EXPERIENCES FROM IMPLEMENTING PDF IN OPEN SOURCE: CHALLENGES AND OPPORTUNITIES FOR STANDARDISATION PROCESSES

Jonas Gamalielsson and **Björn Lundell**
UNIVERSITY OF SKÖVDE

This paper presents novel results concerning specifications of standards and their implementations in open source software. Specifically, our analysis draws from rich insights and experiences related to two open source projects implementing the specification of the PDF format. The study reports on a number of issues, including: lack of clarity in the specification; implementations deviate from specification; licensing and patent issues; and influences between the specification of a standard and its implementations in software systems. Our findings present rich insights from current practice concerning challenges and opportunities for implementing specifications of standards in open source projects, and constitute an important contribution to enhanced standardisation processes.

Introduction

This research addresses standards which can be implemented in software systems, and consider challenges and opportunities concerning their implementation in open source. Specifically, we address the PDF file format which has been standardised by ISO, and investigate experiences from contributors to open source projects with significant experience from implementation of PDF and other document formats. Open Source Software (OSS) is software that is provided under a software license recognised by the Open Source Initiative (OSI, 2013).

There are a number of challenges related to provision of standards in the software sector, which can impact on the extent to which it is possible to faithfully implement the specification of a standard in software systems (UK, 2012). A number of challenges related to implementation of specifications of standards have been identified in the literature, including challenges related to: interoperability (Bird, 1998; Ghosh, 2005; Krechmer, 2005), conformance to the specification of the standard (Egyedi, 2007), and long term availability of software systems which implement specific standards and associated digital artefacts (Behlendorf, 2009; Lundell et al., 2011). Implementations of standards in OSS is one means to address these challenges, and such implementations need to be available over very long life-cycles. For this reason it is important to assess the health and longevity of open source projects (Crowston and Howison, 2006).

PDF is one of the most commonly used document formats and is widely deployed in different types of innovative OSS projects including: PDF viewers (e.g. Evince), Web browsers (e.g. Mozilla Firefox), Office suites (e.g. LibreOffice), and Business intelligence systems (e.g. Pentaho). For the PDF format, issues related to long-term availability of files created in the format are of particular importance. Hence, availability of software systems which can maintain files created in the PDF format over the full life-cycle of the files, is critical.

Previous research related to implementation of standards in software systems include studies addressing different aspects of compliance and interoperability (e.g. Egyedi, 2007; Egyedi and Dahanayake, 2003; Friedrich, 2011) and licensing conditions for standards and their implementations in open source (Ghosh, 2005; Simcoe, 2006; Friedrich, 2011). However, there is a need for further research focusing on specific implementations of specifications of standards and the relationship between specifications of standards and associated implementations, and in particular related to open source implementations. In fact, openness of standards and their implementation in open source has been elaborated more than a decade ago (Krechmer, 2002) and the relationship between standards and their implementation in open source continues to be an issue for ongoing discussion (Krechmer, 2007; Friedrich, 2011, 2013; FRAND, 2012; EU, 2012; Brock, 2013). To the best of our knowledge, this paper presents the first in-depth study on open source implementations of the PDF format.

Our ***overarching goal*** for this study is to investigate open source licensed projects implementing the PDF format. The paper makes three principle contributions. ***First***, we present a characterisation of the longevity of widely deployed open source projects that implement the PDF format. ***Second***, we identify issues raised by contributors to the open source projects that implement the PDF format concerning the specification of the

format as documented and the format as implemented. ***Third***, we report rich insights from experts with experience from implementing PDF and other document formats in open source projects. From this, the paper presents a synthesis of insights concerning the relationship and influences between the specification of the PDF format as documented and the PDF format as implemented in OSS.

The rest of this paper is organised as follows. We present a background on the PDF format and its implementation in open source and our research approach. Thereafter we present results (sections "Characterisation of PDF implementations", "Issues concerning implementations of PDF", and "Experiences from development of document format implementations"), followed by discussion and conclusion.

Background

PDF is a document format that initially was maintained by Adobe. The PDF specification is available free of charge since 1993. PDF version 7 was released in November 2006 (Adobe, 2006), and it was announced in January 2007 that Adobe together with the Association for Information and Image Management (AIIM) and ISO's Technical Committee (TC) 171 will work on making this version of PDF an ISO standard (Adobe, 2007). A ballot prepared by Adobe (with support of AIIM) commenced in July 2007 by converting the PDF version 7 specification into a Draft International Standard (DIS) and submitting it to the standardisation organisations in the different countries that are members of ISO TC 171 (King, 2007). Approval was sought through the "fast-track procedure" since PDF at the time was an existing de facto standard (ISO, 2007; King, 2007). The ballot ended in December 2007 with an overwhelming majority of the votes (13 against 1) for the approval of PDF version 7 for ISO standardisation (Infoworld, 2007). In July 2008 the PDF specification became available as an ISO standard (ISO 32000-1:2008) (ISO, 2008a, 2008b), and it is provided under the following conditions: "In association with the adoption of PDF, version 1.7 as an ISO standard (ISO 32000-1:2008), Adobe issued a Public Patent License[1], granting 'every individual and organization in the world the royalty-free right, under all Essential Claims that Adobe owns, to make, have made, use, sell, import and distribute Compliant Implementations.'" (digitalpreservation.org, 2010). It has also been claimed that "ISO 32000 is equivalent to Adobe's PDF 1.7" (Adobe, 2013) and that it only differs in terms of ISO specific text and removal of Adobe related dependencies. Further, extensions to the PDF standard (ISO 32000-1:2008) are also provided by Adobe that specify "extended features for PDF, beyond ISO 32000-1" (Adobe, 2013). Since the publication of the PDF standard (ISO, 2008a, 2008b) specific versions for specific purposes, for example the PDF/A standard for archiving purposes, have been developed and standardised by ISO (2005a, 2005b).

