Analysis and Visualization of Information Quality of Technical Documentation

Anna Wingkvist[1], Welf Löwe[1], Morgan Ericsson[2], Rüdiger Lincke[1]

[1]Linnaeus University, Växjö, Sweden

[2]Uppsala University, Uppsala, Sweden

anna.wingkvist@lnu.se

welf.lowe@lnu.se

morgan.ericsson@it.uu.se

rudiger.lincke@lnu.se

Abstract

Technical documentation has moved from printed booklets to online versions that need to be updated continuously to match product development and user demands. There is an imminent need to ensure the quality of technical documentation, i.e. information that follows a product.

Moving from printed material to online versions also allows for documentation to become active, to integrate interactive content, which blurs the boundaries between information and software. In order to assess quality of technical documentation, we adopt analyses and visualizations known from quality assessment of software. The analyses assess text copies, usage, structural properties, and the conformance of information to meta-information. The analysis results are visualized using a range of abstractions to aid in identifying and communicating quality issues to different stakeholders.

In a case study, we assessed the quality of real world technical documentations from a Swedish mobile phone vendor, a Japanese camera vendor, and a Swedish warship producer. The study showed that our analyses and visualization are applicable and can identify quality issues. For example, we tested an unclassified subset of the warship's technical documentation and found that 49% of it was redundant information.

The case study was conducted at a Swedish company that is in charge of creating and maintaining technical documentation. While our approach is limited to analysis that can be performed automatically, the company acknowledges that it has great potential and that our results proved helpful.

Keywords: Information Quality, Software Analysis, Software Visualization, Technical Documentation

1 Introduction

Technical documentation (or a user manual) often constitutes the first line of support when users need help with a problem or when they seek to advance their use of a product. The documentation is an important part of the product experience, and the users need to feel confident that it is correct. It does not matter if the product is a mobile phone or a warship. The quality of the technical documentation is determined by the users and how well it supports them, and this quality will in turn reflect on the perceived quality of the product.

Technical documentations are commonly thought of the printed booklets that accompany products. The same information is often available in an electronic version (e.g. PDF or HTML) on the company Website. Both the printed and electronic variants are created in and generated from Content Management Systems (CMS). This allows the producers to keep the documentation correct and up to date.

Many products are updated or changed over their life span; so keeping the documentation correct and up to date is a continuous process. A warship has a life span of over 30 years, and will most likely see many modifications and upgrades, and as software is a pervasive part of almost any product these days, it is feasible to offer modifications and upgrades to consumer products as well. So, there is an immediate need to continuously ensure the quality of the electronic documentation i.e. information, that follows a product.

The main purpose of the technical documentation is to provide the users with information that should correspond to their needs. In the field of Information Quality (IQ) the consumer (user) viewpoint is important and the concept “fitness of use” is widely recognized. IQ can be considered a measure of the value the information provides the user of the information. Quality is often perceived as subjective and the quality of information can then vary among users and among uses of the information.

A general criticism of IQ as a research field is that most approaches lack any means (or even suggestions) to assess quality in an objective way. However, to manage and evaluate the user satisfaction of information in a CMS, objective control mechanisms need to be in place. The subjective and qualitative opinions of users need to be complemented with quantitative and objective analysis procedures.

In order to deal with the opportunities and problems discussed above, we use approaches that assess the quality of technical documentation using technology from software quality assessment. Software quality assessment is built upon analysis and quantitative measurement, testing, and visualization for communicating results to the stakeholders.

Hence, in order to investigate how well the software quality point of view works for technical documentation a limited case study was conducted. A number of software quality analyses and visualizations were adapted and integrated into a commercial CMS. The analyses were used to evaluate (parts of) real world technical documentation from a Swedish mobile phone vendor, a Japanese camera vendor, and a Swedish warship yard. The evaluation found quality issues, which suggests that applying software quality assessment to IQ is a feasible strategy.

The rest of this paper is organized as follows. Section 2 discusses the notion of quality from both the software and the information perspective. Section 3 then defines our stance on how information quality can be assessed similarly to the assessment of software quality can. Section 4 presents the case study and Section 5 concludes the paper.

2 Software and Information Quality

In order to discuss quality, and to be able to assess and improve it, it has to be defined. Crosby (1979) defines quality as “conformance to requirements”. This definition suggests that a set of requirements exists that is defined in such a way that they cannot be misunderstood. Something that is of quality conforms to these requirements and something that does not conform is considered a defect. The requirements are not to be considered universal, but set by an entity or for a single product. For example, a car made by Volvo that conforms to the requirements set by Volvo is considered to be of high quality. Another car made by BMW that conforms to the requirements set by BMW is also considered to be of high quality. Even if the two cars have different goals, sizes, properties, both are of high quality.

