

A Visualization-based Approach to Present and Assess Technical Documentation Quality

Anna Wingkvist, Morgan Ericsson and Welf Löwe

Linnaeus University, Växjö, Sweden

anna.wingkvist@lnu.se

morgan.ericsson@lnu.se

welf.lowe@lnu.se

Abstract

Technical documentation has moved from printed booklets to electronic 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.

In order to ensure the quality of technical documentation, it is important to be able to measure it in a constructive way. In this paper, we approach quality from a software quality perspective and rely on automated measurements and analyses. It is generally not possible to assess quality attributes such as “ease of understanding” using automated measurements. To assess such quality attributes, we suggest the use of visualizations as a communication medium between the machine results and technical writers, and define a visualization-based quality assessment approach for technical documentation.

In order to test our approach, we use it to assess the quality of 3 real-world documentations from a Swedish mobile phone manufacture, a Japanese camera manufacturer, and a Swedish warship producer. The study shows that our approach can be used to identify potential quality defects. For example, we tested an unclassified subset of the warship’s technical documentation and found that 49% of it was redundant text clones. We performed the study in collaboration with a Swedish company that is in charge of creating and maintaining the 3 documentations, and they acknowledge that our approach has great potential and that our results proved helpful to them.

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

1. Introduction

Technical documentation (user manuals) 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.

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. The life span of a warship is in stark contrast to an end-consumer product, such as a mobile phone. However, the pervasiveness of software in products makes it feasible to offer modifications and upgrades to products with shorter life spans as well. But no matter the life span of a product, there is a need to offer documentation that is correct and up to date.

Traditionally, technical documentation was the printed booklets that accompanied products. Currently, most of the technical documentation is also available electronically, from several different sources. The same documentation might, for example, be available as HTML and PDF on the company web site, as online help within the product, and as printed booklets delivered with the product. The electronic versions make the documentation easier to access for the customers and easier to keep correct and up to date for the manufacturers. However, the documentation will now exist in more versions (HTML, PDF, ...) and have a more complex maintenance cycle.

In this paper we investigate how to assess and assure the quality of technical documentation. The field of information quality defines quality as “fitness for use by information consumers”. Quality is perceived as subjective, and can vary among users and uses of the information. Stvilia et al. (2005) point out that one cannot manage quality without first being able to measure it meaningfully. This is a problem, since the subjective nature of information quality makes it difficult to measure constructively. Given the highly structured nature of technical documentation and use of markup languages such as

XML, we view the quality of technical documentation from a software perspective. Software quality is defined from metrics and attributes that can be measured.

We acknowledge that it is hard or even impossible to define important quality attributes such as “ease of understanding” based on metrics and attributes alone. We therefore rely on visualizations to serve as a medium for an efficient cooperation between human (i.e., technical writers) and machines. The rest of the paper is organized as follows. Section 2 introduces the notion of quality from an information and a software perspective. Section 3 introduces visual analytics and shows how it incorporates machine analysis and human expertise and how visualizations serve as a communication medium. Section 4 presents the results of a study of our visualization-based approach to quality presentation and assessment of technical documentation, and Section 5 concludes the paper and presents future directions.

2. The Notion of Quality

Quality can be a confusing and vague concept. It is often used as an intangible trait; something that cannot be exactly measured or weighed, but rather subjectively felt or judged. It is also a multidimensional concept that includes an object of interest, a viewpoint, and a number of attributes that are ascribed to the object.

In order to discuss, assess and improve quality, there is a need to define it better. Juran (1998) acknowledges the multidimensional nature of quality and defines it as “fitness for use”. His definition considers the customers, their requirements and expectations, as well as their particular use. Since different customers may use the product in different ways, it must possess multiple elements of fitness for use.

Crosby (1979) defines quality as “conformance to requirements”. The definition relies on the existence of a set of precisely defined requirements, and in order for something to be considered of quality, it has to conform to these requirements. The requirements are not universal; they are set by an entity or for a single product.

The two definitions both acknowledge the existence of elements, such as requirements, that are used to evaluate quality. The rest of this section studies efforts to find and classify these elements in order to define what quality of software and information, respectively, means.

2.1 Software Quality

An early attempt to define software quality by McCall et al. (1977) present a quality model defining eleven quality factors that relate to the three stages of a simplified software life-cycle: revision, operation and transition. McCall et al. further define about 22 metrics that are used to measure the quality of the eleven 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. This quality model is standardized in ISO/IEC 9126-1:2001 (ISO, 2001). Quality models also differentiate perspectives, stages or phases. There are several aspects of quality and not all of these may be appropriate for all perspectives and phases. McConnell (2004) differentiate between internal and external quality, i.e., quality that affects the product while produced versus quality when the product is in use. These aspects are standardized in ISO/IEC 9126, as well.