Over the years, the PDF format has been implemented in a number of different software systems provided by many different organisations (including commercial companies, such as Adobe, and community driven open source projects, such as Poppler). An inherent characteristic of OSS is that anyone who has adopted such software has the right to freely read, use, improve, and re-distribute the source code for such software. Over the last decade, many professionals and volunteers with strong community values inherent to OSS cultures have contributed to many open source projects in different contexts. Such software is used in many companies and public sector contexts (Brock, 2013; Fitzgerald, 2006; Lundell and van der Linden, 2013). As the PDF format is provided under license conditions which allow for implementation in open source, it is perhaps not surprising that there are a number of different open source projects which implement PDF.

There are many open source licenses under which OSS can be provided (Bain, 2012; Brock, 2013; Engelfriet, 2010; Rosen, 2004), and licenses are often broadly categorised as either copyleft licenses (e.g. GPL and LGPL) or permissive licenses (e.g. BSD and MIT). The main difference between these two license categories is that copyleft licenses ensure that derivative work remains open source, whereas permissive licenses do not (Brock, 2013). The majority of all open source projects use copyleft licenses and the most commonly used license is the GPL (Bain, 2012). This license, with its origin in the free software movement (FSF, 2013), is recognised by OSI (2013) and it has been broadly used by many open source projects, including those which have attracted significant commercial interest (Fitzgerald, 2006; ITEA, 2004; Lundell and van der Linden, 2013). In addition to the most commonly used open source license (GPL version 2), there are also other copyleft licenses in the GPL family (e.g. GPL, AGPL, LGPL) recognised by OSI (Bain, 2012; Brock, 2013; Engelfriet, 2010; Rosen, 2004). For widely deployed implementations of the PDF format, we note that several different copyleft licenses have been used when releasing OSS, including: AGPL (e.g. iText version 5 or later), GPL (e.g. Poppler) and LGPL (e.g. iText before version 5).

An inherent characteristic of open source projects is that such promote an open collaboration between individuals representing companies and other types of organisations. With the inherent transparency stemming from the open collaboration in OSS projects, important details concerning the precise interpretation of a specification of a standard becomes transparent when implemented in OSS. The open source project can thereby aid in promoting faithful interpretations of specifications of standards and thereby constitute a valuable

[1] http://solutionpartners.adobe.com/pdf/pdfs/ISO32000-1PublicPatentLicense.pdf

additional resource for the ongoing development and maintenance of standards. From this, over time the quality of specifications of standards in the software domain can improve. In fact, a number of standards have been implemented in various open source projects, and sometimes such implementations have even been instrumental in promoting the standard itself (Behlendorf, 2009).

Research Approach

To ***characterise PDF implementations***, we analysed the longevity of widely deployed open source projects implementing PDF. Specifically, the iText (Itextpdf.com, 2013) and Poppler (Poppler.org, 2013) projects were chosen. Those projects are amongst the most widely deployed open source libraries for creation (iText) and rendering (Poppler) of PDF files. Both libraries are adopted by many other applications in need of PDF functionality. In our analysis we specifically focused on the release history, number of commits and committers over time, and the proportion of commits for the most influential committers. The analysis covers the time window from the first commit in each project (iText: November 2000, Poppler: March 2005) until 31 March 2013, and data was collected from the project web pages at Ohloh (Ohloh.net, 2013a, 2013b), which is a web site that provides statistics about the longevity of OSS projects.

To ***identify issues concerning implementations of PDF***, we analysed mailing lists for the two projects, which constitute one of the most important channels for communication in the two chosen open source projects. More specifically, the official mailing lists for iText (Gmane.org, 2013) and Poppler (Freedesktop.org, 2013) were downloaded and interpreted holistically. Thereafter, keyword search was used as a basis for further analysis supplemented by manual inspection of statements in individual messages and threads of communication. The analysis covers the time window from the start of each mailing list (iText: June 2002, Poppler: March 2005) until 31 March 2013. We specifically focused on statements from 2006 and later. The vast majority of these statements were made when PDF version 1.7 (the version released as the ISO standard 32000-1 in July 2008) had been released, and a few of the statements were made close to its release.

To ***report on experiences from development of document format implementations***, the two researchers conducted interviews with technical experts with long-term experience of implementing PDF and other document formats in OSS as respondents. Data collection was based on the results of face-to-face interviews (February 2013, i.e. almost five years after the release of the ISO 32000-1 standard) conducted in English. Interviews were recorded, transcribed, and vetted by each interviewee. Questions were prepared in advance, and shown to the interviewee before the conduction of the interview. During each interview, follow-up questions were used in the dialogue. Each interview was conducted in an informal setting and allowed each interviewee to extensively elaborate on all issues covered during the interview. A total of 5 interviews were conducted, ranging in time from 6 to 48 minutes and resulting in 29 pages of transcribed and vetted interview data. In this process each interviewee was allowed to further elaborate and clarify their responses.

Analysis of the transcribed interview data took place within a few weeks after data collection. Individual analysis was supplemented by group sessions in which researchers discussed and reflected on the interpretations from each researcher. The coding of interview data was conducted in a manner which follows Glaser's ideas on open coding (Lings and Lundell, 2005). The unit of coding was sentences or paragraphs within interview notes. The focus was on constant comparison: indicator to indicator, indicator to emerging concepts and categories (Lings and Lundell, 2005). The goal of the analysis was to develop and refine abstract concepts, which are grounded in data from the field (as interpreted via collected data in the transcriptions). The coding process resulted in a set of categories, each presented as a sub-section in the section "Experiences from development of document format implementations" of this paper. These categories were also used as a structure in the section "Issues concerning implementations of PDF" (for presentation of results from the analysis of mailing lists for the two projects).

Characterisation of PDF implementations

This section provides a characterisation of the longevity of iText and Poppler, which are two open source licensed and widely deployed implementations of PDF.