Another notion of quality is given by Juran (1998), who defines quality as “fitness for use”. This definition considers the customers and the requirements and expectations that they have on the product for their particular use. Juran further stats that since different customers may use the product in different ways, the product must possess multiple elements of fitness of use. These elements are quality characteristics that can be divided into parameters.

The two definitions of quality can seem unrelated, but in reality they complement each other very well. The demands and expectations of the customer’s guides the requirements set by the producers. So, a conformance to the requirements generally means that the product will be fit for use as well.

2.1 Software Quality

One of the challenges of Software Quality is that "everyone feels they understand it" Crosby (1979). But, in order to discuss the quality of software, there is a need to define what it means. Software quality at least initially focused on the "conformance to requirements" aspect of quality and defined a range of characteristics that high quality software should possess. McCall et al. (1977) presents model for software. The model defines 11 quality factors that relate to the three stages of a simplified software life cycle: revision, operation and transition. McCall et al. also define about 22 metrics that are used to measure the quality of the 11 factors. Several metrics are weighted and used to determine the quality of each factor. Many of the metrics are based on checklists and a 0 to 10 scale, which means that they are subjectively measured.

Although the assessment was still subjective, the model by McCall et al. introduces several important ideas. First, there is not one product quality, but several factors that affect the product quality. Second, these factors matter during different periods of the life cycle. Third, the quality factors should be measurable and metrics should be defined. Several modern quality models and metrics suites for an automated analysis exist, such as those by Chidamber and Kemerer (1994) and Abreu (1995), but they are generally extensions and modifications to the ideas brought forward by McCall et al. There are several studies that validate the claim that metrics can be used as an indicator of the quality of software, for example Basili et al. (1996) and Harrison et al. (1998).

2.2 Information Quality

In the Information Quality (IQ) field, there has also been work on the expansion on the definition of quality. Wang and Strong (1996) represent the forerunners and define IQ as the information that is fitness for use by information consumers (building on Juran). However, as Klein et al. (1997) showed that information consumers had trouble pointing out and detecting errors in information and altering the way they use it, placing all the responsibility there is not optimal. In response to this, Kahn et al. (2002) suggested a view where quality also depended on conforming to specifications (adopted from Crosby). So, from this view, IQ can be defined as the information that meets the specifications or requirements. These two views are called user perspective and data perspective respectively. And, in combination it would state that high quality in regard to information is when it is free of defects and possesses desired features.

In the process of assessing IQ, Ge and Helfert (2007) classified typical IQ problems in a two by two model based on user and data perspective, and context-independent and context-depended (see Table 1). User perspective/context-independent quadrant indicates the IQ problems that may happen in processing the information. User perspective/Context dependent quadrant indicates the IQ problems that are not fitness for intended use by information consumers. Data perspective /Context independent quadrant indicates the IQ problems in the CMS. These IQ problems can be applied to any data set. Data perspective/context dependent quadrant indicates the IQ problems that violate the business specifications. These IQ problems can be detected by contextual rules.

	User perspective	**Data perspective**
Context independent	The information: is inaccessible, insecure, hardly retrievable, and difficult to aggregate; errors in the information transformation exist.	Inconsistent data format, spelling errors, missing, outdated or duplicate data, incorrect value, incomplete data format, syntax or unique value violation, violation of integrity constraints, text formatting.
Context dependent	The information: is not based on fact, of doubtful credibility, presents an impartial view, irrelevant to the work, compactly represented, hard to manipulate, hard to understand.	Violation of domain constraint, of organization's business rules, of company and government regulations, of constraints provided by the database, consists of inconsistent meanings.

Table 1: Classification of Information Quality Problems – adapted from Ge and Helfert (2007)

2.3 Measuring Quality of Technical Documentation

Nowadays, the technical documentation regardless of printed or electronic variants are created in and generated from CMSs, and the information and software aspects are blending. For example, car manufacturer Hyundai recently announced that their new Equus model would include an Apple iPad tablet computer that will replace the prior printed owner's manual. The documentation on the iPad will be an interactive "application", and it can for example aid the owner in scheduling maintenance appointments for the car. However, even in more traditional electronically available documentation, such as Web pages or PDF documents, there is a strong software presence. Software is used to make the documents available and provide means to access them. The quality of the user experience will to a large extent rely on these pieces of software, and a poor and limited viewer option will reflect badly on the perceived quality of the technical documentation and in turn the product itself.

