ManPro: Framework for the Generation and Assessment of Documentation for Nuclear Facilities

Cristina Olaverri-Monreal*, Carsten Dlugosch, and Klaus Bengler,

Institute of Ergonomics, Department of Mechanical Engineering
Technische Universität München, Germany
Boltzmannstrasse 15, 85747 Garching, Germany
{olaverri,dlugosch,bengler}@lfe.mw.tum.de
http://www.ergonomie.tum.de

Abstract. Nuclear plant operators must act in accordance with a number of requirements which are specified in the technical documentation that describes handling, functionality and architecture of the systems. Due to these many specifications to be considered, it is crucial to create technical documentation in order to improve the operation efficiency of such facilities and prevent human errors whenever possible. In this work, we propose a procedure (ManPro) for the computer-based creation of instruction manuals for the operation of technical systems. The "ManPro" approach is semiautomatic, which underlines its reproducibility and efficiency. Feasibility of the approach and its effects on usability are assessed and outlined.

Keywords: Nuclear facilities, Technical documentation, Mark up languages, XSLT, XML, UML.

1 Introduction

Recent unfortunate events related to safety systems within nuclear facilities have boosted worldwide concern about the downsides of using nuclear energy. Some control systems installed within nuclear facilities have been developed and established many years ago, and therefore no longer comply with the latest technological standards. New components need to replace existing ones in order to maintain the desired safety standards. This also applies to the production of instruction manuals that have been, in some cases, developed over 40 years ago, and have since rarely been updated. Nuclear plant operators must act in accordance with a number of requirements which are specified in the technical documentation.

This documentation describes the specifications that a plant has to meet, as well as its handling instructions, functionality and architecture [1]. The series of standards related to technical product documentation and nuclear power plants have been published by the International Organization for Standardization (ISO) [2, 3]. These

* Corresponding author.

standards describe how to establish which information users need, how to determine the way in which that information should be presented to users, and how to prepare the information and make it available.

The constantly increasing efficiency of computer-based control systems offers a wide range of possibilities for detecting and controlling undesirable operating conditions; it is therefore only natural to take advantage of such new technologies for the benefit of nuclear facility safety. These computer-based control systems require interaction with the facility operator, through their Human Computer Interfaces (HCI). Moreover, the information that these systems provide needs to be depicted to users in the most efficient way to understand electronic messages [4] and facilitate interaction, particularly in situations that poses an immediate risk. Several works have mentioned the fact that the nuclear industry has taken a long time to incorporate computerized systems, even if they have been found to have substantial advantages over the technology that they replace [5]. Instruction manuals have a direct impact on the efficiency of processes within these facilities and the productivity and safety of their employees. Thus, it is important to acknowledge the dynamic nature of these plants handling and functionality instructions for a more flexible representation of documents. We propose a reproducible and efficient procedure for the computer-based creation of instruction manuals to operate nuclear plants systems in this work. To ensure that the manual is searchable, a User Interface (UI) is generated from a modeling specification in the Unified Modeling Language (UML), which defines the components of a system. The information entered by the user is then stored for the further presentation of the document in different formats.

The remainder of this paper is organized as follows. The next section revises related work in the areas of automation and documentation of technical systems. Section 3 presents a detailed description of the methodology used to implement the proposed approach, and section 4 reports on the evaluation of the system. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Automatic Generation of User Interfaces

To automatically generate user interfaces, the information content design has to enable the user to successfully perform the tasks keeping the information reduced to the minimum necessary as suggested in the formal approach and methodology for analysis and generation of human-machine interfaces, with special emphasis on human-automation interaction in [6]. A formal documentation of the software requirements guarantees a precise and correct user interface. Therefore a common practice in the early stages of software development is to build a Unified Modeling Language (UML) system model with UI prototypes that can capture the different aspects of a system [7, 8]. The authors in [9] presented in this context current model-based approaches to user interface automatic generation such as the following:

- The XSI approach, that is based on the segmentation of a model into different sub-models and provides an Entities view, a Use Case view and a User Interface view [10].
- The OO-Method approach that aims at producing a formal specification of a software system in the object-oriented language OASIS through a graphical system model at a conceptual level. This conceptual model is then translated to an OASIS specification [11, 12].
- The ZOOM approach to enable model-driven development through the structure, behavior, and user-interface of an application, providing thus three related notations to describe each of those parts [13].