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 product life cycle. Third, the quality factors should be measurable and metrics should be defined. The quality factors are examples of the elements discussed previously. While the definition of software quality is focused on conformance to requirements (factors) that are universally defined, it also acknowledges the notion of use, i.e., certain factors are relevant at certain stages. This is similar to Juran’s definition of quality, i.e., that quality is related to the use.

2.2 Information Quality

As stated previously, a common definition of information quality is closely related to Juran’s definition of quality; “fitness for use by information consumers” (Wang & Strong, 1996). Klein et al. (1997) argues that this definition of information quality is not always appropriate, as information consumers can find it difficult to both detect and report defects in the information as well as alter the way they use

it as a result of the errors. So, placing the full responsibility on the information consumers is not appropriate. Kahn et al. (2002) suggests that information quality should be extended with a data perspective that views quality according to “conformance to requirements” (adopted from Crosby (1979)). This perspective considers information quality from an “all users” point of view, and does not depend on the actual user. The user and data perspectives complement each other.

Ge and Helfert (2007) rely on the user and data perspective, and further extends the definition of information quality by introducing another dimension with context dependence/independence perspectives. This dimension determines whether the quality depends on the actual use or not. They further classify a number of information quality defects according to the two dimensions. Table 1 depicts the two dimensions and some of the information quality defects, as classified by Ge and Helfert.

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

	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.

3. Visual Analytics and Quality Assessment

The discussion on information and software quality in Section 2 shows that quality is a confusing concept that is difficult to define, and consequently, difficult to analyze. Software quality follows a bottom-up approach, and defines quality based on metrics, methods to measure data, which approximate quality attributes. While the quality model by McCall et al. (1977) uses metrics that can be manually assessed, the current research focus on metrics that can be automatically assessed. The various quality models defined for information shows that there could be a huge gap between an ideal definition of quality, an approximation by attributes and what can be measured by metrics. For example, the negative quality attribute “the information is hard to understand” (cf. Table 1) can be difficult, or even impossible, to approximate by a set of metrics. A quality attribute such as “No spelling errors” can be supported by metrics, but there is often the need for human intervention, since there may be false positives, such as technical terms. There is also a risk that as we include more attributes and metrics, the quality model grows increasingly complex and difficult to understand and use in practice.

The way we deal with a quality attribute such as “No spelling errors” presents an interesting problem. For real-world texts, it is not reasonable to rely on either automatic or manual analysis solely. Manual analysis will be time consuming and error prone, and automatic analysis can produce false positives (both with respect to correct and incorrect spelling). The solution implemented by any word processor is to combine the two, and use manual inspection of the results of the automatic analysis.

This relates to Visual analytics, which is the science of analytical reasoning facilitated by interactive visual interfaces (Thomas & Cook 2005). It can be a helpful way to approach problems that can be difficult to solve using either machine analysis or human analysis, for example due to size or complexity. The combination of visualization, machine analysis and human expertise can help to reduce both size and complexity, for example by applying interactive visualizations and by coupling human and machine analysis. Visual analytics is more than just visualization and can rather be seen as an integrated approach combining visualization, human factors and data analysis (Keim et al. 2006).

Thomas and Cook (2005, 2006) describe visual analytics as a multidisciplinary field with the following focus areas:

- analytical reasoning techniques to support assessment, planning and decision making;
- visual representations and interaction techniques that allows users to see, explore and understand large amounts of information;
- data representations and transformations to support visualization; and
- techniques to communicate the analytical results to a variety of audiences.

Visualization serves as a medium for an efficient cooperation between humans and machines. In our case, the automated quality analyses (that are supported by metrics) provide the data. There is a need to define transformation functions to support visualization as well as to make both the visualizations and analyses interactive. This marriage of computation, visual representation, and interactive thinking supports analytic reasoning.

Further, Cook et al. (2007) mention that the goal of the sense-making process is two fold; detect unanticipated results and confirm the expected. The value lays in being able to cross reference several visualizations to gain greater understanding and see surprising anomalies, changes, patterns, and relationships, which are then presented and assessed to develop new insight about the data. The visual representations make it easy to perceive salient aspects of the data quickly. Augmenting the cognitive reasoning process with perceptual reasoning through visual representations permits the analytical reasoning process to become faster and more focused according to Thomas and Cook (2005).

