy a book of algorithms so they not need to re-invent the wheel when they have to code for a job that has already been worked out. Having bought the book, they would look up the problem of ‘How to sort a list’ and a good book would tell them a number of ways to do this. Algorithms are not computer code,

#### *Android*

Is een opensourceplatform en besturingssysteem voor mobiele telefoons, tablet-pcs, koelkasten, camera’s en meer, gebaseerd op de Linuxkernel en het Java-programmeerplatform. Android is het meest verkochte besturingssysteem op mobiele telefoons ##### Amazon Internetwinkel waar digitale uitgave worden aangeboden in file format .AZW, KF8 ##### API Een “application programming interface” is een formele specificatie van hoe twee software systemen met elkaar kunnen communiceren. De API is als het ware een technische brug tussen



Applicatie A en Applicatie B. Stel je wil met een (web)applicatie de 10 meest recente tweets van een bepaald Twitter-account ophalen dan kan dit via de API die Twitter aanbiedt. Door een verzoek naar de API van Twitter te versturen krijg je via die service een “antwoord” in een gestructureerd standaardformaat (vaak in een taal als XML of JSON, e.d.) terug dat gebruikt kan verder verwerkt kan worden. Veel APIs behelzen ook schrijftoegang, waarbij bestanden of ander-soortige objecten (afbeeldingen, tekst, etc.) van Applicatie A naar Applicatie B verzonden via de API. ##### App Een mobiele applicatie of kortweg app is een software-applicatie die ontworpen is om te draaien op een smartphone, tablet of een ander elektronisch handapparaat. Met behulp van apps is het mogelijk eenvoudig extra functies aan een mobiel apparaat toe te voegen, zodat deze kunnen worden uitgebreid tot multifunctionele communicatieapparatuur. (Er wordt onderscheid gemaakt tussen Web app, Native app en Hybrid app) ##### App store Deze onlinewinkel (ook wel distributieplatform genoemd) is bereikbaar via een app op het apparaat zelf en vaak ook via een website op een desktop of laptop. Elk mobiel besturingssysteem heeft een eigen winkel, zoals App Store (Apple), Google Play, Windows Phone Store en BlackBerry App World, Kindle store Amazon. Apps voor het ene besturingssysteem kunnen niet zomaar op een ander systeem geïnstalleerd worden. Dit betekent dat per besturingssysteem een app ontwikkeld moet worden. De kosten voor de ontwikkeling zijn mede hierdoor relatief hoog. Er is geen internetverbinding nodig (behalve voor updates). ##### azw (.azw) Amazon word, file format used by the Amazon Kindle e-Book reader

#### *Anchor*

An ‘anchor’ is a specific location within a HTML-web page that can be jumped to by clicking on a hyperlink.

#### *Animated GIF*

A GIF2 graphic file, which consists of two or more images shown in a timed sequence to give the effect of motion.

#### *API*

Application Program Interface. An interface between the operating system and application programs that specifies how the two communicate with each other.

#### *Applet*

An Applet is a small program that is downloaded from a web page and executed by browser software. Also, an HTML tag that defines an applet program, which is written in Java and designed to run on a web page. It can be used to display scrolling text such as a marquee across the top of a page or other animations

#### *ASCII*

American Standard Code for Information Interchange. A standard way to encode upper and lower case letters in the English alphabet, numbers, and special characters using only seven bits, and therefore limited to 128 characters.

## *AVI*

Audio Video Interleaved. A Microsoft video format where audio and video coding appears in alternate segments. AVI files will end with an .avi extension.

## *AZW*

# **B**

## *Bandwidth*

Literally, the frequency width of a transmission channel in Hertz, kiloHertz, megaHertz, etc. Often used as an expression of the amount of data that can be sent through a circuit. The greater the bandwidth, the greater the amount of data that can travel in a given time period.

## *Baud*

Rate of transmission speed in a signal - the number of changes of state, such as voltage or frequency, per second in a signal. Named for the French teleprinter inventor Baudot. In simplest systems, it is synonymous with bits per second. In more complex systems, a baud may include more than one bit.

## *Binary*

A number system using base 2 instead of the usual (human) base 10, which is normally referred to as the decimal system. Computers use base 2 because they can only recognise two values, 1 or 0. This is simulated electronically by using a device, such as a switch, which is either on (1) or off (0). All numbers are represented by combinations of ones and zeroes, thus the number 9 is represented as 1001, the right-most column being the units column and the other columns, moving from right to left, being 2, 4, 8.

