Fraunhofer Institut
Experimentelles
Software Engineering

Integrating (Software) Documentation and Training

Authors:
Ines Grützner
Ludger Thomas

Submitted for Publication to
SIGDOC 2006

IESE-Report No. 061.06/E
Version 1.0
June 2006

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach (Executive Director)
Prof. Dr. Peter Liggesmeyer (Director)
Fraunhofer-Platz 1
67663 Kaiserslautern

Abstract

User Documentation and Software Training have several aspects in common: They both target on supporting humans to install, use, apply or repair specific products or components. They also use identical media formats like printed or electronic documents or media, such as animations. Even if they have a lot of similarities, in practice lots of potentials of a better integration are unused, since both documents are developed by different teams, using different technologies and development processes. In this paper, an approach towards a better integration of User Documentation and Training Media development is presented. This approach is underpinned by a sound development process and a set of related technologies.

Keywords: Software User Documentation, E-learning, DocBook, XML, Single-Source-Publishing, Open-Source-Software, Training

Table of Contents

1 Introduction

E-Learning and User Documentation are different sides of the same medal. A case for illustrating the relationships between both areas is the roll-out of a new software system in a company since it includes mainly the training of the prospective users by the beginning of the usage and the continuous support during the whole lifetime of the software system. Both activities are underpinned by user support and training media to make them more effective and efficient. Such support media include various kinds of documents (e.g., OS specific) in various media formats (e.g., online, printed, mobile) for various target groups (e.g., end user, product support/hotline, developer). Training media may be structured alongside similar dimensions, such as a self-paced online training for end-users using the product with a specific OS.

Even if the various programs and media mentioned use similar technologies, similar data and share common objectives (e.g., enable users to using the software) as well as structural characteristics [15], they are often developed in parallel and without mutual interactions ([10], [11], [13]). Often, the development of the manual in a company is mainly done by the Technical Documentation Department, whereas the training media are being developed by the Department for Education and Training. By doing so, several instances of the same information are being developed in parallel and hence have to be maintained separately. Every change of the system, e.g., changing symbols, causes enormous maintenance efforts and might lead to inconsistencies as the same information has to be updated several times. Furthermore is the development of high-quality training media a cost and work intensive process [4].

It has been proposed various times that E-Learning is a part of technical documentation and e-Learning might become a new area of responsibility for technical writers. Apart from these programmatic statements, it seems that in theory and practice both areas are still separated from each other and conversion is proceeding slowly. In this article, we describe our approach on a integrating the development of User Documentation and User Training in terms of authoring and development processes and support by a common technical framework. This approach has been developed in the context of a typical industrial software rollout project. A special focus has been put on the development of well structured, educational sound high-quality courseware, since most of the approaches coming from the publishing domain have a somehow over simplified look on how people learn.

2 Software Documentation and e-Learning

Gery (1991) introduced the term "Electronic Performance Support System" (EPSS) to describe technical systems which integrate information, tools and methodology in order to support the user fulfilling his tasks. An EPSS is "an integrated electronic environment that is available to and easily accessible by each employee and is structured to provide immediate, individualized on-line access to the full range of information, software, guidance, advice and assistance, data, images, tools, and assessment and monitoring systems to permit job performance with minimal support and intervention by others." According to this wide definition, EPSS can include both documentation and training related activities. Especially if taken into account that a wider view on EPSS might include non-technical aspects, such as user support, too, EPSS might serve as a major term for the description of a variety of activities happening before, during or after the roll-out of a new software system.

In reality, however, there are still many steps to take to bring such an integrated view on EPSS to life. Looking, for example, at User Documentation and media for User Training, both document types are often still developed separately ([10], [11], [13]) using different development rules and different information bases. The main reasons for the separate development of both document types are slightly different goals and usage scenarios of both media: While Training Programs and Media are used to *train* humans *systematically* to use a specific software system during and after the roll-out of the system, User Documentation is developed to *support* humans during their work with the software or in fulfilling their user tasks. Support in this context includes the provision of large amount of more or less structured information and search facilities (e.g., index, full text search, FAQ), whereas training is mainly based on the idea of well structured and guided access to information, examples, simulations and exercises.