4. Study

In order to investigate how visual analytics can help assess the quality of technical documentation, we performed a study. We extended the software quality analysis tool VizzAnalyzer (Löwe & Panas 2005) with the ability to read XML documents and implemented 4 quality assessment analyses for technical documentation. We also implemented 12 different ways to visualize the result of the analyses. We tested the analyses and visualizations on 3 different real-world technical documentations that we got access to through a Swedish company responsible for creating and maintaining these. These were from a Swedish mobile phone manufacturer, a Japanese camera manufacturer, and a Swedish shipyard. This section presents the analyses and visualizations, the technical documentations, and the outcome of the analyses.

4.1 Analyses and Visualizations

Clone detection is an analysis done to detect text copies in the technical documentation. 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 clones detected should be presented to different stakeholders using different views. The owners of the documentation are interested in the information on a high level of abstraction, and might prefer a statistical overview. This can be visualized using scientific visualizations suitable for scalar data, such as pie charts, bar charts, etc. A project manager might be interested in the information on a lower abstraction level, for example showing which parts of the documentation are similar, and how similar they are. This can be visualized using information visualizations, where documents are linked and clustered, based on similarity. A technical writer or information engineer is interested in where the exact clones are, so they can decide if it is a quality defect or not, and if so, fix it. This can visualized using views that show the actual text.

Usage profile traces the access behavior of users. It records the access path, access times, and access frequencies of a user using the technical documentation. During test usage, e.g., manual proofreading of the documentation, it helps to improve the coverage of testing. During production usage, it can help to identify which parts are seldom used, hard to find, or confusing. Usage profiles can be used to improve the relevance, accessibility, and completeness of the technical documentation.

Usage profiles can be visualized in several different ways. For stakeholders interested in information on a high level of abstraction, it is visualized using statistical information, such as the percentage of the documentation that was covered. On a lower level of abstraction, the usage profile can be

visualized by a graph showing the document structure that is annotated to show access paths and frequency, for example using color-coding.

Structure analysis is used to analyze the technical documentation is structured, with respect to chapters, sections, paragraphs, etc. It also analyses the references, such as hyper-links within the document, citations, footnotes, links to external 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. It can be used to improve understandability, accessibility, and suitability of presentation of the technical documentation.

Structure analysis can be visualized on different levels of abstraction, as well as using different aspects. For example, hyperlinks between different documents or the relative size of each section can be visualized.

Meta-information analysis detects the structure of meta-information and relates meta-information and information. Meta-information includes data types, database rules, schemata, conventions, and search tags etc., which are usually 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. The analysis assures the existence of meta-information and its appropriate structure. Meta-information analysis can be used to assure that individual documents are (completely) attached to their corresponding meta-information.

Meta-information analysis is general only of interest to technical writers and information engineers, so visualizations should be at a low level of abstraction. The analysis can be visualized by showing the structure of the visualization using graphs, or by relating the document structure to meta-information.

4.2 Technical Documentations

We used the technical documentation of three products: (I) a mobile phone from a Swedish mobile phone manufacturer, (II) a camera of a Japanese manufacturer, and (III) a warship of a Swedish shipyard. All sources in this study are based on real technical documentations. However, in the warship case we were limited to an unclassified subset of the documentation.

Documentation I is by far the largest and comprises 12,286 XML documents, each containing a topic of the documentation. Most of them are no larger than 1 KiB (corresponding to 1/2 a page of description text), none of them is larger than 5 KiB (corresponding to approx. 2 pages of description). The documentation is explicitly structured by means of an XML file of almost 7 MiB size, containing references to the actual topics.

Documentation II is comparably small. It comprises of 143 XML documents and 11 JPEG images. The documentation is structured by means of an XML file of 133 KiB size with references to the topics. Documentation III is of moderate size comprising altogether 3242 documents: 611 PDF and 1715 PNG documents (technical drawings), 913 XML (technical descriptions) and 3 DTD documents (meta-data). As opposed to the former two, the documentation is implicitly structured in about 800 directories and subdirectories. The size of the XML files varies quite a bit: from 500 bytes to up to 800 KiB. We did not perform a usage analysis on this case.

4.3 Setup

All technical documentation sources used in the study were available in XML format. Documentations I and II were produced with DocFactory, a Content Management System (CMS) for technical documentation supporting, for example, version control, language management, information presentation using standard Web browsers etc. In order to provide usage analysis, we extended the CMS to record a time stamp and the source document for each document visited.

We used VizzAnalyzer to perform all the analyses and export data to be visualized. Several external tools, including Microsoft Excel and the yEd graph viewer were used to visualize the data from VizzAnalyzer.