Hence, to reason about the quality of technical documentation, there is a need to consider both information and software quality aspects. It is important to be able to assess — to measure — the quality of technical documents. The measurement should serve as both an absolute and relative value. It should be possible to say that the quality is good (based on the measurement) or that the quality has improved (again based on measurements). Not all documents, products, or projects have the same properties or should be judged by the same quality model, so the ability to adjust which properties are included, i.e. the properties of the quality model, is important. In order to be able to improve the quality, it is also very important to get feedback on where the (potential) issues are. Ergo, one cannot manage IQ without first being able to measure it meaningfully Stvilia et al. (2005)

To the best of our knowledge, our work was the first suggesting IQ assessment with metrics and quality models adopted form software quality assessment and the first showing the feasibility of this approach in a study with real world documentations (Wingkvist et al. 2010).

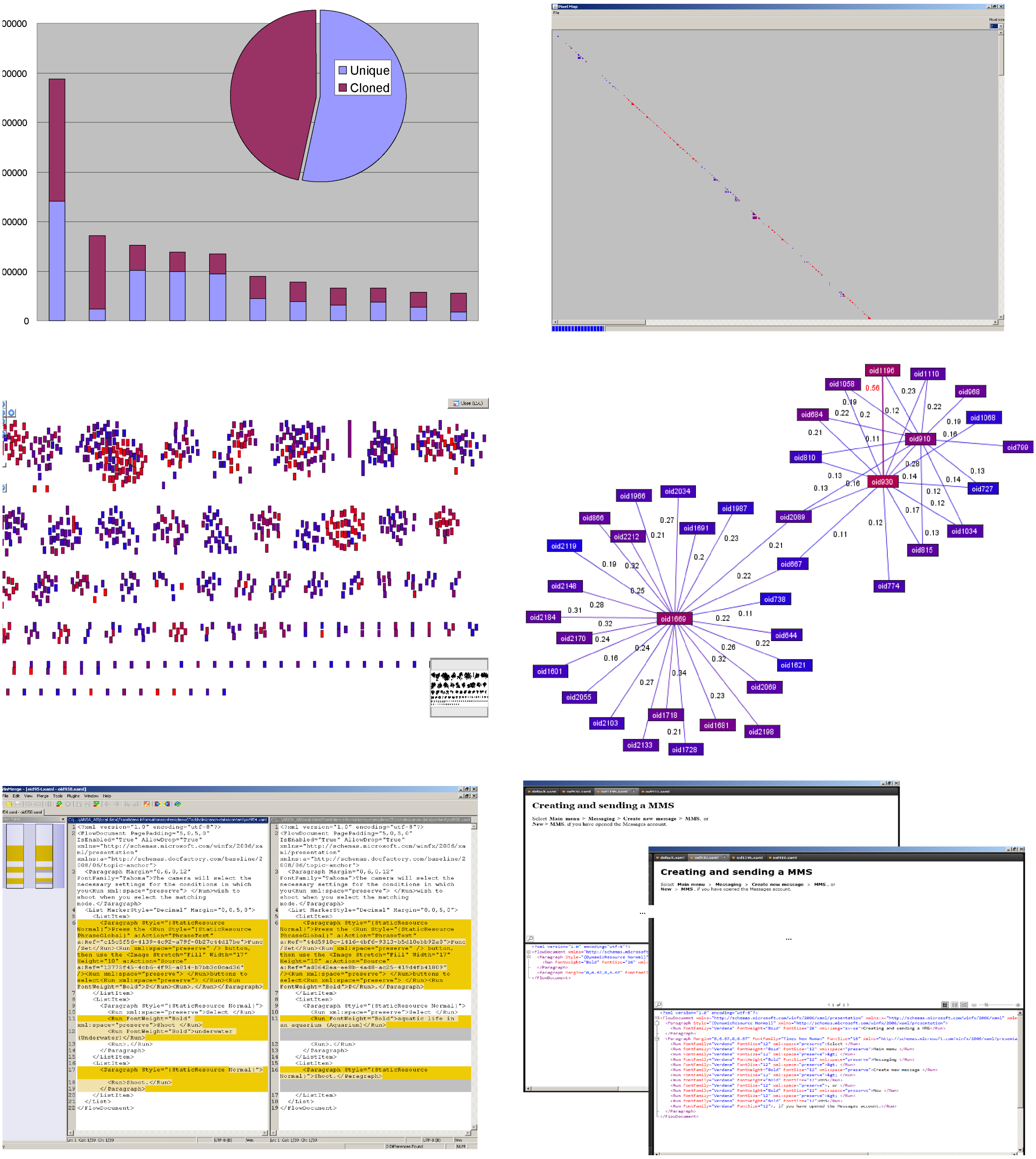
3. Analysis and Visualization of Information Quality

A picture says more than thousand words or metrics values in our case. This is the theme of the relatively young field of "Visual Analytics". Visual Analytics integrates visualization, analysis, and human expert knowledge (Thomas and Cook, 2005). The visualization of an analysis process allows controlling it directly instead of being left with just the results (Keim et al., 2008). In this way, visualization serves as a medium for efficient cooperation between humans and machines, stakeholders of a technical documentation and automated quality analyses. In what follows we suggest a number of analyses and visualizations and later, in Section 4, we demonstrate them in concrete technical documentations and suggest interpretations of the results.