As a consequence User Documentation is written by technical authors following the rules to structure and present technical information and the processes of Technical Documentation. Often, the documentation rules and processes applied by the authors are company-specific. An example of such documentation processes is the IBM documentation process for a typical paper-based User Documentation [3]. A more general approach to User Documentation is presented by Ament [8], which also uses a promising technique to produce different document types from a single information base. However, the approach is only defined to produce different types of User Documentation and does not include the production of Training Media.

The development of paper based as well as electronically Training Media, on the opposite, follows the rules and processes of instructional media development ([2], [14]). Often, the instructional media development approaches presented focus on a single dimension of the development process (for example, pedagogical design and development vs. technical production), which is a major drawback to the development of high-quality Training Media, which requires cooperative efforts of experts from many different disciplines with different views on the Training Media and its development. The *IntView* method for engineering-style training development [4] solves this problem by integrating the tasks of each required discipline in a joint development process. Nevertheless, IntView only covers the production of Training Media and not the development of User Documentation.

Since the development of User Documentation and Training Media follows different processes, it also uses separate information bases. This fact is reinforced by the different rules for structuring and presenting information in these document types. That is, although User Documentations and Training Media contain similar information, this information is presented in different structures using different elements, media types, and designs. Therefore, even in small and medium enterprises, where the different types of User Documentation and Training Media are written by the same authors, separated information sources are used to develop both types of media and information is stored redundantly. This redundancy causes increased development as well as maintenance effort and often decreases the quality of the document types because of information inconsistencies.

3 Towards an Integrated Documentation and Training Media Production Process

A better integration of Documentation and Training Media Production requires not only a common set of data and content, but also a common development process which integrates all relevant roles. Since most of the existing and documented development processes of both User Documentation and Training Media focus on only one specific type of document, they are not applicable for any integrated development of documentation and training media. In the following section we work out the foundations of an integrated development process for documentation and training media production. The benefits of such an integrated approach are that existing synergies in the content and the production of both document types could be used more easily and hence help to reduce the overall production effort.

The process for the integrated production of User Documentation and Training Media proposed in this article is based on the IntView method (see Figure 1) which comes from the domain of courseware engineering. The IntView method was chosen because

- it already supports the production of Training Media, in particular the production of courseware.
- it does not require a specific technology to implement courseware but leave it to the producer to select a suitable technology. Therefore, the activities for Single-Source Publishing can be easily integrated.
- it already supports systematic development following software engineering principles.

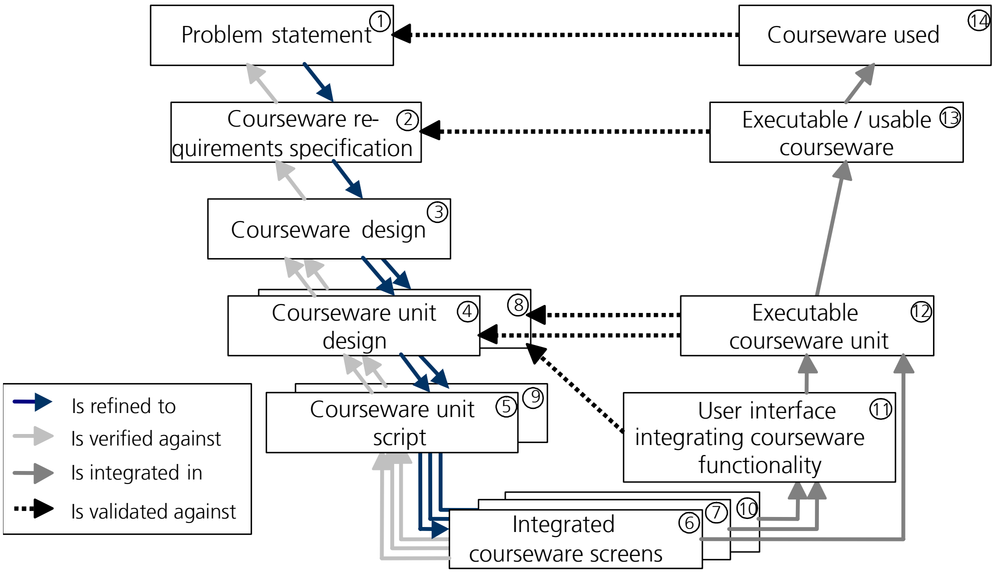
Problem statement ①
Courseware used ⑭
Courseware requirements specification ②
Executable / usable courseware ⑬
Courseware design ③
Courseware unit design ④
⑧
Executable courseware unit ⑫
Courseware unit script ⑤
⑨
User interface integrating courseware functionality ⑪
Integrated courseware screens ⑥
⑦
⑩
Is refined to
Is verified against
Is integrated in
Is validated against