Our approach uses a formal system architecture model in UML form to represent a correct user interface that map the different system components on the user interface so that the user can then enter the information that will later appear in the user manual.

2.1 Automatic Documentation Generation

Design, documentation and evaluation of safety-critical applications used in nuclear power plants have been addressed in several works. For example, the authors in [14] proposed a software formal documentation approach through both requirements and software design based on the systematic comparison of program behavior with the engineering specifications of the computer system. In the same context, an authoring tool for automating the software generation process has been presented in [15]. The authors investigated how to provide on-line assistance for software professionals based on the information users require. Further, an XML-based approach for generating the architectural documentation of a software system from the implementation code was presented in [16]. The authors identified relevant concepts for the software documentation and extracted them through an analysis of the source code. Additionally, they organized the documentation in a hierarchical form, which they presented in a human-readable format.

In the same XML content, a software solution for a variable information structure display that was based on the combination of markup languages was proposed in [17]. The authors used XML for representing a document structure, which could subsequently and independently be transformed into new ones using XSLT Stylesheets. In [18], the authors also introduced a related approach to an iterative reconstruction process that allows users to create a source code model in a database, and extract the reconstructed architecture through Structured Query Language (SQL) queries. Most approaches to generate software documentation use UML diagrams that include the system design information. For example, an approach to access and review UML software engineering diagrams on the Web-based on markup technologies was the topic in [19]. Additionally, an approach was developed in [20] which ensured a stronger link of user documentation's generation with software's life cycle phase. The approach enabled the transfer of required information from the software's functional specification described in the Unified Modeling Language (UML) into user documentation. Also several techniques have been proposed more

recently which aim to integrate Web-based data into a suitable format for system designers and end users, constructing UML diagrams from XML data [21, 22]. In other works, tools have been proposed to automatically create graphical input dialogs from text specification entered by software developers [23]. These tools differ from our approach in that they automatically extract information from an existing system. Our approach bases on user specifications combining some technologies from these frameworks to generate specific documentation for nuclear facilities semi-automatically through a computer-aided process.

3 Methodology

The main goal of the "ManPro" procedure is to guarantee that information contained in the final user manual is searchable and understandable for the user, and that ambiguity does not exist. The tool adheres to internationalization principles, allowing for multilingual content. In "ManPro", a relational database contains all the tables and relationships needed for the reliable and correct storage of information previously entered by the user through a Web-based form, which has been dynamically created from a UML specification of the system architecture, behavior, structure and maintenance. The descriptive components of the instruction manuals are then stored in XML format for the further compilation of the final document. "ManPro" consists of the following components:

- An XMI file containing the structure for a modeling specification of a nuclear plant system in UML;
- A Web form that ensures accuracy in the content of the documentation through guided questions and predefined fields extracted from the UML system specification, preventing errors. Additionally, information is immediately available and can be accessed from any computer with Internet access. It also assists system developers with templates and descriptions of requirements for nuclear facilities during the documentation writing step;
- A relational database that enables simple data updates, validation and accuracy through error check. Additionally, the storage of data in a database allows the user to cross-check data against existing information and to map already available documents. Information can be easily accessed, visualized and manipulated through SQL. The information available in the database can then be extracted through Java and MySQL queries, creating information related to chapters, sections, graphics, paragraphs or subsections, also including different kind of lists. Document internationalization requirements are also specified in Java;
- An XML document, extracted from the database's information using JDBC and containing the database structure mentioned above that will be further transformed into the final instructions manual version in PDF;
- The final instructions manual in PDF achieved through the XSL Transformation language XSLT;

The tool implementation process is depicted in Fig. 1.