## *Bit*

Contraction of *binary digit*. A bit is the smallest measurement unit of computer memory or data transmission speed.

## *Bitmap*

A computer graphic which is defined by specifying the colors of dots or pixels which make up the picture. Also known as raster graphics. Common types of bitmap graphics are GIF, JPEG, Photoshop, PCX, TIFF, Macintosh Paint, Microsoft Paint, BMP, PNG, FAX formats, and TGA. The file extension is .BMT.

## *Blog*

Short for web log; usually a chronological record of thoughts, links, events, or actions posted on the web.

## *Browser*

Software that will load and display a web page. A browser interprets the HTML or XML code from the web page files, executes embedded scripts and programs, provides encryption/decryption for security where needed, displays graphics (except text-only browsers), plays music and video, and provides links to related pages. Browsers are based on standards developed by the World Wide Web Consortium. The major browser software developers participate in these standards, but each of them also builds in their own proprietary codes, whether or not they are approved. These differences in browsers create a challenge for web page developers.

#### *Byte*

A measurement of computer memory or disc capacity. A byte comprises 8 *bits*.

## C

#### *CAD/CAM*

Abbreviations for Computer Aided Design / Computer Aided Manufacturing. A process of drafting, designing and manufacturing with the aid of a computer. CAD enables the user to manipulate drawings, including 3D drawings, and viewing them from a variety of angles. CAM is a general term for computer support during the manufacturing process.

#### *Cascading Style Sheets (CSS)*

Cascading Style Sheets are a feature of HTML4 that enables a range of styles for headers, body text, bullet points, links etc., to be specified for hypertext documents. This makes it possible to set up CSS file containing a library of styles that are used throughout a website, thereby facilitating consistency. If a style needs to be changed throughout a website it only needs to be changed once in the CSS file and then it will be applied automatically.

#### *CMS*

Abbreviation for *Content Management System*, a software package that makes it possible for non-technical users to publish content (text, images, etc) on a website.

#### *Compatibility*

Pieces of hardware and/or software which are capable of being used together are described as *compatible*.

#### *Compression*

Compression is a technique to make a file or a data stream smaller for faster transmission or to take up less storage space. There are a number of programs that will compress files, such as PKZIP, WinZip, Stuffit, gnu zip, and many more. Files with the following extensions are almost always compressed files: arc, arj, gz, lha, lhz, tar, tgz, Z, zip, and zoo.

### *Computer Program*

A set of instructions that the computer carries out in sequence to perform a given task. Programs are written in English-like programming languages (e.g. C, Pascal, JAVA, C++) , and are then converted into binary machine instructions via a compiler or an interpreter.

### *Cookie*

A cookie is a short file put on your system by a web page when the user visits a website for the first time. It includes information about your usage and facilitates the current interaction. For example, it may include the information that you have logged into a passworded area already in the current session and don't need a second password check. There are many uses for cookies, they may be erased at the end of a session or retained until the next session, and they may be encrypted or in plain text. If retained, the next time that the user visits that site, the information in the cookie is sent back to the site so that the site can tailor what it presents to the user, e.g. tastes in music or shopping habits.

## **D**

### *Data*

Strictly speaking the plural of “datum”, but now usually considered as a collective noun in the singular, with the plural form “data items” or “items of data”. Data is information in a form which can be processed by a computer. It is usually distinguished from a *computer program*, which is a set of instructions that a computer carries out. Data can be text or sets of figures on which a computer program operates.

### *Device*

Is een elektronisch handapparaat zoals: smartphone, tablet-pc, phablet, personal digital assistant (pda), mobiele telefoon, draagbare spelcomputer (PlayStation Portable, Nintendo DS, PlayStation Vita). #### Devices Tablets: iPad, Androids, Kindle Fire. Smartphones: iPhone, Blackberry... Desktops #### DRM Digital Rights Management is technologie die digitaal opererende uitgevers in staat stelt om het gebruik en verbruik van hun auteursrechtelijke beschermde materiaal te controleren. Als zodanig is de technologie omstreden, de technologie zit veelal de betalende gebruiker in de weg en vormt een belemmering voor het vrij gebruik van gekochte digitale goederen. Een MP3 of EPUB beveiligd met een specifiek soort DRM werkt bijvoorbeeld alleen op apparaat X van fabrikant Y en is niet vrij te transporteren naar andere apparaten zonder extra ingrijpen (bijvoorbeeld het kraken van de beveiliging).