Figure 1: The product-centered IntView life-cycle model

One of the main characteristics of the IntView engineering process is its ability to integrate all relevant roles and viewpoints into all relevant phases of the production and lifecycle of a product. This is essential for an integrated process since various roles and disciplines are involved in the development process. According to IntView, the production of User Documentation and Training Media starts with a specification of the requirements for all documents to be developed (see Figure 2). The method applied during this specification is the task oriented IntView requirements specification method [17]. In this phase, requirements valid for *all* documents to be developed as well as the document-specific requirements are elicited from *all* stakeholders and documented in a specification.

The requirements specification is the prerequisite for the development of separate designs for all documents. Separate designs are developed because the documents are to be used in different contexts during the work with the documented software and therefore require different content structures, document designs, and functionalities. Nevertheless, the content structures have to be designed as similar as possible in order to allow producing the documents in a Single-Source Publishing environment.

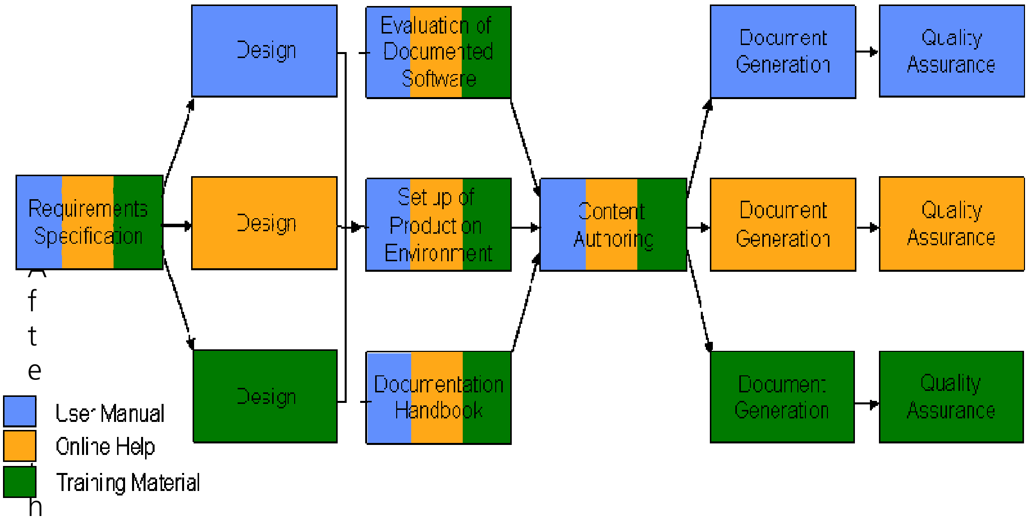

Figure 2: Product-centered development process for the production of User Documentation and Training Media

After the development of all concepts, the software to be documented is evaluated in order to elicit the content of the documents, the Single-Source Publishing environment is set up, and writer's handbook is developed and maintained in parallel activities. The software evaluation aims at the elicitation of all software functionalities and of the user interface elements. In this activity, a list of the terminology used in the software is produced besides the functional and user description. If the elicited information is arranged according to the content structures developed in the design phase, it corresponds to courseware unit design of the *IntView* method. The set up of the Single-Source Publishing environment is comparable to the user interface implementation of the *IntView* method. During the set up, the tools of the production environment are installed and integrated to a tool chain. Furthermore, the document designs are implemented as XML templates and XSL transformations. The templates and transformations are also elements of the writer's handbook, which is developed and maintained in the third activity. This writer's handbook, which is a main product of User Documentation processes, comprises all rules, regulations, and processes. Therefore it is important tool for the systematic and consistent document production [18]. It contains not only the rules and processes specified in the requirements specification and the document designs; it also integrates the rules, regulations, and decisions made during the production effort. The writer's handbook is therefore the common memory of the whole documentation team, which is set up and maintained over several projects and enables the reuse of all rules, regulations, processes, and designs from former projects.