3.1. Analyses

Clone detection is an analysis detecting text copies in the technical documentation's source as stored in a CMS repository. Awareness of text copies is important for quality as it allows fixing errors consistently, guarantees the unambiguousness of information, and generally leads to low maintenance costs, most notably of all translation costs.

The text topics detected ought to be presented to different stakeholders in different views. Owners of the product documented are interested in a higher level of abstraction and prefer a statistical overview as depicted in Figure 2(a). It presents the fractions of cloned vs. unique text in a pie chart for the whole document and in a bar chart for the 10 largest documents. A project manager responsible for the technical documentation might prefer a lower abstraction level like the one given by a so-called "pixmap" view – Figure 2(b) – or a cluster view – Figures 2(c, d). The pixmap in Figure 2(b) presents the analysis results in a pixel matrix where each pixel corresponds to a document pair sharing some text fraction; the color schema additionally encodes the degree of similarity (ranging from blue = only little text in common to red = a large fraction of the text is common). Finally, the information engineer or technical writer, who eventually needs to remove the copies, is interested in a document comparison view highlighting communalities of two texts – Figure 2(e) – and eventually in the view of an editor that a CMS usually provides – Figure 2(f).



C:\...\ARISA_AB\local data\Framtidens informationssystem\demoSTockholm\canon-data\content\oid954.xaml

```
<?xml version="1.0" encoding="utf-8"?>
<FlowDocument PagePadding="5,0,5,0"
IsEnabled="True" AllowDrop="True"
xmlns="http://schemas.microsoft.com/winfx/2006/xa
ml/presentation"
xmlns:a="http://schemas.docfactory.com/baseline/2
008/06/topic-anchor">
  <Paragraph Margin="0,6,0,12"
FontFamily="Tahoma">The camera will select the
necessary settings for the conditions in which
you<Run xml:space="preserve"> </Run>wish to
shoot when you select the matching
mode.</Paragraph>
  <List MarkerStyle="Decimal" Margin="0,0,5,0">
    <ListItem>
      <Paragraph Style="{StaticResource
Normal}">Press the <Run Style="{StaticResource
PhraseGlobal}" a:Action="PhraseText"
a:Ref="c15c5f56-4139-4c92-a79f-0b27c44d17be">Func
/Set</Run><Run xml:space="preserve" /> button,
then use the <Image Stretch="Fill" Width="17"
Height="10" a:Action="Source"
a:Ref="13775f45-4cb6-4f95-a814-b7bb3c0cad36"
/><Run xml:space="preserve"> </Run>buttons to
select<Run xml:space="preserve"> </Run><Run
FontWeight="Bold">P</Run><Run>.</Run></Paragraph>
    </ListItem>
    <ListItem>
      <Paragraph Style="{StaticResource Normal}">
        <Run xml:space="preserve">Select </Run>
        <Run FontWeight="Bold"
xml:space="preserve">Shoot </Run>
        <Run FontWeight="Bold">underwater
(Underwater)</Run>
        <Run>.</Run>
      </Paragraph>
    </ListItem>
    <ListItem>
      <Paragraph Style="{StaticResource Normal}">

        <Run>Shoot.</Run>
      </Paragraph>
    </ListItem>
  </List>
</FlowDocument>
```

Creating and sending a MMS

Select **Main menu** > **Messaging** > **Create new message** > **MMS**, or **New** > **MMS**, if you have opened the Messages account.

Figure 1: Visualizations of clone detection results, from top left to bottom right: (a) statistics view, (b) pixmap view, (c) cluster overview, (d) detailed cluster view, (e) document comparison view, (f) document editor view

Usage analysis traces the access behavior of users. It checks if the actual accesses are as expected including the access path, access times, and access frequencies. During test usage, e.g., proofreading the documentation, it helps to improve the coverage of testing. During production usage, it allows identifying useless, hard to find, and confusing parts. Altogether, it improves the relevance, accessibility and completeness of the technical documentation.