### *Desktop Publishing*

### *Directory*

A location on a disc containing a group of *files* and *subdirectories* grouped together for organisational purposes. The term is used synonymously with *Folder*, which has become a more common term since the introduction of Windows'. Subdirectories are sometimes referred to as "child directories" of the "parent directory". The topmost directory on a computer, which is the parent of all directories on the disc, is known as the *root directory* and usually has the *pathname* C:\

#### *Dithering*

The technique of combining dots of primary colours to give the appearance of intermediate colours. Dots are combined in a square area, known as a *dither matrix*, to simulate a dot of an intermediate colour.

#### *DOS*

Disk Operating System

#### *Dot Matrix Printer*

An older type of printer that works by firing sets of pins in different combinations at an ink ribbon located against a sheet of paper. Such printers produce text that looks "ragged". Laser printers and ink-jet printers are now much more common..

#### *Dpi*

Abbreviation for Dots Per Inch. A measure of the of the quality of output, i.e. the number of dots per square inch produced by a *printer* or *scanner*, also referred to as its *resolution*. A resolution of at least 300 dpi is considered reasonable for the production of high-quality output by a printer and 1200 dpi by a scanner, but modern printers and scanners can produce many more dots per square inch. The resolution of a scanner may also be expressed by two numbers. These are mostly the same, e.g. 1200 x 1200, but you may also see 1200 x 2400, which means that the number of horizontal dots is different from the number of vertical dots

#### *Drm*

## **E**

#### *e-book*

Is een verzamelnaam voor digitale documenten die weergegeven kunnen worden op een e-reader evenals op beeldschermen van computers en smartphones. ##### EPUB Electronic publication is een vrij en open e-bookformaat dat ontwikkeld werd door het International Digital Publishing Forum (IDPF). EPUB wordt soms ook wel eens aangeduid als ePub, ePUB, EPub of gewoon epub, maar de voorkeur van de maker is EPUB. EPUB is ontworpen voor reflowable content. ##### e-reader Is een apparaat waarmee men digitale teksten kan lezen. De

twee voornaamste vormen van digitale teksten op een e-reader zijn het e-book en de e-paper, als equivalenten van respectievelijk het boek en de krant.

#### *E-ink*

Electrophoretic ink, a sytem in which small particals are electrostatically charged in such a way they can become black or white and remain so. Hence, electricity is in fact only used by refreshing the page, which reduces considerably the power consumption and makes it an ideal technology for long reading. The resolution of the stable screen is very high. A disadvantage is that, like paper, the presentation does not allow motion pictures or animations.

#### *Epub*

#### *Encryption*

A system of coding that helps prevent access to private information on computer networks or on the Web.

## **F**

#### *FAQ*

Abbreviation for Frequently Asked Question.

#### *File*

A *file* in computer jargon can be used to describe many different things. It may be a Computer Program, a document file created with a Word-processor, an image file, an audio file, a video file, etc. Think of it in the same way as you would think of a file in a filing cabinet. A file has a name that describes what it is, and the file is stored in a place where you can easily find it.

#### *File extension*

One popular method used by many operating systems, including Windows, Mac OS is to determine the format of a file based on the end of its name—the letters following the final period. This portion of the filename is known as the filename extension. For example, HTML documents are identified by names that end with .html (or .htm), and GIF images by .gif. In the past names were limited to an eight-character identifier and a three-character extension. There are only a limited number of three-letter extensions, so, often any given extension might be linked to more than one program. Many formats still use three-character extensions even though modern operating systems and application programs no longer have this limitation. Since there is no standard list of extensions, more than one format can use the same extension, which can confuse both the operating system and users.

#### *File format*

A file format is a standard way that information is encoded for storage in a computer file. It specifies how bits are used to encode information in a digital

storage medium. File formats may be either proprietary or free and may be either unpublished or open. A specific file format demands specific programs to read.

#### *File Permissions*

Files stored on a computer usually have *permissions* governing which users are allowed to read, amend or execute them. This is particularly important in a school, college or university network environment, where teachers and lecturers may have the permission to amend certain files, e.g. documents that they have created, but students are only allowed to read them. File permissions are usually determined by network managers.

#### *Flat text*

#### *Formatting*

The process of preparing a writeable disc for use. Formatting creates a structure on the disc which enables it to hold data.

#### *Freemium*

## **G**

#### *GIF*

Abbreviation for Graphic Interchange Format. A file format used for storing simple graphics. GIF files use a palette of 256 colours, which makes them practical for almost all graphics except photographs. Generally, GIF files should be used for logos, line drawings, icons, etc, i.e. images that don't contain a rich range of colours. A GIF file containing a small number of colours tends to be quite small, but it will be big if the image has a wide range of colours, e.g. a photograph. GIF files are commonly used for storing images on the Web. GIF files are also suitable for storing animated (i.e. moving) images.