The production of the different documents finishes with typical Single-Source Publishing activities like the authoring of the content as well as with the separated generation and quality assurance of the documents. The authoring of the content, maybe already as XML files in the Single-Source Publishing environment, is equivalent with the Courseware Unit Scripting and the Implementation of Courseware Screens in the *IntView* method. The production of the documents ends with generation of the documents in the Single-Source Publishing environment and with the assurance of the document quality, for example, in an evaluation [18].

4 An OpenSource and XML-based Single-Source Production Environment

Traditional approaches to documentation were based on the paradigm, that each type of documentation is developed separately, by using Desktop Publishing Systems as well as clearly distinguished roles, e.g., someone who developed the handbook, an other person or team who the help system, and a third group of people who cared about the user or the maintainer training and support. By using such an approach, practitioners faced the problem, that same or similar information must be generated and stored several times in several instances and data sources. This lead to an increased number of maintenance efforts since every instance of the same information has to be maintained severalty. Simple mathematics show how an increased number of options (versions of documents) lead to multiple numbers of versions to maintain (see Figure 3). This example is based on the case that a company might want to provide various documents for *one* of their products.

- DT = Number of Document Types; e.g., online help, manual and courseware
- OF = Number of Output Formats; e.g., online, printed, mobile
- TG = Number of Target Groups; e.g., end user, support/hotline, developer
- OS = Number of Operating Systems; e.g., Windows 2000, Windows XP

The simple formula DT × OG × TG × OS, in this case 3 × 3 × 3 × 2 = 54, shows that this fictive company might run into problems when they try to maintain such an amount of document versions with a traditionally approach to publishing.

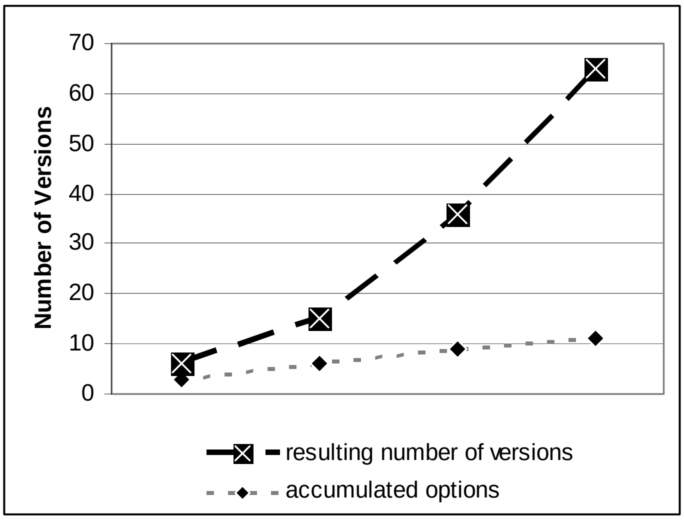

Figure 3: Relationship between the number of options and the resulting number of document versions

The single-source approach to documentation bases on the idea of using *one* common data source *for all* kinds of documents, such as manuals, help systems and training media. Such a data source could be a common data base or, as in this case, a collection of XML-Files. Since XML can be generated by various technologies, such as editing source-code, XLS-transformation or database export-to-XML functions, it has become *the* central element in such a single-source approach to documentation.

Apart from the paradigm, that information has to be stored only once in a common data source, such a single-source approach is based on the idea of separating the main dimensions of text: content, structure and layout. Various XML-technologies support this by providing a framework for defining new semantic markup dialects for content, structures and layout. Modern XML-technologies such as XML-Inclusion provide native support for the multiple usage of same information across various documents.