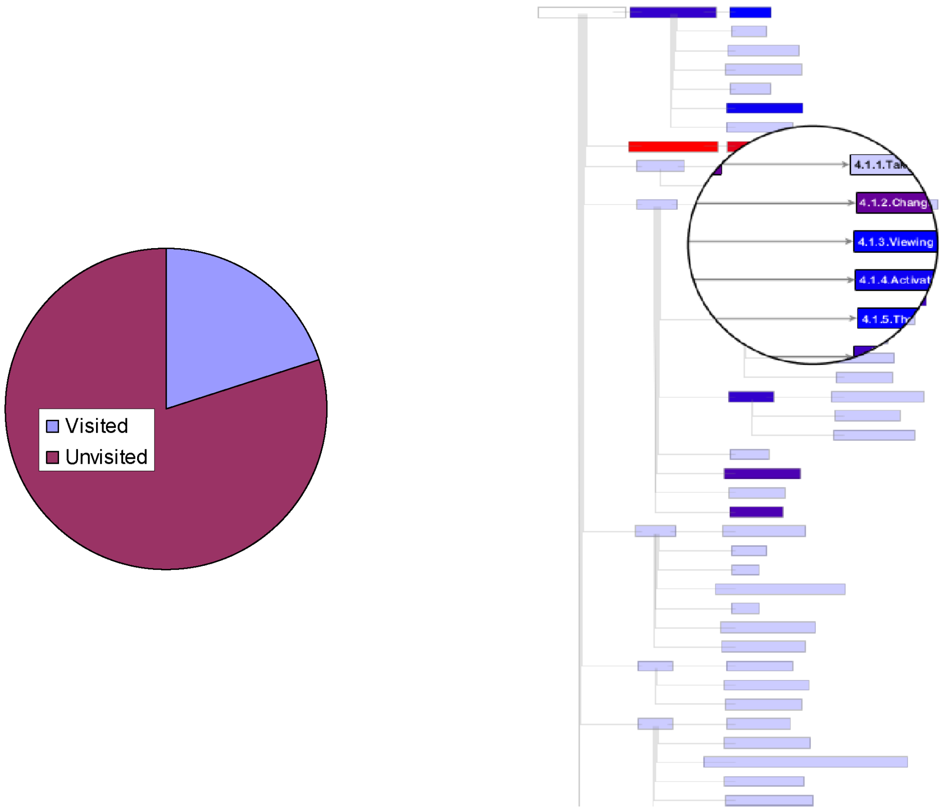


Figure 2: Visualizations of usage analysis results, left to right: (a) statistics view, (b) graph view

The results of usage analysis could again be presented on different levels of abstraction, depending of stakeholder roles. We present two examples. A statistics view – Figure 2(a) – shows the percentage of documents covered in an (test) use of the documentation. A graph view – Figure 2(b) – depicts the documentation structure and color-codes actual usage of the individual documents contained (light blue = documents not visited at all, then dark blue = document visited briefly to red = document visited extensively).

Structure analysis detects the hierarchy (chapters, sections, subsections etc.) of documents in technical documentations and the documents' references. References include cross-references, external hyperlinks, citations, cross-language links, references to media resources etc. Structure analysis helps to assess balance and concinnity of the whole documentation, logic cohesion of its sections and subsections, and uncovers missing and unnecessary references. Altogether, it helps improving qualities like understandability, accessibility, and suitability of presentation of the technical documentation.

Views can be presented on different level of abstraction. Orthogonally to the abstraction level, different aspects of structure can be presented, which we exemplify here. Figure 3(a) depicts the document hierarchy (each document box color stands for a section, height of document boxes corresponds to the relative document size). Figure 3(b) depicts the cross-reference structure of another documentation (color still encode sections). The proximity of document boxes corresponds to the number of cross-references.