#### *GitHub*

Collaboration platform, code review, and code management for open source and private projects.

#### *Graphical User Interface (GUI)*

An Interface<sup>1</sup>, i.e. a software package, that enables human beings to control what happens on their computers. A GUI consists of graphical elements known as icons and enables the user to run programs and to carry out other operations such as copying information from one Folder/directory to another, deleting files, etc by clicking on these icons, opening and shutting windows and dragging and dropping with a mouse.

## H

### *Hexadecimal*

A number system used in computers in which numbers are composed of combinations of 16 digits, using 0-9 then the letters A-F to represent 10-15. Hex allows binary numbers to be expressed in a more compact and comprehensible form. For example,  $255 = \text{FF (hex)} = 11111111 \text{ (binary)}$

### *HTML*

Abbreviation for Hypertext Markup Language, structuring language based on SGML. The coding system used for creating pages on the World Wide Web1. HTML enables the author to control how the page appears and to insert Hypertext links within one Web page or to other pages anywhere on the Web.

### *HTTP*

Abbreviation for Hypertext Transfer Protocol. The transfer method (*protocol*) used by the World Wide Web to transmit and receive Web pages.

### *Hyperlink*

A contraction of hypertext link, the essence of Hypertext and the HTML language used for creating pages on the World Wide Web. In a Web document a hyperlink can be a sequence of letters or an image. By clicking on the area designated as a hyperlink by the person who created the Web page, it is possible to jump quickly to another part of the page, a different page on the same website, or to a completely different website.

### *Hypertext*

A system for the non-sequential presentation of text, the fundamental concept of the World Wide Web, whereby the user can jump from one part of a text to another, from one Web page to another, or from one website to another, by clicking on highlighted (and usually underlined) hyperlinks. The concept of hypertext predates the Web by many years.

## I

### *iBooks*

Is een programma voor het lezen en kopen van e-books, ontwikkeld door Apple Inc voor besturingssysteem iOS. ##### iBookstore Hier is het mogelijk om boeken e-books te kopen en te downloaden. De boeken in de winkel zijn in het ePub-formaat opgeslagen. ##### Interface Is een intermediair waarmee twee systemen met elkaar communiceren en zet informatie van het ene systeem om in begrijpelijke en herkenbare informatie van een ander systeem. ##### iOS Het mobiele besturingssysteem iOS (vóór juni 2010 iPhone OS) is het besturingssysteem van de iPhone, iPad, iPod touch en Apple TV. ##### iPad



Is een in 2010 geïntroduceerde tablet-pc van het Amerikaanse elektronicabedrijf Apple. Net zoals de iPhone en de iPod touch bevat de iPad een multi-touchscherm, waardoor deze met de vingers is te bedienen

#### *Icon*

A small symbol or picture used in a Graphical User Interface (GUI). The icons on the computer screen represent programs or files, e.g. a picture of a painter's palette might represent a program used for drawing and editing pictures, and a picture of a book with a question mark on its cover might represent the text of a manual or a help file.

#### *Ink Jet Printer*

A type of Printer that fire little jets of ink at the page in order to form the characters and graphics. One of the commonest forms of printers currently in use and capable of producing high-quality output in black and white and in full colour.

#### *Interface*

An interface in computer jargon is a connection between two systems. It can be Hardware or Software. It may take the form of a plug, cable or socket, or all three, for example where a Printer or Scanner is connected to a computer, and then it's a hardware interface. There are also software interfaces that enable one program to link with another, passing across data and variables. The term interface, also known as user interface, also describes the software that is used to enable human beings to communicate with a computer.

#### *Internet*

The Internet, or simply "the Net", is a computer network connecting millions of computers all over the world. It provides communications to governments, businesses, universities, schools and homes. Any modern computer can be connected to the Internet using existing communications systems. Schools and universities normally access the Internet via their own educational networks, but private individuals usually have to take out a subscription with an Internet Service Provider (ISP). Although the Internet is in fact a network of networks, it appears to users as a network of individual computers. The Internet dates back to the group of interconnected networks that evolved from the ARPANET of the late 60's and early 70's. It has grown from a handful of interconnected networks into a huge network of millions of computers.