We performed all the analyses on all the documentations with the exception of Documentation III. Since it was not produced using DocFactory, we could not perform a usage profile.

4.4 Results

The main goal of the study was to evaluate the feasibility of the approach, i.e., the efficiency and effectiveness. In general, the analyses performed well, and found potential quality defects there were detectable from the visualizations. We present part of the results below.

The clone detection determines the similarity between two documents by first comparing the text on paragraph level and then the XML structures. The result is a percentage that indicates the relative size of the two documents that is unique. In Documentation III the clone detection found that of the 913 XML documents, only 6 were completely unique. 20 documents were complete clones of another document, and the remaining documents were clones to some extent. On average, a document was 54% unique.

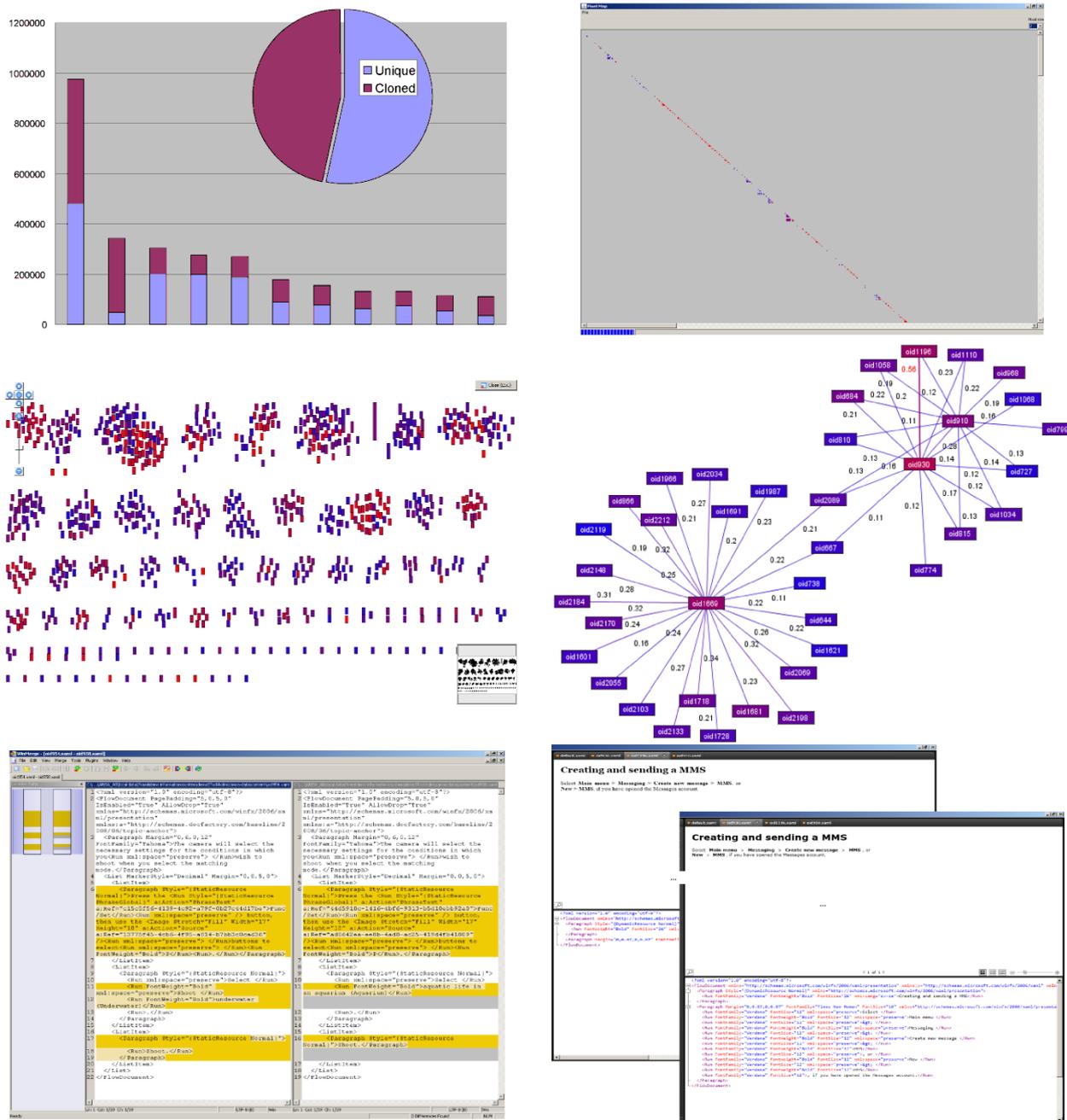


Figure 1: Visualizations of clone detection results, from top left to bottom right: (a) statistics view, (b) pixmap view, (c) cluster overview, (d) detailed cluster view, (e) document comparison view, (f) document editor view