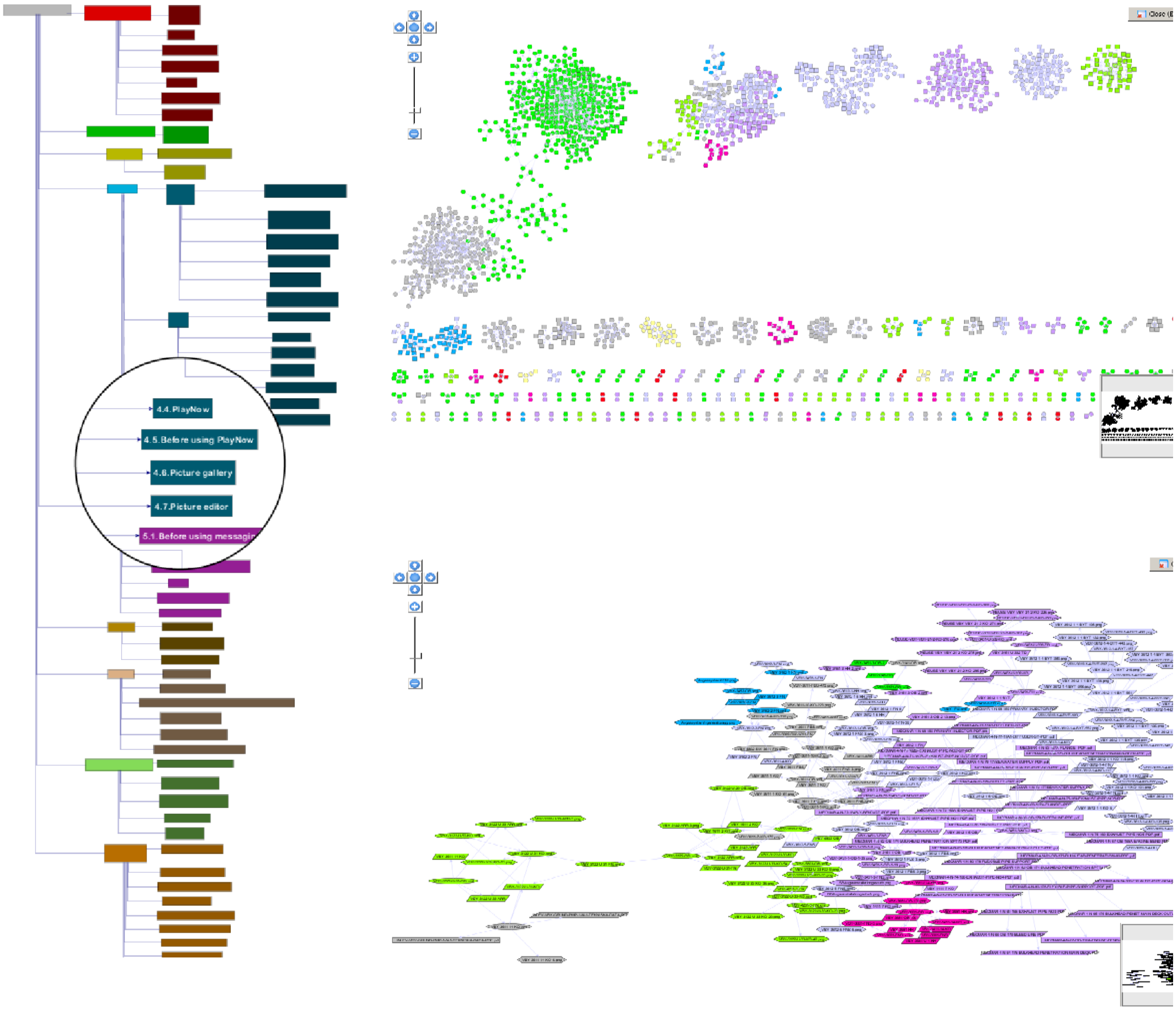


Figure 3: Visualizations of structure analysis results, left to right: (a) document hierarchy view, (b) cross-reference overview (top) and detailed view zoomed in the overview (bottom)

Meta-information analysis detects the structure of meta-information and relates meta-information and information. Meta-information includes data types, database rules, schemata, conventions, search tags etc., which are usually further, grouped in type hierarchies or classes, which constitutes the structure of meta-information. They are connected to the documents of technical documentations that adhere to or could be described by meta-information. This analysis assures the existence of meta-information and its appropriate structure. It helps to assure that individual documents are (completely) attached to their corresponding meta-information.

Visualizations of meta-information analysis results do not need to vary in abstraction since they support engineers rather than managers. Figure 4(a) shows the structure of meta-information; each meta-information document (box) belongs to a category (color encoded). Figure 4(b) relates meta-information documents (boxes still with different colors for different categories) to information sections (white boxes).

4. Case Study

The example visualizations in the previous section originate from a case study described in this section. The case study was carried out in an attempt to assess quality of technical documentation of three products: (a) a mobile phone from a Swedish mobile phone vendor, (b) a camera of a Japanese vendor, and (c) a war- ship of a Swedish shipyard. All sources in this case study are based on real technical documentations. However, in the warship case we were limited to an unclassified subset of the documentation.