<<<<<< HEAD *iPad* ===== iOS

*Ipod* >>>>>> FETCH\_HEAD

#### *iPod*

The name of a portable (mobile) Media Player designed and marketed by Apple.

#### *IT(C)*

Abbreviation for Information Technology. Essentially, technology relating to information processing, i.e. computer technology, The term IT is rapidly being replaced by ICT in order to reflect the important role that information technology plays in communications by email, the Web, satellites and mobile phones.

## **J**

### *Java*

Java is a high level programming language. It is useful for creating small web applications and applets for use as part of a web page.

### *Javascript*

JavaScript is a scripting language originally developed by Netscape mainly for use as code within a web page.

Javascript is very popular for common tasks such as validating data entry forms within a web page before they are submitted to the server. Javascript can also alter the way a page looks by dynamically changing CSS format settings within its page.

Another very popular JavaScript linked technology these days is AJAX which stands for Asynchronous Javascript and XML which is used to update details on a web page without having to refresh it as you go along.

Recently a lot of interest has arisen in using non-Flash technology for interactive web pages especially with HTML 5 and to that end a very popular library called JQuery has come along that offers users a lot of interactivity without using Flash. JQuery also supports a host of specialist plug-ins to extend the library.

### *JPEG or JPG*

Abbreviation for Joint Photographic Expert Group. Pronounced “Jaypeg”. A file format used for storing images. The JPEG/JPG format uses a palette of millions of colours and is primarily intended for photographic images. The internal compression algorithm of the JPEG/JPG format, unlike the GIF format, actually throws out superfluous information, which is why JPEG/JPG files containing photographic images end up smaller than GIF files containing photographic images. If you store an image, say, of a flag containing just three colours in JPEG/JPG format it may end up bigger than a GIF file containing the same image, but not necessarily a lot bigger - it depends on the type and range of colours it contains. JPEG/JPG files containing photographic images are normally smaller than GIF files containing photographic images.

## **K**

### *KF8*

Is Amazons new Kindle Format 8 used on the Kindle Fire ##### Kindle Fire Is een tablet van het Amerikaanse bedrijf Amazon. De tablet maakt gebruik van het besturingssysteem Android 2.3. Kindle devices do not support the EPUB file format used by many other e-book readers. Instead, they are designed to use Amazon's own e-book formats: AZW and, in later devices, KF8. Like EPUB, these formats are intended for reflowable, richly formatted e-book content. ##### Kobo Maakt fysieke e-readers en applicaties voor iPad, Android, e.d. Tevens is er een Kobo winkel waar epub's gekocht kunnen worden.

## L

### *LAN*

Abbreviation for Local Area Network. A Network of computers at one site that provides services to other computers connected to it. A LAN is usually limited to an immediate area, for example the floor of a building, a single building or a campus. The most important part of a LAN is the Server that delivers software to the computers (also known as workstations or clients) that are connected to it. The server is usually the most powerful computer in the network. Users of computers connected to a LAN can access their own files remotely and exchange information with the server and other users connected to the network.

### *LAMP*

Is een acroniem voor een set van gratis vrije softwarepakketten, die vaak samen gebruikt worden om dynamische websites te draaien:

- Linux, het besturingssysteem;
- Apache HTTP Server, de webserver;
- MariaDB of MySQL, het databasemanagementsysteem (of databaseserver); of in plaats daarvan PostgreSQL (LAPP) of SQLite (LASP)
- P for PHP, Perl, en/of Python, de programmeertaal ##### Linuxkernel Is de kernel (kern) van besturingssystemen uit de Linuxfamilie.

### *Latex*

### *LCD*

Abbreviation for Liquid Crystal Display. A technology used for producing a type of flat panel computer Display Screen<sup>1</sup>, which is replacing the older type of Cathode Ray Tube display screen. A more advanced form of technology for producing flat panel display screens is known as TFT (Thin Film Transistor). LCD and TFT screens are also used in digital cameras and camcorders.

### *Linux*

Unix-type Operating System. Linux was originally created by Linus Torvalds with the assistance of developers around the world. The Source Code for Linux is open source and hence freely available to everyone.

## M

### *Macro*

A sort of mini-program that can be incorporated into other programs, comprising a series of keystrokes that you may wish to use over and over again. A macro is a series of commands and instructions that you group together as a single command to accomplish a task automatically. Instead of manually performing a series of time-consuming, repetitive actions, you can create and run a single macro - in effect, a custom command that carries out the task for you. A macro can be saved and called up whenever you need it.