The iText library for PDF generation is written mainly in Java (Itextpdf.com, 2013), and was initially provided under the MPL v1 and LGPL v2 licenses. However, the license was changed to the AGPL v3 license on 5 Dec. 2009 with the release of version 5.0.0. The library is widely adopted in applications that include functionality for creation of PDF documents. There have been 25 committers who have contributed a total of 10661 commits over 594093 lines of code (Ohloh.net, 2013a). The first commit was contributed in Nov. 2000, and the most recent commit in Mar. 2013. There have been 4 first level iText releases in the interval Feb. 2001 through Mar. 2013 (v0.x in 2000, v1 in 2003, v2 in 2007, and v5 in 2009). There have been 99 releases (evenly distributed in time) in total since v0.30 including second and third level releases. The latest release (version 5.4.0) was made available on 14 Feb. 2013.

The Poppler library for rendering of PDF documents is written mainly in C++ (Poppler.org, 2013). Poppler was initially provided under the GPL v2 license, but has been provided under GPL v2 or later since Aug. 2011. The library is widely adopted in applications that include functionality for rendering of PDF documents. There have been 156 committers who have contributed a total of 3820 commits over 164578 lines of code (Ohloh.net, 2013b). The first commit was contributed in Mar. 2005, and the most recent commit in Mar. 2013. There have been 110 Poppler releases (evenly distributed in time) in the interval Mar. 2005 through Mar. 2013 including second and third level releases. The latest release (version 0.22.1) was made available on 10 Feb. 2013.

Figure 1 illustrates the number of active committers during each month in the iText project (blue trace) and in the Poppler project (red trace). We note that there has been a long-term and continuous activity in both iText and Poppler since the start of the projects, and that there are periods of shifting activity. We also observe that peaks often co-occur with events in the projects like major releases. For example, the peak in Dec. 2009 for iText co-occurs with a major release (version 5.0.0), and the peak in Sep. 2011 for Poppler co-occurs with a major release (version 0.18.0).

Figure 2 shows the number of active committers each month in the iText project (blue trace) and the Poppler project (red trace). We note that the number of monthly committers is generally higher for Poppler from Jul. 2007 and onwards. There is also a long-term trend over time towards increased participation in the two projects. The peaks in participation are also more distinct for Poppler. Like in Figure 1, peaks often co-occur with events like new major releases in the projects.

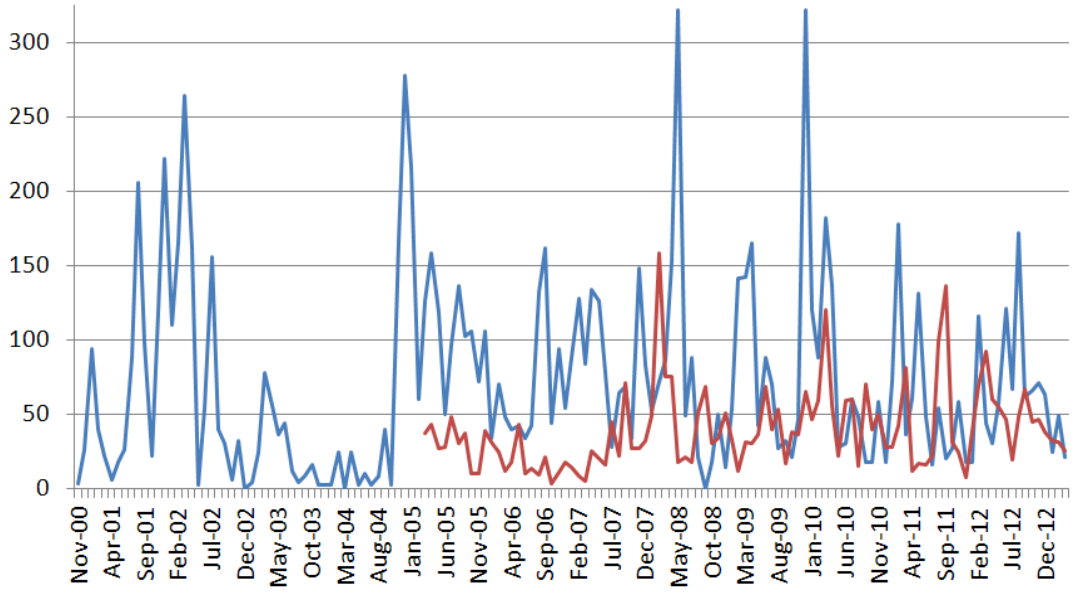

Figure 1: *Number of monthly commits for iText (blue trace) and Poppler (red trace).*

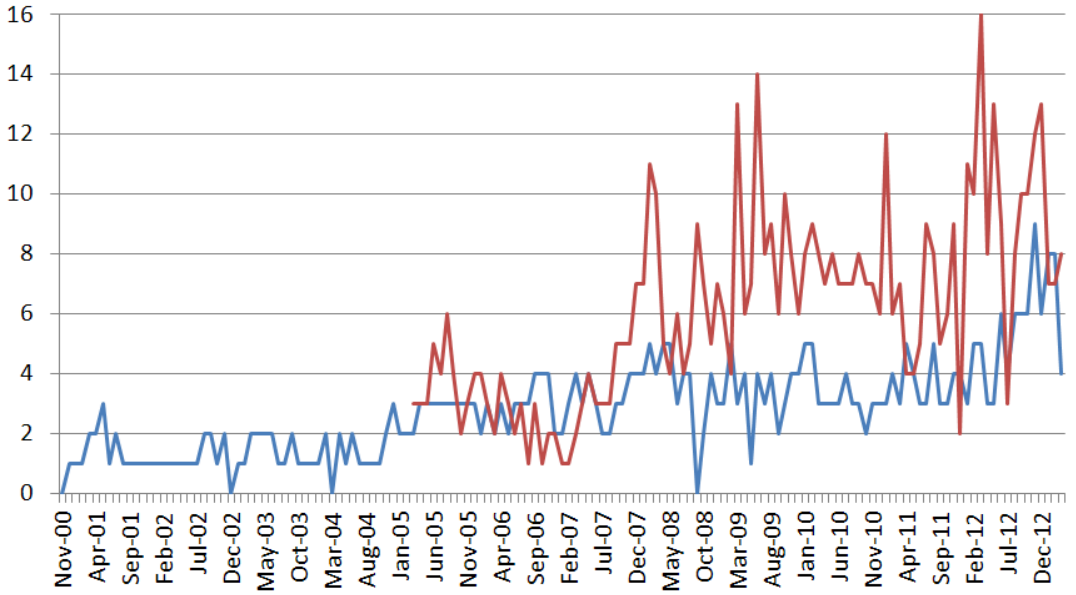

Figure 2: *Number of monthly committers for iText (blue trace) and Poppler (red trace).*