DocBook as one of the most mature and best supported dialects in this area, is in first manner such a standardized XML dialect which consists of more than 400 elements, some of them generic (e.g., section, chapter, glossary), and some of them specialized for software documentation (e.g., funcsynopsis). Since it has been established in the early 90s by a group of O'Reilly, it increased its popularity. Nowadays DocBook is a standard and used by a continuous increasing number of companies and (open-source) projects worldwide (e.g., SUN, linux documentation project). Over the years various related tools, activities and services evolved. Most important is a free XSL-Distribution, with which DocBook can be transformed into viewable Formats such as, XSL-FO/PDF, HTML and Online-Help-formats.

Since DocBook is focusing on software documentation, its usage for producing other kinds of documents such as Training Media has been underestimated over the last years and there are only a few applications in this area. Nevertheless several people use DocBook for their teaching and a few applications evolved which use DocBook in the area of e-Learning. Compared to specialized markup for e-Learning, such as the IMS standards or the various Educational Markup Languages (EML), DocBook doesn't provide any specialized educational elements. Furthermore an integration of interactions such as quizzes marked up with IMS' QTI standard might be applicable once the DocBook schema can be better customized to individual needs (V5). The strength of DocBook is that it is a universal format, well suited for help systems and manuals and with strong support by various tools as well as an active community. In practice some of the weaknesses of DocBook for modeling training related media, don't count that much as its strength do. One can find workarounds, e.g., customizations of the XSL-distributions or the schema, which cope with the deficits. For example, Thomas and Ras [15] showed that DocBook markup can also serve as a basis for generating current e-Learning standards.

From a technical perspective, a DocBook and open-source software work hand in hand. Such an approach on single-Source publishing and e-Learning production can be structured alongside a four layer model [15] (see Figure 4):

- The *Storage Layer* focuses on data collection and data management. The content is described on a semantic level (e.g., RDF, XML) and kept in a media-neutral, purpose-free way. No further structure and layout elements have been applied to the content.
- The *Composition Layer* contains the composition process. In this key process, the content objects are selected and eventually transformed into learning objects. A structure is given to the collection of content objects based on the purpose, and the objects are linked based on their semantic relationships or on given links. At the end of the process in this layer, a structured, customized (profiled) document, containing the structure, the content and all relations between structure and content elements, is available.
- The *Transformation Layer* applies the layout to the composed document. It is transformed into the final media format (e.g., HTML, PDF), which is ready for distribution.
- The *Publication Layer* focuses the processes to make the documents available to the target group and to collect feedback for the continuous improvement of the documents.