Figure 1a depicts the fraction of cloned vs. unique text in Documentation III. Figures 1b, 1c, and 1d present the result of the clone detection using pixmap and clustered views, respectively. The pixmap view presents a pixel matrix, where each pixel corresponds to a document pair with some degree of similarity. The color schema is used to encode the degree of similarity; blue represents a low degree, while red represents a high degree. Figures 1e and f shows how the clones are presented using a tool to compare texts and finally using the editor of the CMS.

The clone detection analysis is efficient. Running it on the quite large Documentation I takes less than 30 seconds. For smaller documentations, such as Documentation II the analysis is instantaneous.

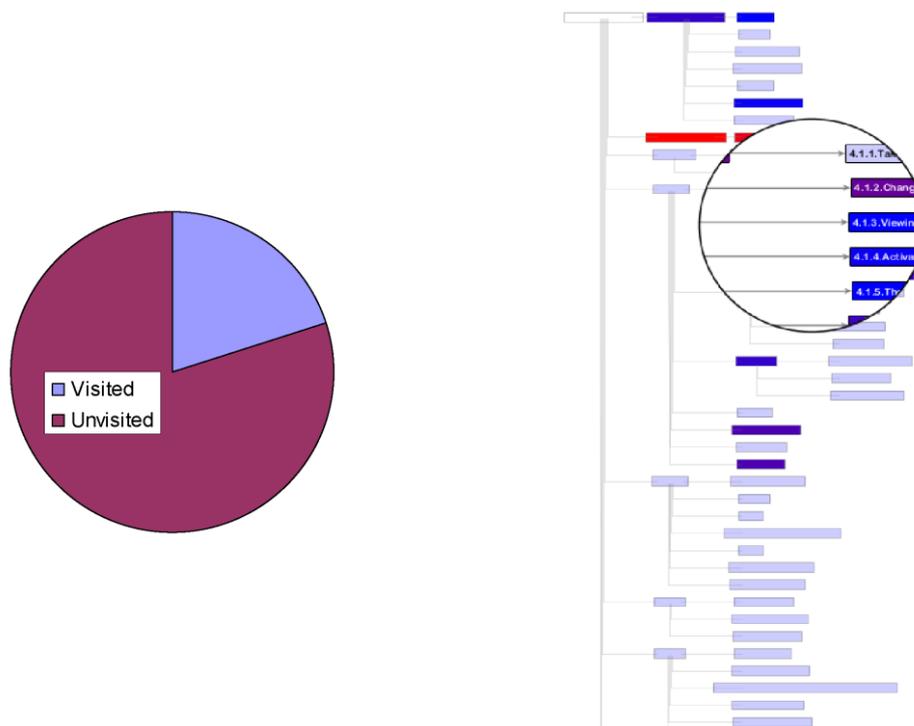


Figure 2: Visualizations of usage analysis results, left to right: (a) statistics view, (b) graph view

The usage profile detects which parts of a documentation a user visits. For Documentation III we asked a user to take a picture using the camera of the mobile phone and send it as an e-mail to an existing contact. The user spent 2 minutes browsing the documentation and 22 document visits recorded. Figure 2a shows the coverage (based on a subset of Documentation I that contains 72 XML documents). Figure 2b shows the documents visited and the time spent on each document. Each box is a document, and the color represents how long the user spent on that document. Dark blue represents not visited and red is the document the user spent the most time on. It is interesting to note that the red document that the user spent the most time on is unrelated to the task he or she was performing.

We performed the usage profile on Documentation II as well. This time the user spent 30 minutes browsing the documentation and generated 228 document views. In both cases, the analysis is very fast. For the larger case, it took less than a minute to analyze the output (not including the time the user spent browsing the documentation).

The structure analysis determines the structure of the document. We applied structure analysis to Documentations I and II. Figure 3a shows the hierarchy of Documentation I. Each box is a document; the color represents which section it belongs to and the height represents the relative document size. The structure analysis reveals a potential quality defect; two of the sections are much smaller than the others, containing only 1 and 2 documents, respectively. The documents are not much larger than the others, so the two sections are significantly shorter than the rest. Figure 3b shows an analysis of the cross-references in Documentation III. Each box is a document, color represents the section and the distance between boxes represents the amount of cross-references. The top shows all the documents

and the bottom shows a zoom-in on one of the clusters. One of the clusters has a large amount of cross-references between different sections. This can sacrifice the cohesion and may make the document less readable.