Figure 3 shows the proportion of commits for the 8 all-time most active committers in the iText project and the remaining developers (light green colour). We note that the top committer contributes 68% of the code, the other 7 committers together contribute 30% of the commits, and remaining committers contribute 2% of the commits. Similarly, Figure 4 illustrates the proportion of commits for the 8 all-time most active committers in the Poppler project and the remaining developers (light green colour). We note that the top committer contributes 38% of the code, the other 7 committers together contribute 45% of the commits, and remaining committers contribute 17% of the commits. Hence, the top committer in Poppler is less dominant when comparing with the top committer in iText. Further, the larger proportion of commits for remaining contributors in Poppler (17%) suggests that a larger number of committers contribute significantly in Poppler compared to iText.

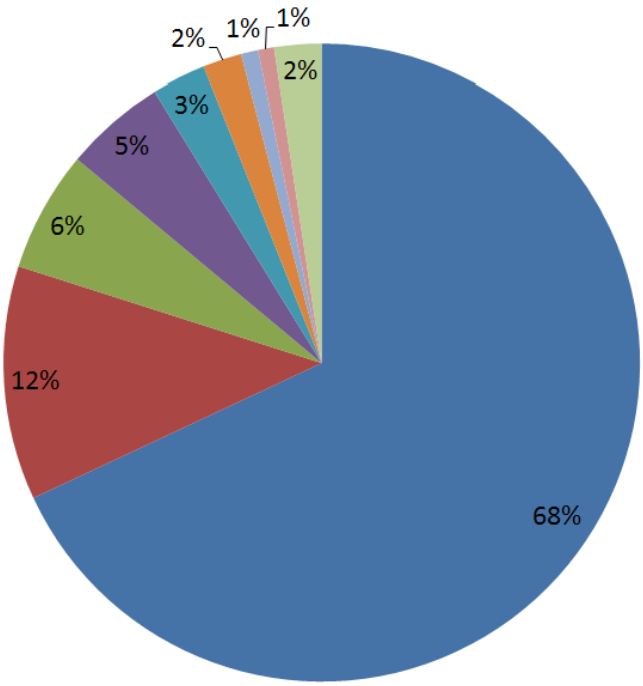

Figure 3: *Proportion of commits for the top 8 committers in iText.*

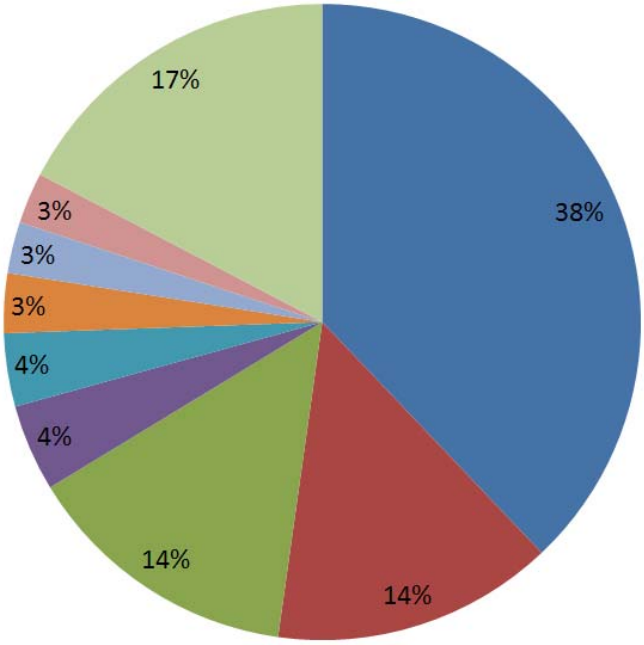

Figure 4: *Proportion of commits for the top 8 committers in Poppler.*

Issues concerning implementations of PDF

This section draws from experiences from two specific open source projects. We report on issues and comments raised by contributors to the official mailing lists in the iText and Poppler projects, and in this section we use the same categories (as sub-sections) for the presentation of results as in the next section ("Experiences from development of document format implementations").

On clarity and detail in specification

From comments by contributors to the projects it is evident that there are issues related to lack of clarity and detail in the PDF specification. For example, one contributor pointed out that optional elements in PDF files can be problematic: "PDF DOES support rich semantic structure including all of things listed below ... HOWEVER, it is optional and therefore many PDF documents do not contain the necessary elements. And, as pointed out, without the presence of such elements already in the PDF - the best you can do is GUESS". Further, lack of clarity concerning character mapping and encoding has been stressed by another contributor: "When I looked at the PDF / CMap specs, it was unclear whether the one-byte characters were allowed there or not. Since it was pretty easy to accept one-byte characters, and since it doesn't really hurt anything, and since Acrobat seems to do it too, I decided to change my CMap parser."

Ambiguity and vagueness in the PDF specification is a recurrent theme amongst contributors. For example, one contributor commented that "if you ignore the cross reference tables or streams, interpretation of where objects start in a PDF can be ambiguous". Another contributor expressed concerns regarding vagueness in the PDF specification when it comes to support for image decoding: "I've looked at the PDF spec, and it seems a little vague on this point". Similarly, a different contributor commented on the lack of clarity concerning how to define the output intent, e.g. PDF/X compliance, according to the PDF specification: "I read the PDF spec but it is vague in that it says one does not have to add an output ICC profile if the given output intent is an 'industry standard' or something like that (which ISOcoated_v2 sure is), but there is not a list of 'registered names' for output intents". A similar view has been expressed concerning signature validation: "After reading a lot a documents, from PDF-spec to technical recommendations, I feel that the signature validation subject is still unclear". On the same topic, a different contributor noted that the ISO 32000-1 specification described that all fields need to be locked in a PDF document in order to sign it, but not how fields are locked: "Unfortunately I cannot find any explanation there what locking a field means".