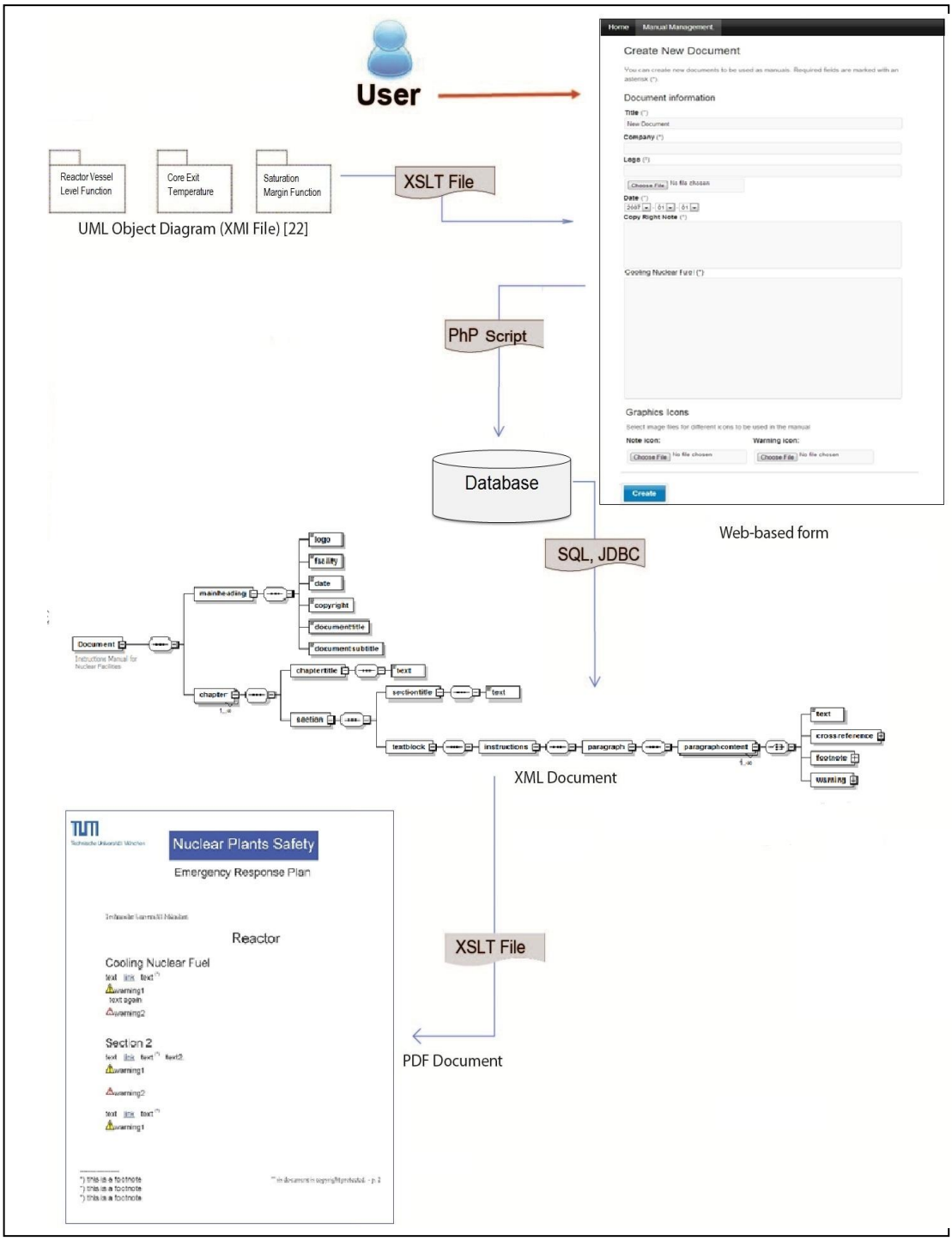

Fig. 1. "ManPro" information processing steps and output files.

The figure illustrates the output files resulting from the database and XML structure after processing the information with the "ManPro" tool. The input UML depicts an example of three possible functions of a Core Cooling System [24]. To edit an instruction manual, the user needs only to accesses the appropriate Web forms and can then create a document, add information or change contents for an existing document, and then save it to the database.