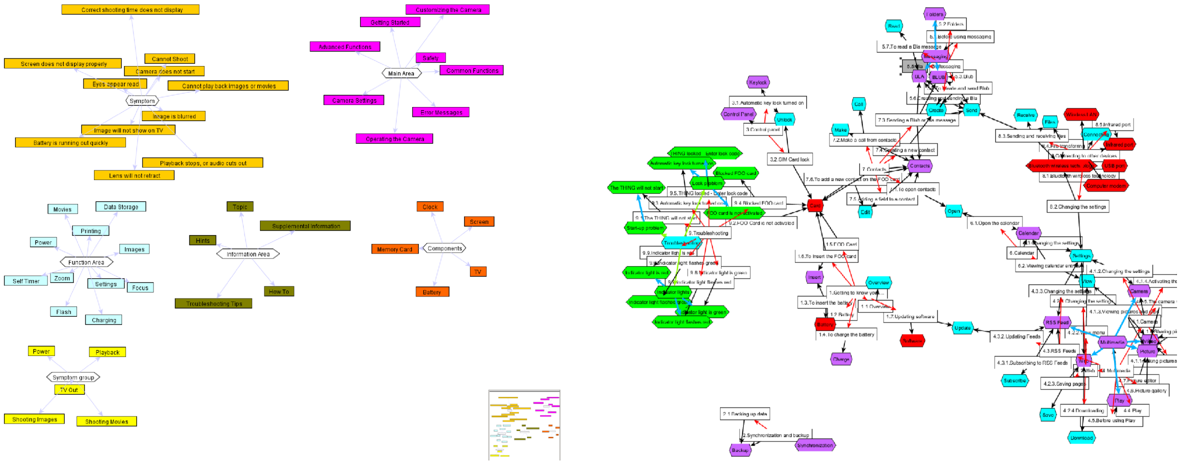


Figure 4: Visualizations of meta-information analysis results, left to right: (a) meta-information structure view, (b) view of relation meta-information to information

4.1 Setup

All technical documentation sources used in the case study were available in XML format. Documentations (a) and (b) were produced with *DocFactory* (DocFactory, 2010), a CMS for technical documentation supporting, for example, version control, language management, information presentation using standard Web browsers etc. In order to assess the quality of the documents, the software quality analysis tool *VizzAnalyzer* (Löwe and Panas, 2005, VizzAnalyzer, 2010) was used. The VizzAnalyzer supports a wide range of analyses, for example clone detection and usage analysis used as part of the case study. VizzAnalyzer outputs visualization data that is displayed using several information visualization tools such as Microsoft *Excel* and the *yEd* graph viewer (yEd, 2010).

4.2 Results

Clone detection determines the similarity between two documents by first comparing the text on a paragraph level and then the XML structures of the two documents. The result is a percentage that indicates the relative size of the parts of the two documents that are unique.

Clone detection was applied to 72 documents of a first version of (a) and 890 XML documents of documentation (c). The analysis of (a) showed that only one document was unique. The remaining 71 documents were at least in part cloned. On average, a document was 76% unique. In study (c), 6 documents were unique and 20 were completely cloned. The remaining 864 documents were at least in part cloned. On average, a document was 54% unique. Figure 1(a, b, c) shows results of study (c) and Figure 1(d, e, f) shows results of study (a).

Usage analysis allows assessing the behavior of users when browsing documentations. We analyzed and visualized demo presentations of the documentations (a) and (b). Figure 2 depicts the analysis of (a). The usage analysis was performed as a feasibility study, and while we never intended to draw conclusions from the results, they show that it is in principle possible to analyze and visualize usage traces.

Structure analysis was applied to documentations (a) and (c); the results are displayed in Figures 3 (a) and (b) respectively. Analysis of documentations (a) uncovered a certain imbalance:(a) contained two very short sections (Sections 2 and 3) with only one and two, resp., subsections. Analysis of documentation (c) discovered documents with many cross-references between different sections, cf. Figure 3(b) bottom, which sacrifices cohesion of sections and may affect how readable the documentation is.

Meta-information analysis was applied to documentations (a) and (c). Figure 4(a) shows the grouping of so-called "applicability" tags of documentation (c) in different categories. The analysis shows that each tag belongs to just one category. Figure 4(b) shows the tagging of sections with applicability tags. The tags of some categories were either concentrated to few subsections and the tags of other categories spread all over the documentation. Both observations were confirmed as intended design.

4.4 Evaluation

The case study shows that the selected software quality analyses and visualizations can be applied to technical documentation and that they are useful to find quality issues within documentation. The different analyses can pinpoint potential problems that can then be investigated in detail. The different visualizations can provide a basis for communication suitable for different groups of stakeholders.

However, there are open issues in interpreting the analysis and visualization results. For instance, what does it mean that 49% of the documentation is non-unique; is this poor quality? In software quality assessment, there exist certain recommendations for interpretation based on a huge amount of systems analyzed, e.g., by Barkmann et al. (2009). Such investigations are lacking when it comes to technical documentations.

The case study was conducted at a company that produces both content and CMS tools. The company is responsible for maintaining the technical documentations investigated. They showed great interest in the results and expressed that they saw great potential of the approach.

5. Conclusions and Future Directions

This paper adopts analyses and visualization from software quality management to technical documentation. The approach was tested using the software analysis tools and actual data of a company producing technical documentation. The initial results shows that even simple software analyses and visualizations can be used to find quality issues for technical documentation.

The case study presented uses a limited number of analyses and visualization that were applied to three technical documentations. The next step is to increase both and improve the confidence of the interpretation of results.