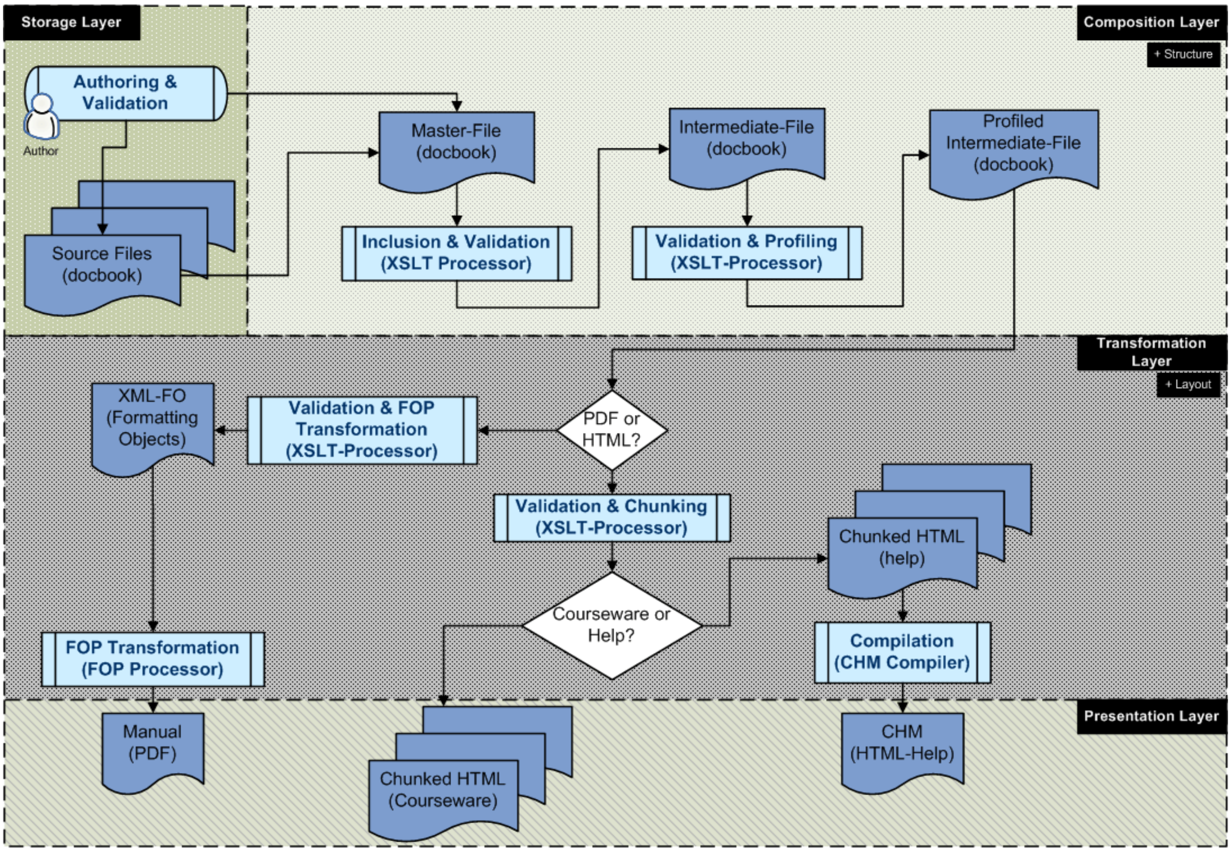
Storage Layer
Composition Layer
+ Structure
Authoring & Validation
Author
Source Files (docbook)
Master-File (docbook)
Inclusion & Validation (XSLT Processor)
Intermediate-File (docbook)
Validation & Profiling (XSLT-Processor)
Profiled Intermediate-File (docbook)
Transformation Layer
+ Layout
XML-FO (Formatting Objects)
Validation & FOP Transformation (XSLT-Processor)
PDF or HTML?
Validation & Chunking (XSLT-Processor)
Chunked HTML (help)
Courseware or Help?
Compilation (CHM Compiler)
FOP Transformation (FOP Processor)
Presentation Layer
Manual (PDF)
Chunked HTML (Courseware)
CHM (HTML-Help)

Figure 4: Four Layer Model, based on DocBook

The four steps layer production model is generic and can be used for any kinds of contents (E-Learning contents, documentation). Its main technology is the XSL-transformation (XSLT), which is since the end of the 90s in the status of a recommendation by the W3C. Several commercial and open-source tools use this technology to transform on XML dialect into another dialect or into a viewable format such as HTML. In case of the generation of the popular Portable Document Format (PDF), a specific sub-standard of XSL, XSL Formatting Objects (XSL-FO), must be used first. XSL-FO technology provides markup for the textual description of page and document layout (e.g., how the page number in the footer looks like) and can be transformed into PDF by using a (commercial) PDF-Rendering engine, e.g., RenderX XEP.

As Thomas and Ras [15] pointed out, DocBook can be used not only for generating printed-like documents and web-based help systems, but also generating modular e-Learning contents which are enriched by meta-data and descriptions of educational scenarios. Since the used technologies such as XSLT are widespread with a variety of tools available, DocBook might serve as one way to generating several components of an EPSS from a single data source.

5 Experiences

The described approach for the integrated, systematic development of media for the user support as well for User Training has been successfully used and refined in some internal projects as well as in external projects with industry. Even if not really validated from a scientific viewpoint, we can draw some preliminary conclusions which underpin experiences reported for the area of single-source-publishing [5].

5.1 Saving Potentials

For the area of single-source publishing, practitioners report about cost savings of 30% compared with a traditional documentation approach. [16]. In the past and recent context, these figures could not be scientifically validated, but it is to expect that similar results might be obtained by a deeper analysis. It is to expect, that most of the savings will come through the more efficient maintenance of the three most relevant media, manual, help system and courseware. Especially if similar documentation and learning projects are being conducted once again, there will be cost savings in terms of reuse of the reuse of the production environment, the underlying processes, and the already existing knowledge and experiences.