Our approach uses markup languages as the common feature in the development process, integrating XML-based technologies with the rest of the development framework, Java technology, databases, and the Web. We use a modular structure, with each module responsible for one part of the process. The framework begins with a UML object diagram specification of the components of a nuclear power system describing the system's behavior, structure and operating rules.

The input fields for the Web form, which enables text data entry and submission to the relational database is generated from the objects specified in the UML diagram. In the next step, an XML document is extracted for further transformation into a human-readable format.

3.1 Web Form Generation

The XML Metadata Interchange (XMI) format for UML model interchange was accessed through a XSLT file to generate the input fields of the Web form.

Since XMI is XML-based, the file contained the patterns to match the elements within the XMI document and the transformations to apply when a match was found.

To build the Web form for data entry, we used the HyperText Markup Language (HTML), to display information in a Web browser.

Additionally, the open source, server-side PHP scripting language was used to insert the form data into the database and to validate the user input on the server.

3.2 Database Platform

To develop the relational database platform, we used the XAMPP PHP-Apache-MySQL solution. The "document" table is connected to the rest of the tables containing document parts. This architecture allows for reuse of such data for multiple documents. Many-to-many relationships are saved in separate tables, which refer to primary keys with foreign keys of further tables to easily join information for a later data extraction based on queries.

Fig. 2 illustrates the architecture of a sample database section. Table 1 describes the tables' content.

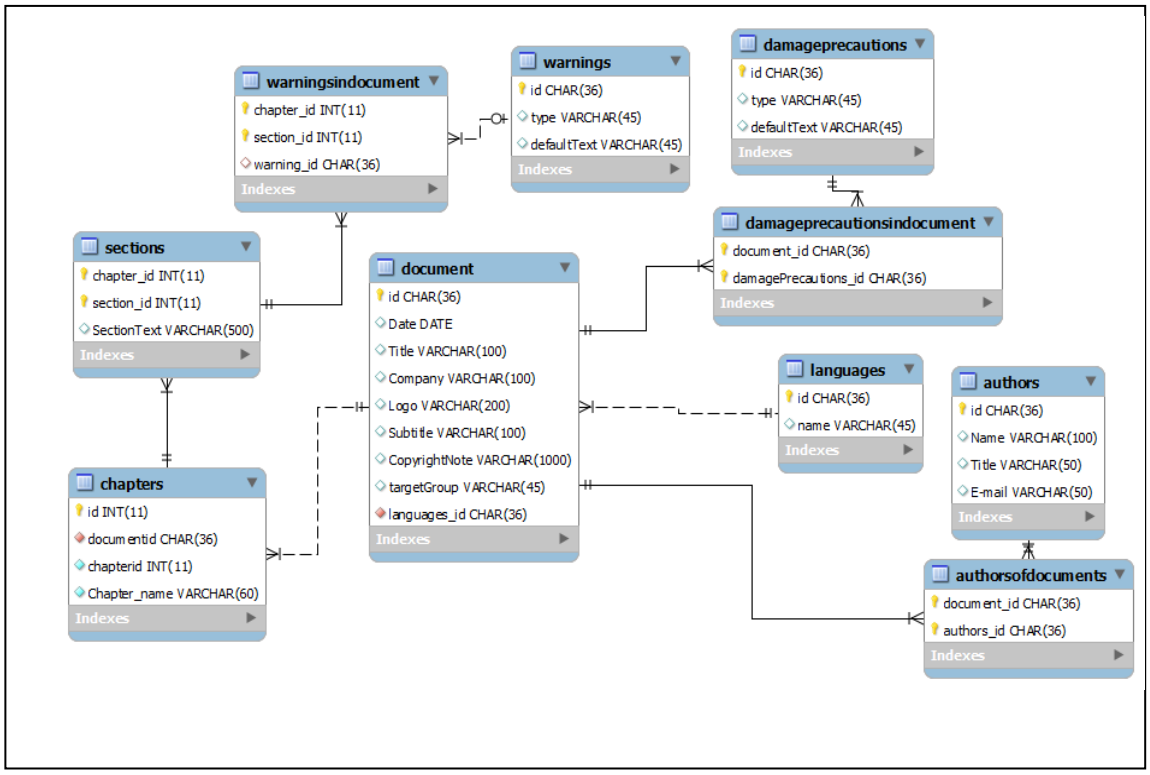

Fig. 2. Database Section Architecture

3.3 Database XML Extraction

To be able to access database information in an XML format, we have built our approach upon the JDBC data access technology.