Structure analysis is performed as part of the usage profile, and represents the significant part of that analysis. So, the performance is similar to usage profile analysis.

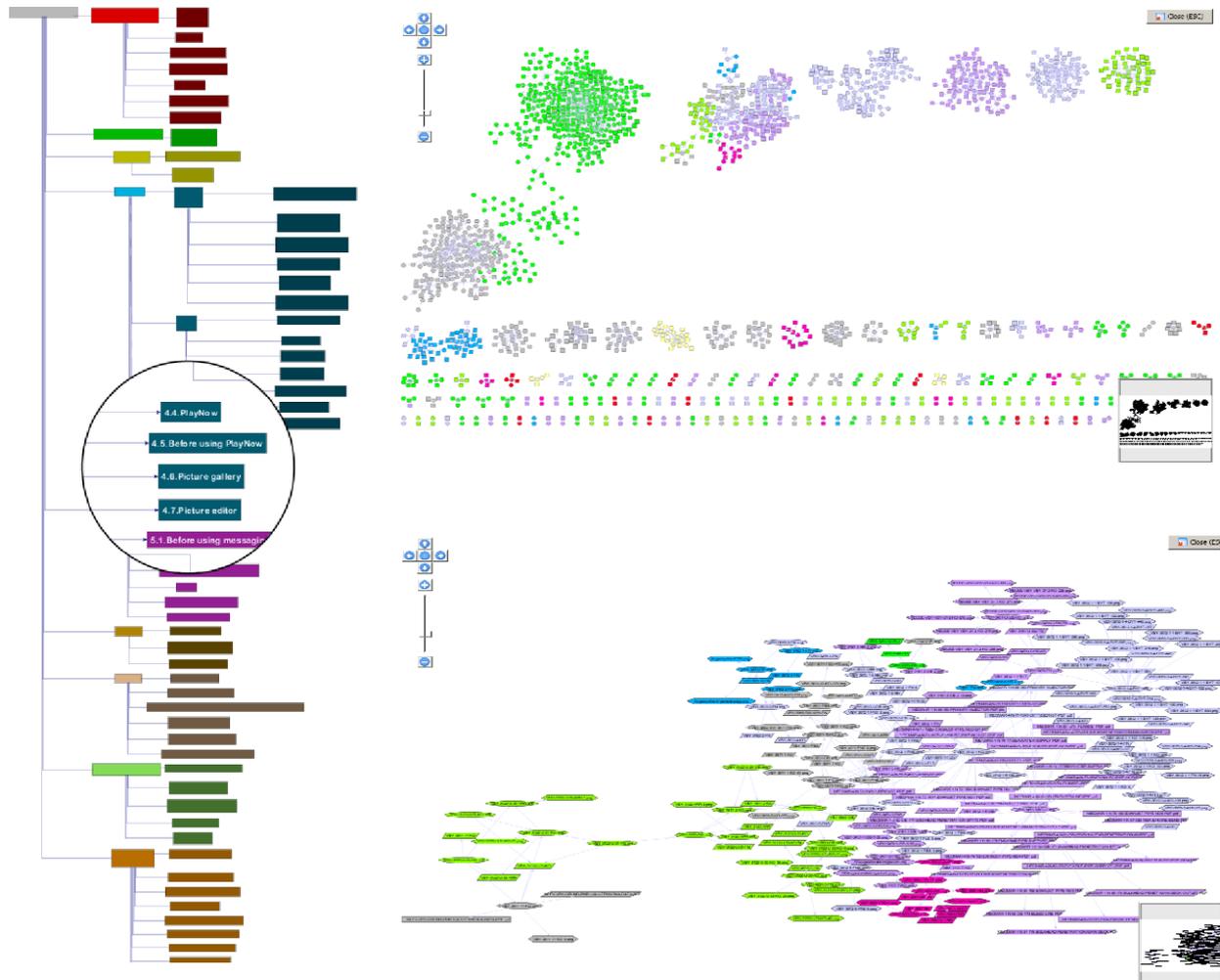


Figure 3: Visualizations of structure analysis results, left to right: (a) document hierarchy view, (b) cross-reference overview (top) and detailed view zoomed in the overview (bottom)

The meta-information analysis analyses the use of meta-information in the documentation. Each document can have a number of “applicability” tags. These tags can be considered topics, or specific terms that describe the document. For example, in Documentation I tags are used to describe topics such as “Call” or “(SIM-) card”. These tags can in turn be categorized, and “call” belongs to the “Actions” category, for example.