### *Mark-up Language*

### *Mashup*

A *mashup* is a Web page that brings together data from two or more Web services and combines the data into a new application with added functionality.

### *Metadata*

### *MIDI*

Abbreviation for Musical Instrument Digital Interface. A format for synthesised music. Music in MIDI format is created and played through the use of synthesisers, unlike “real” music which is normally recorded in MP31, WMA2 or WAV3 file format.

### *Mobi*

(.mobi, ook wel dotMobi genoemd) is een topleveldomein (TLD). Het hoofddoel van deze TLD is om websites voor mobiele telefoons beter te kunnen onderscheiden. Een TLD is het meest rechtse gedeelte in een internetdomeinnaam, zoals .nl, .org, .gov, .brussels etc. ##### Mobipocket Is a French company incorporated in March 2000 that produces Mobipocket Reader software, an e-book reader for some personal digital assistants (PDA), wireless telephones and desktop operating systems.

### *Monitor*

The screen on which output from a computer is displayed. Also referred to as Display Screen.

*Monochrome* Monochrome describes paintings, drawings, design, or photographs in one color or shades of one color. So a monochromatic object or image has only colors in shades of limited colors or hues.

### *MOV*

The format for storing and playing back audio and video files using the QuickTime media player on the Apple Macintosh, but also available for the multimedia PC. Economical in terms of storage space.

#### *MP4*

Abbreviation for the MPEG-4 file format. There are two basic types of MP4: MP4 AAC (Advanced Audio Coding) and MP4 AVC (Advanced Video Coding). The MP4 AAC file format is used to store audio files in a more manageable size without affecting the quality. MP4 AAC's best known use is as the default audio format of Apple's iPhone, iPod and iTunes Media Player. The MP4 AVC file format is used to store video files in a more manageable size without affecting the quality. It is also increasingly being used for storing video on iPods and similar portable devices.

#### *MPEG or MPG*

Abbreviation for Motion Picture Expert Group. Pronounced "Empeg". A standard file format for storing movies in digital format and high-quality audio files in a variation known as MP3. Video files stored MPEG format can be recognised by the Extension.mpg or .mpeg. MP3 audio files can be recognised by the Extension .mp3. A newer file format is MP4. MP4 files that can be recognised by the Extension .mp4. See ASF, AVI7 MOV8 RM, which are alternative video file formats.

#### *Multimedia*

The integration of two or more types of information (text, images, audio, video, animation, etc.) in a single application.

## **N**

#### *Native app*

Een native applicatie is een programma dat specifiek ontwikkeld en geïmplementeerd is voor het gegeven model, microprocessor of besturingssysteem. Deze app is geschreven in de programmeertaal (Java voor Android, Objective-C in combinatie met Cocoa voor iOS) van het platform waarop de applicatie gelanceerd wordt. De meeste native applicaties kunnen worden gedownload uit de app store.

## **O**

#### *OCR*

Abbreviation for Optical Character Recognition. OCR software is used conjunction with a *scanner* to convert printed text into digital format. For example, a page from a printed book can be placed on the scanner and the OCR software will be used by the scanner to detect the individual words from which it is made up and then convert them into a form that can be stored on a computer,

### *Objective-C*

De programmeertaal waar veel OS X en iOS applicaties in geschreven worden. De taal heeft de programmeertaal C als basis. Objective-C is vergelijkbaar met het door Microsoft ontwikkelde C Sharp, ook een dialect van C. ##### Opensource Beschrijft de praktijk die in productie en ontwikkeling vrije toegang geeft tot de bronmaterialen (de source) van het eindproduct. De bekendste vorm is opensourcesoftware.

### *Open source*

*OS* Operating system

## **P**

### *PDA*

### *PDF*

An abbreviation for Portable Document Format. This is a file type created by Adobe that allows fully formatted, documents to be transmitted across the Internet and viewed on any computer that has Adobe Acrobat Reader software - a proprietary software viewing program available for free at the Adobe website: <http://www.adobe.com/uk/>. Businesses and educational institutions often use PDF-formatted files to display the original look of their brochures or for publishing a complete magazine in electronic format. Using the full Adobe Acrobat software package, it is possible to create a high-quality piece of artwork or a brochure which preserves the look of the original, complete with fonts, colours, images, and formatting. Documents in PDF format can be published on the Web without having to be converted into HTML.