Through a mapping process, we first selected the information that we wanted to retrieve leaning on an intermediate xml mapping file used by java code to acknowledge the database structure [25]:

1. An intermediate mapping XML file retrieved the necessary information from different database tables through Standard Query Language (SQL) queries.
2. In the same file, we specified the overall structure of the new XML document, describing the tree elements and their attributes, namely the root and the rest of elements that represented the database rows.
3. Additionally, we completed the structure information with the names and contents of data elements, creating new elements and attributes.

An XSL Stylesheet contained the rules for depicting information included in the XML with the appropriate layout. We exploited the resources of XSLT 2.0 for regular expression matching, to describe the text strings patterns we wanted to manipulate. To create the new XML document, the extracted data was made available by parsing the file created above. We then recovered the data element and the SQL statement, determining first the root element and then obtaining data nodes. The data was then stored in the root of the document tree, in order to gain primary access to the document's data, and to be able to create element nodes, text nodes, comments, processing instructions, etc. through the methods contained in the Document Object [26]. A section of the mapping file code is illustrated below.

Table 1. Description of the database architecture.

Table Name	Functionality
document	Main table containing the documents
authors	Table of all authors
authorsofdocument	Many-to-many relationship table between the "document" and "authors" tables
warnings	List of possible warnings
warningsindocument	Many-to-many relationship table between the "document" and "warnings" tables
damageprecautions	List of damage precautions or necessary precautions to prevent damage
damageprecautionsindocument	Many-to-many relationship table between the "document" and "damageprecautions" tables
languages	List of languages
chapters	List of chapters
sections	List of sections

Section of the intermediate mapping XML file to retrieve selected information from a database

```
<?xml version="1.0"?>
<mapping name = "map">
   <data sql="select * from document" />
   <root name="documentInfo" rowName="document">
      <element name="Title">
         <content>Title</content>
      </element>
      <element name="Date">
          <content>Date</content>
      </element>
      <element name="Company">
          <content>Company</content>
      </element>
      <element name="Logo">
          <content>Logo</content>
      </element>
      <element name="subtitle">
          <content>Subtitle</content>
      </element>
   </root>
</mapping>
```

3.4 Instructions manual creation in PDF format

Using XSLT, we performed the final XML document transformation into a PDF. For this step, Apache FOP Java-based open source application was used to generate PDF documents from FO files using XSL. We accessed the tree nodes of the basic XML document and selectively copied the content of the XML document into a new XML document, which represented the final structure. Pattern matching was used to identify the variable options for the copying process.

4 Final Document Evaluation

The manuals that were semi-automatically generated by “ManPro” were intended to serve as a reference for the evaluation of other existing manuals, as they enabled the verification of Quality Assurance (QA) requirements. Thus, we tested the approach to produce technical documentation through an evaluation process that was performed in terms of a level of detail analysis.

The printable PDF file produced in the process was tested by persons familiar with the technology described in the manual in early stage tests to improve the tool through a formative evaluation early in the design process. Relying on the guidelines to

evaluate information contained in instructions manuals [27], we made sure that the document followed the under mentioned specifications:

- The instructions described all the product characteristics in a step-by-step procedure;
- The manual included a quick start guide;
- It also includes a list of the functions;
- The manual included line numbers to help with cross references;
- Instructions were presented in the form of step-by-step procedures;
- The information was written in a consistent way;
- Sections were ordered by frequency of use;

After this, we asked five additional persons to analyze the final document and look for potential inconsistencies. We re-designed the first prototype based on these evaluations results. The Graphical User Interface created to enter the manual information through the web form was additionally evaluated to determine the usability of the software. The five users considered the platform to be friendly and easy to use.