We applied the meta-information analysis to Documentation I and Documentation III. The analysis is instantaneous, even for large documentations. Figure 4a shows the applicability tags of Documentation I grouped by categories (color coded). The analysis shows that each tag belongs to exactly one category. Figure 4b shows the applicability tags for each document. The colors still represent categories. The analysis shows that tags of some categories were concentrated to certain subsections, while tags of other categories are spread throughout the documentation. For example, tags in the category “Troubleshooting” were concentrated to the documents related to troubleshooting, while tags in the category “Actions” were spread throughout the documentation. While both observations can be considered potential quality defects, the designed confirmed that they were intentional and part of the design.

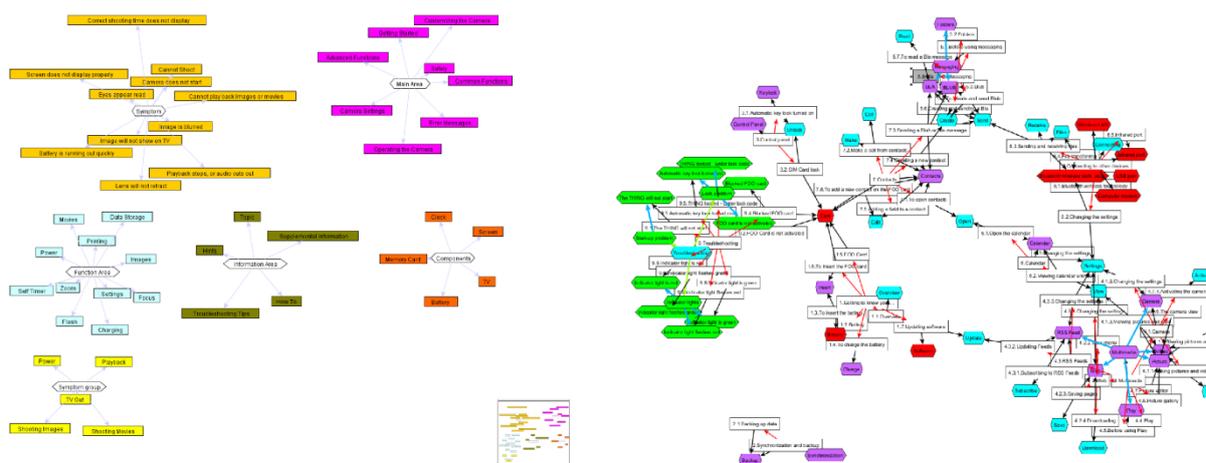


Figure 4: Visualizations of meta-information analysis results, left to right: (a) meta-information structure view, (b) view of relation meta-information to information

5. Conclusions and Future Directions

This paper investigates methods to present and assess the quality of technical documentation. A major part of finding such methods is to define what quality actually means, and based on this definition, find good ways to assess it. Given the structured nature of technical documentation and the reliance on markup languages such as XML, we start from a software quality perspective with a focus on metrics that can be automatically assessed. The differences of how quality is approached by information and software, respectively, makes it hard, or even impossible, to fully rely on metrics that can be automatically assessed. There is a need for human intervention.

Visual analytics is an approach to data analysis that combines human expertise, automated machine analysis and visualization. Data is produced, analyzed and transformed to a format suitable for visualization. The visual representations are interactive, and can be manipulated by human experts. The human expert can guide the visualizations and find not only that which can be predicted by models, but also the unexpected.

Inspired by the idea behind visual analytics, we develop an approach to quality assessment that is based on automated analysis and interactive visualization. We rely on the software quality analysis tool VizzAnalyzer, and extensions to the DocFactory CMS. We perform 4 analyses on 3 real-world technical documentations, together consisting of more than 15,000 documents. The results are visualized using 12 different visualizations.

By combining the visualizations with human expertise, we are able to identify possible quality defects. For example, an analysis of the cross-reference structure of the documentation of a warship shows that there are several cross-references between different sections. This can be an indication that the documentation is hard to read, and that the sections lack cohesion. We were also able to identify a potential problem with the structure of the documentation related to a mobile phone. An analysis of usage profiles shows that users visited parts that were completely unrelated to the task they were performing. This suggests that either the descriptions or references are confusing, or even wrong. The work presented in this paper is based on an initial study using a limited number of analyses and visualizations. We are working on including more analyses and visualizations, both by adapting existing ones defined for software quality assessment and by creating new ones specific to technical documentation quality assessment. In the latter case, we are particularly interested in automatic analyses that consider natural language. One particular use case that is highly interesting is to be able to assess the quality of translations.