There is also a need to integrate analyses that are not as easily or even impossible to perform automatically. Some analyses will have to be done by user sampling, for example. There is also a need to consider a wider concept of quality that includes the consumers of the documentation. The approach taken in the current case study follows the "conformance to requirements" view on quality, and future experiments should include the "fitness of use" model and consider how the technical documentation fits the needs and requirements of the end users.

Acknowledgment

Anna Wingkvist is supported by the Swedish Research School of Management and IT (MIT), Uppsala. Morgan Ericsson is supported by the Uppsala Programming for Multicore Architectures Research Center (UPMARC). We would like to thank the Knowledge Foundation for financing part of the research with the project, "Validation of metric-based quality control", 2005/0218. We also would like to extend our gratitude to Applied Research in System Analysis AB (ARiSA AB, http://www.arisa.se) for providing us with the VizzAnalyzer tool and to Sigma Kudos AB, (http://www.sigmakudos.com) for providing us with DocFactory and raw data.

References

Abreu, F. B. (1995), “The MOOD metrics set”, in Proceedings of the ECOOP Workshop on Metrics.

Barkmann, H., Lincke, R. and Löwe, W. (2009), “Quantitative Evaluation of Software Quality Metrics in Open-Source Projects”, in Proceedings of the IEEE International Workshop on Quantitative Evaluation of large-scale Systems and Technologies (QuEST ’09).

Basili, V. R., Briand, L. C. and Melo, W. L. (1996), “A Validation of Object-Oriented Design Metrics as Quality Indicators”, IEEE Trans. Software Engineering. 22(10), 751–761.

Basili, V. R., Caldiera, G. and Rombach, H. D. (1994), “The goal question metric approach”, in Encyclopedia of Software Engineering, Wiley.

Chidamber, S. R. and Kemerer, C. F. (1994), “A Metrics Suite for Object-Oriented Design”, IEEE Transactions Software Engineering 20(6), 476–493.

Crosby, P. B. (1979), “Quality is free: the art of making quality certain", McGraw-Hill, New York.

DocFactory (2010). http://www.sigmakudos.com/data/services/docfactory, accessed April 2010.

Ge, M. and Helfert, M. (2007), “A review of information quality research — develop a research agenda”, in Proceedings of the 12th International Conference on Information Quality.

Harrison, R., Counsell, S. J. and Nithi, R. V. (1998), “An Investigation into the Applicability and Validity of Object-Oriented Design Metrics”, Empirical Software Engineering 3(3), 255–273.

Juran, J. (1998), “Juran’s Quality Control Handbook”, 5th edition, McGraw-Hill.

Kahn, B. K., Strong, D. M. and Wang, R. Y. (2002), Information quality benchmarks: product and service performance, Commun. ACM 45(4), 184–192.

Keim, D., Andrienko, G., Fekete, J., Görg, C., Kohlhammer, J. and Melançon, G. (2008), “Visual Analytics: Definition, Process, and Challenges”, in A. Kerren, J. Stasko, J. Fekete, and C. North, editors, Information Visualization – Human-Centered Issues & Perspectives, volume 4950 of LNCS State-of-the-Art Survey, Springer.

Klein, B. D., Goodhue, D. L. and Davis, G. B. (1997), “Can humans detect errors in data? Impact of base rates, incentives, and goals”, MISQ. 21(2), 169–194.

Löwe, W. and Panas, Th. (2005), “Rapid Construction of Software Comprehension Tools”, International Journal of Software Engineering and Knowledge Engineering. Special Issue on Maturing the Practice of Software Artefacts Comprehension, Ed. Nenad Stankovic, 15(6), 905-1023, World Scientific Publishing.

McCall, J. A., Richards, P. G. and Walters, G. F. (1977), “Factors in Software Quality”, Technical Report Volume I, NTIS, NTIS Springfield, VA. NTIS AD/A-049 014.

Stvilia, B., Twidale, M. B., Smith, L. C. and Gasser, L. (2005), “Assessing information quality of a community-based encyclopedia”, in Proceedings of the International Conference on Information Quality’, pp. 442–454.

Thomas, J.J. and Cook, K. A. (2005), “Illuminating the Path”, IEEE Computer Society.

VizzAnalyzer (2010). http://www.arisa.se/vizz_analyzer.php, accessed June 2010.

Wang, R. Y. and Strong, D. M. (1996), “Beyond accuracy: what data quality means to data consumers”, J. Manage. Inf. Syst. 12(4), 5–33.

Wingkvist, A., Ericsson, M., Lincke, R. and Löwe, W. (2010), “A Metrics-Based Approach to Technical Documentation Quality”, in Proceedings of the 7th International Conference on the Quality of Information and Communications Technology (QUATIC’2010).

yEd (2010). http://www.yworks.com/en/products_yed_about.html, accessed June 2010.