5.2 Keeping media up-to-date

Being up-to-date and keeping pace with the continuous developments is essential for any kind of documentation. In practice this means, that the textual and medial descriptions documentation as some kind of depending, secondary product, continuously have to keep up with the original software products development. Since software is a fluent product and changes dramatically, it is important but most difficult to start documentation as soon as possible but as late as necessary and with respect to the scheduled release planning. If documentation and media development starts too early, this might result in increased rework since documented and undocumented changes have to be incorporated into the documentation. Especially undocumented changes, which happen in practice even in well documented development projects, cause overhead, as most of the software continuously has to be checked for such relevant changes. The described approach can not completely prevent this, but helps keeping the document up-to-date by modifying only one source file for various kinds of media.

5.3 Systematic Approach

Especially when working in teams, technical writers as well as instructional designers need a systematic, engineering-style approach on document development. In our approach we introduced a central writer's handbook, which helps to keep the various media and documents constituent by defining common rules. It includes e.g., rules how to use functional entities, language, terminology or rules for authors how to use XML markup. In general this works fine, but especially in bigger projects the handbook suffers from the problems like other documents: It has to be well structured, easy to use, maintained and updated to be really useful. A further problem practitioners face, is the time consuming control of the various document's quality. Since there are only a little common criteria, every document has to be checked against specific criteria, e.g., the rules defined in the writer's handbook, and with a specific viewpoint by itself.

5.4 Information Modeling and Re-Use

One of the central advantages of the single-source approach on publishing is, apart from the improved maintenance, the re-use of information across several media formats. By using XML, various kinds of formats can be incorporated into one common data basis. The degree of re-use depends mainly on how strong the intended information models are linked and how much information can be re-used in same or even similar ways. For example, a procedure in a handbook might look different and provides different data as the same procedure in a courseware, but both build upon the same information model and markup of a procedure. Using XML provides authors with flexible techniques for re-using information on every possible granularity, e.g., by using Entities of the XML-Inclusion technology. In practice this isn't always trivial, since modeling of variants and the re-use of information entities in DocBook has some restrictions, like for example the unique id-paradigm which keeps authors from re-using information with an id-attribute. A further problem practitioners may face is related to hierarchies: Manuals and Help-Systems can provide deeply nested hierarchies, whereas courses are most times structured along a mostly flat sequence of units and lessons. Since media used for learning have to provide interactive exercises, quizzes, simulations and so on, these types of information are not relevant for classic software documentation and therefore not foreseen in any XML dialect.

5.5 Supported by Open-Source-Software

As mentioned earlier, Open-Source-Software (OSS) supports the integrated approach on building software documentation and training media from the same source in many aspects. The XML dialect DocBook has been designed especially for documenting (non-gui) software systems, but is flexible enough to work out for other types of software and media formats. Since DocBook is a standard and well introduced in the Open-Source-Scene, many projects and individuals provide free software for building a toolchain, authoring DocBook markup and transforming the data into various output formats. Since WYSIWYG is a established paradigm in the publishing world, authors have to get used to exact opposite when writing XML: The layout of a document is applied in later stages and is not relevant on authoring stage. It is more important to build a sound information model from beginning, which provides a separation of structure, content and layout. Since this is a major problem when switching from WYSIWYG to XML, commercial products try to simulate WYSIWYG by providing a visual view on the data.

6 Summary

In this article we illustrated an integrated approach on developing (software) User Documentation and training media from a single data source and by using open-source-software. Such an approach is designed to cope with the growing amount of document versions being developed once an organization is providing intensive roll-out support for their products. Our approach consists of an integrated conceptual and process model, but also takes into account how it can be supported by available tools. We showed that standard XML technologies like DocBook as well as freely available software support the application of such an approach and provide useful possibilities for the systematic, structured and re-use oriented documentation of (software) products.