Implementation deviates from specification

From statements in the project communities it is evident that implementations of PDF deviate from the specification in different ways. For example, it has been stated that "PDF has some features that are really exotic, and that are probably hardly ever used" and that "even Adobe Reader doesn't support everything that is in the PDF specs". Another contributor also commented: "I would argue that 99% of the pdfs in existence use 10% of the specification." Further, there are statements indicating that implementations of PDF are designed to handle files that deviate from the specification, for example: "But we also know that Adobe Reader is quite forgiving regarding off-spec/damaged PDFs". On the other hand, it has been stated that there are features not present in the specification which for that reason are not implemented: "Elliptic Curves are not supported by Adobe Reader X, as they are not documented to be supported in ISO 32000-1:2008."

There are extensions to the specification of PDF that expand the scope of the standard. Further, there are statements indicating that PDF extensions are planned for inclusion in future versions of the PDF specification: "Though, you should know that the Adobe extension in question (AES-256) has been approved by the ISO 32000 committee for inclusion in the next version of ISO PDF (32000-2, aka PDF 2.0)" and "The specs of what is implemented in Acrobat/Reader today can be found on the Adobe website, HOWEVER, as it was introduced to ISO a few changes were made so the final ISO 32000-2 will have the revisions". There are also details concerning PDF generation that are only available through specific implementations and not in published specifications or extensions to specifications, as illustrated by the following comments: "You can CREATE the PDF with iText - no problem there. HOWEVER, in order to have it enabled for saving with Adobe Reader, you need to add some 'secret sauce' to the PDF. The only server-based product that can do this is Adobe LiveCycle Reader Extensions Server. This is because the only Adobe knows the secret of the sauce." and "This model is employed by other vendors as well which have their PDFs have basic functionality in their free products and add extra value for their commercial solutions".

Influences between implementations and specification

Analysis shows that different implementations of a specification of PDF can influence each other. For example, the features of one implementation could be *added* in another implementation, as commented by one contributor: "It is unfortunate that only Adobe's tools correctly support Tagged PDF and use those features to provide richer semantic extraction of PDF content. I would love to see someone add such support to Poppler."

Similarly, another contributor stated that “I’m planning to add HTTP streaming support to poppler. Similar to the way Adobe Reader does it.”

A more elaborate type of influence between implementations is when the behaviour of one implementation is *mimicked* in another implementation. This is exemplified in a view expressed by a contributor concerning creation of watermarks: “You might want to use Adobe Acrobat to create such a watermark annotation, analyze the PDF generated, and emulate that process in iText.” In a different discussion there was the question “Is there something in the itext api to mimic the autosize feature that Adobe Acrobat provides for form fields?”, which was answered with “Yes. If you set the fontsize to 0, the font-size will be adjusted automatically so that the text fits the text field.”

The behaviour of different implementations can also be *compared* as illustrated by the following comment by a contributor discussing page rendering: “can’t tell if better or worse, we render different than Adobe Reader”.

Another observation is that implementations of a specification (apart from influencing each other) also can influence the specification, which is illustrated by the following statement: “I could be remembering it wrong, but I think that many years ago the ghostscript developers said that they found many inconsistencies in Adobe’s documentation, and when they weren’t sure what to do, they looked at what Adobe products did. Many times when they informed Adobe of the inconsistency or of something that Adobe’s products did differently than the specifications, Adobe would revise the specifications to match what their products did rather than changing their products to match the specification.”

Deployment of open source implementations of specification

Open source contributors express the view that the PDF specification is provided under conditions which allow OSS implementations under the GPL. For example, as expressed by one contributor: “the PDF license is GPL compliant”. Further, it is evident from discussions that some contributors express some concern for potential risks for patent infringements related to the PDF format.

Discussions also address how to avoid patent issues when trying to avoid problems associated with patents when implementing normal rendering. In such discussions contributors raise an open question on how other OSS implementations can achieve the desired rendering.

The relationship between patent issues and the choice of open source license is complex and impact on implications when distributing OSS. In the view of one contributor: “We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary.”

Discussions related to use of fonts in PDF and associated legal issues have raised the concern for the need for legal advice on these issues, which lead to some frustration amongst contributors. Further, discussions amongst contributors have raised issues concerning the extent to which it is possible to use and distribute copyrighted fonts, as illustrated by the following comment: “Can you confirm that Adobe has allowed the distribution of these copyrighted fonts?”

Views are also expressed that the ISO standardisation process and the position of Adobe related to the PDF format should not cause implementers to vary, as elaborated by one contributor: “As an ISO standard, PDF was vetted according to the ISO rules for such things and Adobe made sure all of our patents in this area were released accordingly.”

The necessity of re-licensing of an OSS implementation of PDF has been raised as an issue: “Keeping iText a free library has become an almost daily struggle as pressure from sales grows. I don’t think I can keep a license change away much longer. The MPL/LGPL will probably be dropped and replaced by the AGPL.”

Experiences from development of document format implementations

This section draws from rich insights from experts with long-term experience from implementing the PDF format (and other document formats) in several open source projects. Four broad categories emerged from our analysis of experiences of respondents. Each is presented as a separate subsection below, with a subheading aimed to characterise the category.

On clarity and detail in specification

We find that there are details lacking in the PDF specification that make it uncertain how to interpret PDF files. For example, one respondent stated that “the standard doesn’t say which encoding is to be used” and that “you get some bytes and you don’t know how to interpret them, so you really have to guess”. Further, respondents commented on problems associated with uninstantiated attributes and handling of incorrectly generated PDF files. For example, one respondent expressed that such problems can imply that “you get a PDF which has colour but doesn’t have size, because someone created the PDF wrong”.

Another respondent commented that the PDF standard has evolved for a long time, and that there have emerged multiple ways of implementing the same feature according to the specification, which adds complexity. This makes it more challenging to develop an implementation. Concerning embedding of video in a PDF file, the same respondent expressed that you "can do it like in four different ways, right, and that's not good". Another aspect adding complexity is that there has evolved a variety of different font formats in the PDF specification. As put by one respondent: "we have to support Type 3 format, Type 3 fonts, Type 1 fonts, embedded fonts, not embedded fonts, TrueType fonts, OpenType fonts, lots of different fonts, right."