### *Pixel*

A contraction of picture element. What you see on a computer Display Screen<sup>1</sup> is made up of thousands of coloured pixels or small dots, which can be set according to the user's choice to produce either low-resolution output, medium-resolution output or high-resolution output, the usual combinations of pixels across each line of the screen (horizontal pixels) and down each line of the screen (vertical pixels) being 640 x 480, 800 x 600, 1024 x 768, 1280 x 1024. Thus, the more pixels on the screen the higher the resolution (i.e. producing a finer, sharper image) and the greater the variety of colours that can be displayed.

### *Platform*

Often used as an alternative term for a computer system, including both the hardware and the software. Essentially this term describes something that is used to build something else. The term *platform-independent* - used to describe software - means that the software can be run on any computer.

### *Podcast*

A podcast is a broadcast digital audio recording, made available via the Web in a way that allows the recording to be downloaded for listening at the user's convenience. Cf. Vodcast, which is often used to describe a downloadable broadcast digital video recording.

#### *Pop-up*

A small Window that appears within a program or over the top of a Web page to deliver additional information. Pop-ups on the Web can be annoying as they are often used for unwanted advertising material.

*Print-on-Demand* Print on demand (POD) is a digital printing technology in which a book or other publication is printed in the amount one needs it. The POD model flies in the face of traditional printing, where large quantities of books are produced in initial print runs to reduce costs prior to distribution. The sales of online books and e-books facilitate POD and eliminate the need for hard copy book displays. POD has also changed the publishing industry by reducing the need for traditional publishing houses, allowing authors to self-publish at very low costs. Print on demand is sometimes called publish on demand.

#### *Program*

The American spelling is standard in computer jargon, enabling a useful distinction to be made in British English between a *computer program* and a *programme* in the sense of a *programme of study*. A talk with the title "Turning programmes into programs" (or maybe it was the other way round) was presented by a British Council officer at the annual TESOL conference in the USA in 1987 - which puzzled the American audience but made sense to the British participants.

#### *Protocol*

In Internet terminology protocol usually refers to a set of rules that define an exact format for communication between systems. For example the HTTP protocol defines the format for communication between Web browsers and Web server.

## Q

#### *QR Code*

Short for Quick Response Code. A QR Code is a two-dimensional barcode that can store a variety of different types of information, e.g. text, a website URL, a telephone number, an SMS message, an email address, an email message, contact details, information about an event, a Google Maps<sup>1</sup> location, your social media profile (Facebook, Twitter, etc), an iTunes link, a YouTube link, etc. QR Codes can be read by barcode readers and Smartphone cameras. Instead of writing down the information relating to a website URL or map location etc, you just take a photo of the QR Code.

#### *QuickTime*

Software by Apple used for viewing movies and listening to audio recordings

## **R**

### *Reflowable*

A reflowable document is a type of electronic document that can adapt its presentation to the output device.

### *Resolution*

A measure of the number of pixels or small dots displayed on a computer display screen, printer or scanner. One normally talks in terms of the quality of resolution, using the expression low-resolution, medium-resolution and high-resolution. The resolution of a computer display screen is normally expressed as two numbers representing the horizontal and vertical resolution, i.e. dots across each line of the screen and down each line of the screen: e.g. 640 x 480, 1024 x 768, etc. The resolution of a Printer is normally referred to by the number of dots per inch (dpi)- i.e. square inch.

### *RTF*

Abbreviation for Rich Text Format, an alternative way of storing a document created with a MS-Word-processor. RTF-formatted files can be moved relatively easily between different computer systems. RTF format is also recommended when transmitting an Attachment by Email as it is much safer than the Microsoft Word formats, which can harbour Word Macro viruses. RTF files preserve most of the formatting contained in DOC-formatted files.

## **S**

### *SDK*

### *Semantic Web*

The *Semantic Web* is not a new type of Web, but rather an extension of the Web whereby data available in different locations on the Web is linked together in a way that allows the user to search the Web in a more sophisticated way, e.g. by requesting information in forms such as “Tell me where I can find information about 21st-century writers who live within 50 miles of my home town”:

### *Setup Program*

A program that enables the user to set up a program or suite of programs on the computer’s hard disk. Also known as Install Program or Installation Program.

### *Shockwave Player*

Software developed by Adobe that enables Web pages containing interactive multimedia materials to be played on the Web. Such materials may contain games, product demonstrations and online learning applications.



### *Streaming*

Playing audio or video in real time from a website. In order to play streaming multimedia files you need a specific Plug-in program that links in with your Browser and plays the file as it is transmitted rather than downloading it to your computer first. Streaming requires a Broadband connection to the Internet since multimedia files are not stored on your computer but played in a continuous stream direct from the computer where they are stored.