Another future direction is to evaluate both the visualizations and how we interact with them. At the moment, we rely on basic scientific and information visualizations, such as pie charts and graphs, although interactive down to the data level they were generated from. We need to evaluate the effectiveness of the existing visualizations and investigate state of the art scientific and information

visualizations. We currently rely on external tools such as Microsoft Excel and yEd to visualize the output from VizzAnalyzer. While many of the visualizations are interactive, in the sense that we can zoom, pan and use links to load other visualizations, there is still room for much improvement. We need to improve the integration between the CMS, VizzAnalyzer and the visualization tools.

This approach to technical documentation quality assessment suggested in this paper is exploratory, and based on an understanding of quality as a notion rather than a specific quality model. In effect, we are currently investigating quality models for technical documentation. In one of our efforts, we correlate positive and negative quality attributes to properties of the technical documentation. For example, there is often a correlation between the ease of understanding and the total size. The ease of understanding decreases, as the size grows either too large or too small. Our work on establishing quality models and using visual analytics to assess quality benefit from each other. The exploratory nature of visual analytics can discover correlations, and a quality model can support analyses and visualizations, as well as the human expert.

Acknowledgment

Anna Wingkvist is supported by the Swedish Research School of Management and IT (MIT), Uppsala. Also, 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). Further, our gratitude goes 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. Please note that an earlier version of this material, see Wingkvist et al. (2010), was published in the Proceedings of the 5th European Conference on Information Management and Evaluation (ECIME 2010).

References

- Cook, K., Earnshaw, R. & Stasko, J. (2007), 'Guest editors' introduction: *Discovering the unexpected, computer graphics and applications*', 27(5), 15–19.
- Crosby, P. B. (1979), *Quality is free: the art of making quality certain*, McGraw- Hill, New York.
- DocFactory (2010), 'Docfactory'. <http://www.sigmakudos.com/data/services/docfactory>, accessed April 2010.
- Ge, M. & Helfert, M. (2007), "A review of information quality research — develop a research agenda", in *Proceedings of the 12th International Conference on Information Quality*.
- ISO (2001), 'ISO/IEC 9126-1 Software engineering - Product Quality - Part 1: Quality model.
- Juran, J. (1998), *Juran's Quality Control Handbook*, 5th edn, McGraw-Hill.
- Kahn, B. K., Strong, D. M. & Wang, R. Y. (2002), 'Information quality benchmarks: product and service performance', *Communications of the ACM* 45(4), 184–192.
- Keim, D., Mansmann, F., Schneidewind, J. & Ziegler, H. (2006), 'Guest editors' introduction: Discovering the unexpected, computer graphics and applications', *Proc. of Information Visualization (IV 2006)*, IEEE 27(5), 9–16.
- Klein, B. D., Goodhue, D. L. & Davis, G. B. (1997), 'Can humans detect errors in data? Impact of base rates, incentives, and goals', *MIS Quarterly*. 21(2), 169–194.
- Löwe, W. & Panas, T. (2005), 'Rapid construction of software comprehension tools', *International Journal of Software Engineering and Knowledge Engineering* 15(6), 995–1026.
- McCall, J. A., Richards, P. G. & Walters, G. F. (1977), *Factors in Software Quality, Technical Report Volume I*, NTIS, NTIS Springfield, VA. NTIS AD/A-049 014.
- McConnell, S. (2004), *Code Complete*, Second Edition, Microsoft Press, Redmond, WA, USA.
- Stvilia, B., Twidale, M. B., Smith, L. C. & Gasser, L. (2005), Assessing information quality of a community-based encyclopedia, in *Proceedings of the International Conference on Information Quality - ICIQ 2005*, pp. 442–454. URL: <http://mailer.fsu.edu/bstvilia/papers/quantWiki.pdf>
- Thomas, J. & Cook, K. (2005), *Illuminating the Path: The Research and Development Agenda for Visual Analytics*, National Visualization and Analytics. Thomas, J. & Cook, K. (2006), 'Computer Graphics and Applications', 26(1), 10–13.
- VizzAnalyzer (2010), 'Vizzanalyzer'. http://www.arisa.se/vizz_analyzer.php, accessed April 2010.
- Wang, R. Y. & Strong, D. M. (1996), 'Beyond accuracy: what data quality means to data consumers', *Journal of Management Information Systems* 12(4), 5–33.
- Wingkvist, A., Löwe, W., Ericsson, M. & Lincke, R. (2010), Analysis and visualization of information quality of technical documentation, in 'Proceedings of the 4th European Conference on Information Management and Evaluation', pp. 388–396.