Views also indicate that a standard can be too complex, which can be an inhibitor to implementation. For example, one respondent highlighted the limited use of the complex Open Systems Interconnection (OSI) model standard in software implementations and that it "was used in university for teaching, but down there on the road people just used TCP/IP". Further, another respondent commented that "the problem is that standards are there in the void" and that those involved in standardisation often are "people that don't really have much coding or programming experience". Similarly, another respondent expressed a view that "standards are written like in vacuum". Comments concerning difficulties in interpreting text in the PDF specification were also raised: "sometimes the English is a bit, I don't know, cumbersome to read".

Implementation deviates from specification

Analysis of responses show that implementations of the PDF specification are often perceived to deviate from the specification for different reasons. As part of this, several respondents explained that the PDF standard is extensive and that there are plenty of features in the specification of the standard that deliberately never get implemented in OSS. As commented by one respondent: "The problem with the PDF standard is like it's huge, right. So, we can't really implement it all". In the experience of another respondent, an open source project often "attempts to follow the standard as best as possible. In practice, this isn't always possible though because some features are very difficult to implement or viewed as unnecessary". According to another respondent, a feature in a specification may have been "a good idea when it was created, but there was no business use for it". Related to this, another respondent stated that "which features we actually implement is kind of just driven by how often we see it come up in the real world" and that "it's just what people need is what we add". Similarly, respondents also elaborated on existence of features in the PDF specification that a limited number of users have interest in (e.g. support for 3D models and Javascript) and that are therefore not yet implemented.

We also found that users do not accept that a PDF file is not rendered by a specific PDF implementation due to non-conformance with the specification, especially when the file can be rendered in another PDF implementation. In such cases users tend to blame the PDF rendering implementation, as illustrated by the following comment: "All they care is that they have a file and their file is being shown in one of the PDF readers and it is not shown in your PDF viewer, so they see the problem not in the file but in your software." The same respondent stressed that the PDF specification only dictates what is going to happen under correct conditions, and not when a PDF file contains errors or is incomplete. Some implementations go beyond the specification in order to deal with imperfect PDF files, and such implementations can influence other implementations. As expressed by one respondent: "So, then you have to like resort to open the thing in Adobe Reader and do what they do, which is ok but sometimes it is not that easy because it is hard." Further, some respondents expressed frustration over incorrectly generated PDF files, as stated by one respondent: "I have encountered a lot of PDFs in the world that weren't built according to the standards, and that is really annoying."

Influences between implementations and specification

From responses we find that there are different kinds of influences between implementations of specifications and specifications of standards.

We find that the communities involved in the implementation of a specification influence each other through different feedback processes. One respondent active in the community of an OSS project implementing PDF explained that they do not directly participate in the standardisation of PDF, but that they have good contact with individuals in a company that participates in the PDF standardisation committee. Hence, the community can through these contacts influence the PDF standardisation process and also be influenced by the same process. Another respondent, active in the community of another project implementing PDF is also a member of a national standardisation bureau, and the project can in this context both influence standardisation and get influenced by standardisation processes. Specifically, the respondent identifies this as an opportunity to improve the clarity of the specification: "I already have some comments that are no part of the standard, so I think it's kind of contributing back, but also in my own interest because the more clarity there is in the standard the better you can justify why you have implemented something like that".

Respondents elaborated on that the specification of a standard influences the implementation of the specification. As commented by one respondent: "of course those standards very much influence the open

source software because it's following the standards". Similarly, another respondent stated: "many open source projects are based around standards, so they are deeply influenced by the standard".

From responses we found that implementations of a standard can influence the specification of a standard. For example, one respondent commented that an open source project "can choose how to interpret a standard and what features it implements it can indirectly alter the standard if the project becomes popular or other projects also choose to follow the same approach. If there is a healthy relationship between the implementors and the standard bodies then these changes can be addressed in the standard by either adding or removing features or redefining features."

Similarly, another respondent stressed the potential for open source implementations to influence and set standards: "open source is able to set de facto standards, which then later on just get documented properly and get someone's standard approval". The respondent specifically mentions web standards as a type of standard where "people implement something and then go and talk to others and fix their implementation and then eventually come up with a standard."

Further, another respondent elaborated on several benefits of having implementations of specifications driving the standardisation: "I prefer that approach to things because it makes at least sure that what eventually becomes a standard makes sense, is technically feasible, implementable, reasonably interoperable."

Deployment of open source implementations of specification

Issues concerning deployment of open source implementations were identified amongst respondents. Specifically, issues concern conditions under which implementations and specifications are provided.

We found that licensing of code in open source projects can be an issue. For example, one respondent explained that there was a need to change the license in a project to a dual licensing model involving a stronger copyleft license since a large company "made money with it, but they didn't want to support me". The respondent also commented that "if you don't have any form of revenue it is very hard to maintain a project" and that the license was made "more viral in the sense that there is more reason for companies, if they don't want to disclose their source, they have to buy a license". Further, another respondent active in an OSS project implementing PDF explained that a new similar OSS project implementing PDF was started mainly due to the licensing conditions: "the main point was that we were GPL 2 only and they wanted a GPL 3 implementation" and "we couldn't change from GPL 2 to 3 because we had some other internal licensing problems". Similarly, it was also commented that a large company approached an OSS project implementing PDF due to incompatible software licenses: "they asked me change the license of our code, because it was not compatible with their code".

Several respondents stressed the importance of providing standards under royalty free conditions and that standards are not encumbered by patents, and that such standards should be implemented in open source. For example, one respondent stated that "wherever there royalty is involved, even if it's reasonable royalties, it goes against open source. You just can't do that" and that "it's a barrier for entrance to any open source project if there is a standard that is not royalty free". Other respondents expressed that "the best standard to use is a standard which unencumbered by patents, which is well documented" and that "I'd implement first of all with standards that are not patent or copyright or like something encumbered". One respondent also commented concerning promotion of royalty based standards: "And when you as a policy maker advocate or recommend such a standard you're doing the community a disfavour, because you unilaterally favour a rather … usually a rather small number of commercial vendors". It was also commented that "standards at the very minimum should be publicly accessible by everybody".