## **T**

### *Tablet Computer*

A tablet computer is compact portable computer that makes use of a Touchscreen instead of a keyboard for typing and running applications. Apple's iPad is a typical example of a tablet computer.

### *Tag*

Tags are small alpha-numeric indicators around a word or part of text to define the role and/or function of that text. It is an essential tool in XML and HTML.

Tagging has become more common in recent years as a result of the widespread use of Social Media for sharing images, audio recordings, video recordings, website references, etc. Tags are labels that briefly describe the what the media or references are all about and help other people find them quickly.

### *TCP/IP*

Abbreviation for Transfer Control Protocol / Internet Protocol. The main data transfer protocol used on the Internet.

### *Tex*

### *Text editor*

A computer program that allow the manipulation of text.

*Tumblog* A tumblog is similar to a blog. Whereas the full name for a blog is "Web log," the full name for a tumblog is a "tumble log." It is named *tumble* because it is designed for posting quick snippets instead of long articles.

## **U**

### *Unicode*

The Unicode Worldwide Character Standard is a character coding system designed to support the interchange, processing, and display of the written texts of the diverse languages of the modern world. In addition, it supports classical and historical texts of many written languages: <http://www.unicode.org>.

## *Unix*

An Operating System widely used on large computer systems in corporations and universities, on which many Web servers are hosted. A PC version of Unix, called Linux, is becoming increasingly popular as an alternative to Windows.

## **V**

### *Vector Graphic*

A method of creating graphic images on a computer by telling it to draw lines in particular positions. An advantage of a *vector graphic* is that it can be enlarged or reduced in size without loss of sharpness or distortion. Most modern image creation and editing packages can save images in vector graphic format. Vector graphics can be contrasted with *bit-mapped graphics*, which are made of a fixed number of pixels (small dots), and therefore sharpness may be lost when the image is resized.

### *Vodcast*

A contraction of Video Podcast. A type of Podcast1 that incorporates video as well as audio.

## **W**

### *WAV*

Short for Waveform Audio Format. A format for storing high-quality audio files.

### *Word-processor*

Probably the most widely used computer Application. Modern word-processors allow the user to create fine-looking documents including images, tables, photographs, and even sound and video recordings if they are to be viewed on screen rather than from the printed page. In many respects they are similar to *Desktop Publishing* applications. Word-processors normally include a spellchecker, a grammar checker, a style checker and a thesaurus, as well as tools for writing in HTML, the coding language used for producing Web pages. Word-processors have been widely used in teaching and learning foreign languages ever since they first appeared.

### *World Wide Web*

Usually referred to simply as the Web. This is the most powerful and fastest growing Internet service. The World Wide Web was the brainchild of Tim Berners-Lee, who in 1989 invented the HTML coding language that is the basis of the Web. The Web became a public service in 1993. It is a huge collection of resources of information, including learning materials, which is accessed by

means of a computer program known as a Browser. The World Wide Web is only part of the Internet, but many people treat both terms as synonyms.

#### *WYSIWYG*

Acronym for What You See Is What You Get, dating back to the pre-Windows and pre-Mac period, when what you saw on the screen, e.g. in a Word document, was not necessarily what appeared on your Printer - something we now take for granted.

## **X**

#### *XML*

Abbreviation for eXtensible Markup Language. XML is a specification emanating from the World Wide Web Consortium (W3C) that allows Web designers to create their own language for displaying documents on the Web. XML is an extension to the standard language for creating Web pages, HTML, and makes it possible to create websites containing more complex interactivity.

#### *XHTML*

(Extensible Hypertext Markup Language) is een opmaaktaal voor vooral websites, die de functionaliteit heeft van HTML, maar een striktere syntaxis waardoor deze gemakkelijker verwerkt kunnen worden door een XML-parser. ##### XML XML is a generalized Markup up Language for the exchange of information. It is generalized in that it allows users to define their own tags and thus there is a data definition table required to decode the data.

## **Y**

## **Z**

#### *Zebras*

Zebras are several species of African<sup>20</sup> equids united by their distinctive black and white stripes. Their stripes come in different patterns, unique to each individual. They are generally social animals that live in small harems to large herds. Unlike their closest relatives, horses and donkeys, zebras have never been truly domesticated.

## 12 Software Catalogue

### Pre-existing software

pandoc

git

github

- Open Source
- close-source Freeware
- Proprietary

### Self-written software

## Appendix

Keyword index

Bibliography

Link list

Instructions used in the preparation of this document

[Instructions for placing images](#)