Nevertheless, the joint development of training media and documentation might be an interesting area for both disciplines, software documentation as well as for education and training. Currently both of them are strictly separated or only loosely linked. This keeps them from using synergies, in terms of shared processes, objectives, principles and content. Future works will have to deal with the efficiency of the approach and future research has to be done to validate industrial applications of the approach. One of the next steps for improving the approach might be the provision of improved workflow support. A further future step might be trying out alternative XML-markup in order to validate if the approach is not bound to specific markup.

References

[1] Walsh N, Muellner L: “DocBook”. O'Reilly, 1999.

[2] Lee WW, Owens DL: “Multimedia-based instructional design: computer-based training, web-based training, distance broadcast training.” Jossey-Bass/Pfeiffer, San Francisco, 2000.

[3] IBM: “Book Design for Non-Designers.” International Business Machines Corporation, 1983.

[4] Grützner I, Pfahl D, Ruhe G: “Systematic courseware development using an integrated engineering style method.” In: Proceedings of the World Congress “NETWORKED LEARNING IN A GLOBAL ENVIRONMENT: Challenges and Solutions for Virtual Education”. Technical University of Berlin, Germany, 2002.

[5] Gersdorf R: ”Eine Content-Management-Architektur für die Umsetzung verteilter Redaktionsprozesse bei der Erstellung wieder verwendbarer Inhalte für das eLearning.“ In: Uhr W, Esswein W, Schoop E (Hrsg) Wirtschaftsinformatik 2003 / Band I. Physica-Verlag, Heidelberg, S 633-652, 2003.

[6] Fraley L: “Beyond Theory: Making Single-Sourcing Actually Work.” *Proceedings of the 21st annual international conference in documentation*, 2003.

[7] Carroll JR, Smith-Kerker PL, Ford, JR, Mazur-Riemtz SA: “The minimal manual.” Human Computer Interaction, 3, 2: S 123-153, 1987.

[8] Ament K: “Single sourcing: building modular documentation.” William Andrew Publishing, Norwich, New York, U.S.A., 2003.

[9] Ausubel DP: “In Defense of Advanced Organizer: A Reply to the Critics.” Review of Educational Research , 48, 2: S 251-257, 1978.

[10] Dicks, S.: “Integrating Online Help, Documentation, and Training.” *Proceedings of the 12th annual international conferende on Systems documentation: technical communications at the great divide* (1994, Banff, Alberta, Canada), pp. 115 – 118. ACM Press, 1994.

[11] Grützner, I., Thomas, L.: "Systematische Entwicklung von Medien zur Benutzerunterstützung und –schulung mit einem XML basierten Single-Source-Ansatz." To appear in *Proceedings of the Multikonferenz Wirtschaftsinformatik, ZTeilkonferenz eLearning* (February 20 – 22, Passau, Germany), 2006.

[12] Gery, G.: "Electronic Performance Support Systems: How and Why to Remake the Workplace through the Strategic Application of Technology". Boston: Weingarten Publications, 1991.

[13] Ramu, J.-P.: "Task Structure Methodology for Electronic Operational Documentation". *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics* (Menlo Park, California), pp. 62 – 68. AAAI Press, 2002.

[14] McCormack C., Jones D.: "Building a Web-Based Education System." Wiley & Sons, New York, 1998.

[15] Thomas L., Ras E.: "Courseware Development Using a Single-Source Approach." Proceedings of the World Conference on Education Multimedia, Hypermedia and Telecommunications, 2005.

[16] Sall, K: "How the US Federal Government is Using XML: One Year Later" *Proceedings of the XML 2004 conference*, 2004.

[17] Grützner I., Paech B.: "Requirements Engineering for Courseware Development." In: Mate JL, Silva A (eds) "Requirements Engineering for Sociotechnical Systems." Idea, Madrid, pp. 170-188.

[18] Fraunhofer IESE: „Erstellen von Dokumentationen für Technical Writer." Lernsoftware, Fraunhofer IESE. Available at http://www.uml-kurs.de/ITW/Produktseiten/einstieg.html (in German).

Document Information

Title:	Integrating (Software) Documentation and Training
Date:	June 2006
Report:	IESE-061.06/E
Status:	Final
Distribution:	Public