A need to implement de-facto standards was also identified, since support for such standards are often expected by users. However, such standards can be covered by patents and may therefore be problematic to implement. It may also be of limited interest for a patent owner to pursue a lawsuit against a volunteer community, which can also result in bad-will. As commented by one respondent: "we are mostly poor. So we are not really a target for people suing us" and that patent owners "probably don't want sue small open source projects because it would be too bad for PR."

Discussion and conclusion

We find that there are multiple OSS projects implementing the PDF format. These have attracted significant contributions over many years, which is considerably longer than the PDF format has been an ISO standard. OSS projects have contributed to increased transparency and opportunities for precise inspection of how the specification of the format has been interpreted. This allows anyone to contribute to further development of the standard through participation in OSS projects.

Our results show that developers experience problems concerning the implementation of the PDF specification, something which confirms previous observations by providers involved in PDF standardisation when they started to implement a new version of the PDF specification (Gwg.org, 2012).

Analysis shows that there are several problematic issues related to clarity and detail in the specification of the PDF format. Such issues make, in many cases, the implementation of the specification unnecessarily challenging and complex. This, in turn, increases the risk that different implementations of the specification of the PDF format deviate and ultimately lead to problems related to interoperability.

We find that implementations of the specification of the PDF format may deviate from the specification. Implementations may cover a subset of the features in the specification, and also features beyond the specification. Implementing a subset of the features implies that the implementation is incomplete and therefore will not support all usage scenarios as can be anticipated from the specification of the PDF format. Implementing features beyond the specification may negatively impact on interoperability, and in particular in cases when such extensions are not documented and not openly available.

The analysis revealed that there are different kinds of influences, both between different implementations of a specification, and also between a specification and its implementations. A potential implication of influences between implementations of a specification may be that implementations of a specification are primarily influenced by the market leading implementation instead of the specification of a standard. Similarly, an implication of influences between a specification and its implementations may be that the implementation contributes to increased precision in the specification, and in particular when implementations are deployed as open source projects as such allow for scrutiny of all details in an open collaboration.

Analysis shows that there are issues related to deployment of OSS implementations of the PDF specification. Specifically, licensing and patent issues are raised as a concern by contributors and respondents. The license was changed in both open source projects to a stronger copyleft license. For one OSS project, the change to AGPL v3 was motivated by business reasons. For another project, license incompatibility issues with other OSS projects were discussed, and eventually the license for this OSS project was changed to GPL v3. These changes did not result in any significant effect on the number of contributors or contributions to the code developed in each of the projects. Despite the royalty free licensing conditions under which the PDF standard is made available, it is evident that patent related issues for software implementations is a concern amongst contributors to both OSS projects. Patent related concerns may be explained by strong community values amongst contributors, which in turn may inhibit potential contributions to an OSS project.

In conclusion, by drawing from analysis of two open source projects implementing the PDF format, the study shows supporting evidence for that: *i)* there can be a number of different problematic issues related to clarity and detail in the specification of standards; *ii)* implementations of a specification of a standard may deviate from the specification; *iii)* licensing and patent issues are perceived as a concern by contributors to open source projects implementing a specification of a standard; *iv)* there are influences between the specification of a standard and its implementations in software systems. In particular, we found how influences from two open source projects implementing the PDF format impacts on standardisation of the format, which in turn may influence other implementations of the format. From this, the paper illustrates the potential benefit and innovative ways of using open source licensed implementations of a standard as a means for an improved standardisation process through increased precision in the specification of standards.

References

Adobe (2006). PDF Reference – Adobe Portable Document Format, Version 1.7. http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/pdf_reference_1-7.pdf, Accessed 6 August 2013.

Adobe (2007). Adobe to Release PDF for Industry Standardization, http://www.adobe.com/aboutadobe/pressroom/pressreleases/200701/012907OpenPDFAIIM.html, Accessed 6 August 2013.

Adobe (2013). PDF Reference and Adobe Extensions to the PDF Specification. http://www.adobe.com/devnet/pdf/pdf_reference.html, Accessed 6 August 2013.

Bain, M. (2012). Scene Setting - Licensing models for standards and for open source. EC Workshop: Implementing FRAND standards in Open Source: mission impossible?, Brussels, Belgium, 22 November.

Behlendorf, B. (2009). How Open Source Can Still Save the World. Keynote Presentation, In 5th IFIP WG 2.13 International Conference on Open Source Systems (OSS 2009), Skövde, Sweden, 5 June.

Bird, G.B. (1998). The Business Benefit of Standards. StandardsView, 6(2), pp. 76-80.

Brock, A. (2013). Understanding Commercial Agreements With Open Source Companies. In Coughlan, S. (Eds.) Thoughts on Open Innovation - Essays on Open Innovation from leading thinkers in the field, OpenForum Europe LTD for OpenForum Academy, Brussels.

Crowston, K. and Howison, J. (2006). Assessing the Health of Open Source Communities. Computer, 39(5), pp. 89-91.

digitalpreservation.org (2010). PDF/A-1, PDF for Long-term Preservation, Use of PDF 1.4. http://www.digitalpreservation.gov/formats/fdd/fdd000125.shtml, Accessed 6 August 2013.

Egyedi, T.M. (2007). Standard-compliant, but incompatible?!. Computer Standards & Interfaces, 29(6), pp. 605-613.

Egyedi, T.M. and Dahanayake, A. (2003). Difficulties implementing standards. In Proceedings of the 3rd Conference on Standardization and Innovation in Information Technology (SIIT 2003), 22-24 October, Delft, The Netherlands, pp. 75-84.
Engelfriet, A. (2010). Choosing an Open Source License. IEEE Software, 27(1), pp. 48-49.
EU (2012). Guidelines for Public Procurement of ICT Goods and Services: SMART 2011/0044, D2 - Overview of Procurement Practices. Final Report, Europe Economics, London, 1 March, http://cordis.europa.eu/fp7/ict/ssai/docs/study-action23/d2-finalreport-29feb2012.pdf
Fitzgerald, B. (2006). The transformation of Open Source Software, MIS Quarterly, 30(4), pp. 587-598.
FRAND (2012). Implementing FRAND standards in Open Source: Business as usual or mission impossible?. EC Workshop, Report of FRAND and OS event, Brussels, Belgium, 22 November, http://ec.europa.eu/enterprise/sectors/ict/files/ict-policies/report-from-frand-os-conference-22nov12_en.pdf
Freedesktop.org (2013). The poppler Archives. http://lists.freedesktop.org/archives/poppler/, Accessed 15 April 2013.
Friedrich, J. (2011). Making innovation happen: The role of standards and openness in an innovation-friendly ecosystem. In Proceedings of the 7th International Conference on Standardization and Innovation in Information Technology (SIIT 2011), 29-30 September, Berlin, Germany, pp. 1-8.
Friedrich, J. (2013). Getting Requirements Right: Towards a nuanced approach on standardisation and IPRs. In Coughlan, S. (Eds.) Thoughts on Open Innovation - Essays on Open Innovation from leading thinkers in the field, OpenForum Europe LTD for OpenForum Academy, Brussels.
FSF (2013). The Free Software Foundation. http://www.fsf.org/, Accessed 6 August 2013.
Ghosh, R.A. (2005). Open Standards and Interoperability Report: An Economic Basis for Open Standards. FLOSSPOLS, Deliverable D4, 12 December.
Gmane.org (2013). Information about gmane.comp.java.lib.itext.general. http://dir.gmane.org/gmane.comp.java.lib.itext.general, Accessed 15 April 2013.
Gwg.org (2012). General Meeting Minutes – Ghent Workgroup. 28 August 2012, http://www.ghentworkgroup.com/wp-content/uploads/attachments/526788bd009909f8f0914d142c7cb571.pdf, Accessed 6 August 2013.
Infoworld (2007). PDF approved as international standard. http://www.infoworld.com/t/applications/pdf-approved-international-standard-725, Accessed 6 August 2013.
ISO (2005a). Document management -- Electronic document file format for long-term preservation -- Part 1: Use of PDF 1.4 (PDF/A-1). ISO/TC 171/SC 2, ISO 19005-1:2005.
ISO (2005b). New ISO standard will ensure long life for PDF documents. 7 October 2005, http://www.iso.org/iso/home/news_index/news_archive/news.htm?refid=Ref974, Accessed 6 August 2013.
ISO (2007). Draft International Standard ISO/DIS 32000. http://pdf.editme.com/files/PDFREF/ISO_DIS_32000__E_.pdf, Accessed 6 August 2013.
ISO (2008a). Document management -- Portable document format -- Part 1: PDF 1.7. ISO/TC 171/SC 2, ISO 32000-1:2008.
ISO (2008b). PDF format becomes an ISO standard. 2 July 2008, http://www.iso.org/iso/home/news_index/news_archive/news.htm?refid=Ref1141, Accessed 6 August 2013.
ITEA (2004). ITEA Report on Open Source Software. January 2004, Eindhoven: ITEA Office Association.
Itextpdf.com (2013). iText – Free/Open Source PDF Library for Java and C#. http://itextpdf.com/, Accessed 6 August 2013.
King, J.C. (2007). Inside PDF - Submission of PDF to ISO. 2 October 2007, http://blogs.adobe.com/insidepdf/2007/10/submission_of_pdf_to_iso_1.html, Accessed 6 August 2013.
Krechmer, K. (2002). Cathedrals, Libraries and Bazaars. In Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002), Madrid, Spain, 10-14 March, pp. 1053-1057.
Krechmer, K. (2005). The Meaning of Open Standards. In Proceedings of the 38th Hawaii International Conference on System Sciences – 2005, IEEE Computer Society, Los Alamitos, 10p.
Krechmer, K. (2007) Event report - The Open Standards International Symposium. Journal of IT Standards & Standardization Research, 5(2), pp. 59-62.
Lings, B. and Lundell, B. (2005). On the adaptation of Grounded Theory procedures: insights from the evolution of the *2G* method. Information Technology & People, 18(3), pp. 196-211.
Lundell, B., Gamalielsson, J. and Mattsson, A. (2011). Exploring Tool Support for Long-term Maintenance of Digital Assets: a Case Study. In Fomin, V. & Jakobs, K. (Eds.) Proceedings: 16th EURAS Annual Standardization Conference, European Academy of Standardisation, The EURAS Board, pp. 207-217.
Lundell, B. and van der Linden, F. (2013). Open Source Software as Open Innovation: Experiences from the Medical Domain. In Eriksson Lundström, J.S.Z., Wiberg, M., Hrastinski, S., Edenius, M. & Ågerfalk, P.J. (Eds.) Managing open innovation technologies, Berlin: Springer, pp. 3-16.
Ohloh.net (2013a). Ohloh – iText. http://www.ohloh.net/p/itext, Accessed 15 Apr. 2013.
Ohloh.net (2013b). Ohloh – Poppler. https://www.ohloh.net/p/poppler, Accessed 15 Apr. 2013.

OSI (2013). The Open Source Initiative. http://opensource.org/, Accessed 6 August 2013.
Poppler.org (2013). Poppler. http://poppler.freedesktop.org/, Accessed 6 August 2013.
Rosen, L. (2004). Open Source Licensing: Software Freedom and Intellectual Property Law. Upper Saddle River: Prentice Hall.
Simcoe, T.S. (2006). Open Standards and intellectual property rights. In Chesbrough, H., Vanhaverbeke, W. and West, J. (Eds.) Open Innovation researching a new paradigm, Oxford: Oxford University Press.
UK (2012). Open Standards Principles: For software interoperability, data and document formats in government IT specifications. Cabinet Office, UK, 1 November, https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/183962/Open-Standards-Principles-FINAL.pdf