5 Conclusion and future work

In this paper, we have presented "ManPro", a novel framework for the computer based creation of instruction manuals for the operation of technical systems in nuclear power facilities.

Our approach is language and platform independent, ensures accuracy of documentation content through predefined fields for data entry, preventing errors, and the information is accessible to multiple users. Additionally, the tool allows for multilingual content. Our approach has been proven to be efficient and user-friendly, as it guarantees that the information contained in the final user manual is searchable and understandable for the user. Therefore, the results presented in this paper establish the base for our further research in the field. We will refine UML diagrams in future work adding more extensive information to the instruction manuals and will assess the content design, organization and operating instructions through a comprehensive database-system for the evaluation of human factors and ergonomics.

Acknowledgments This work was supported by the VeNuS 2 Project - Approach to the Efficient Assessment of Safety and Usability of Computer-Based Control Systems - Number 1501282/2011, funded by national funds through the Federal Ministry of Economics and Technology, Germany, BMWi. It contributes to the research on reactor safety promoted by the Federal Ministry of Economics and Technology, Germany, BMWi.

References

1. Pyzdek, T., Keller, P.: Quality engineering handbook. Volume 60. CRC (2003)
2. ISO: 01.110: Technical product documentation, international organization standardization (July 2012) http://www.iso.org/iso/catalogue_ics_browse?ICS1=01&ICS2=110&.
3. ISO/IEC: Isoiec 26514-2008 systems and software engineering. Requirements for designers and developers of user documentation (July 2012) http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=43073
4. Carver, D.: Designing the user interface, strategies for effective human-computer interaction. Journal of the American Society for Information Science 39(1) (1988) 22–22
5. Parnas, D., Asmis, G., Madey, J.: Assessment of safety-critical software in nuclear power plants. Nuclear Safety 32(2) (1991) 189–198
6. Heymann M, Degani A, "Formal analysis and automatic generation of user interfaces: Approach, methodology, and an algorithm," Human Factors: The Journal of the Human Factors and Ergonomics Society, V. 49, No. 2. 2007, pp. 311-30.
7. Jacobson, I., Booch, G., Rumbaugh, J. (1999). The Unified Software Development Process. Addison-Wesley.
8. Pressman, R. S., 2005. Software Engineering – A practitioner's approach, 6th edition. Mc Graw Hill.
9. da Cruz AMR, Faria JP, "Automatic Generation of User Interface Models and Prototypes from Domain and Use Case Models."
10. Silva, A.R., Saraiva, J., Silva, R., Martins, C. (2007). XIS - UML profile for extreme modeling interactive systems. In Proceedings of the 4th International Workshop on Model-based Methodologies for Pervasive and Embedded Software (MOMPES 2007). IEEE, March.
11. Pastor, O., Molina, J. (2007). Model-driven Architecture in Practice – A software production environment based on Conceptual Modeling. Springer-Verlag.
12. Molina, P.J., Hernández, J. (2003). Just-UI: Using patterns as concepts for IU specification and code generation. In Perspectives on HCI Patterns: Concepts and Tools (CHI'2003 Workshop).
13. Jia, X., Steele, A., Liu, H., Qin, L., Jones, C. (2005). Using ZOOM approach to support MDD. In Proceedings of the 2005 International Conference on Software Engineering Research and Practice (SERP'05), Las Vegas, USA.
14. Courtois PJ, Parnas DL, "Documentation for safety critical software," Book Documentation for safety critical software, Editor, ed.^eds., IEEE, City, 1993, pp. 315-23.
15. Johnson, W., et al.: Dynamic (re) generation of software documentation. In: Proceedings of the Fourth Systems Reengineering Technology Workshop. (1994)
16. Riva, C., Yang, Y.: Generation of architectural documentation using xml. In: Reverse Engineering, 2002. Proceedings. Ninth Working Conference on, IEEE (2002) 161–169
17. Olaverri-Monreal, C., Bengler, K., Breisinger, M., Draxler, C.: Markup languages and menu structure transformation during the internationalisation process of driver information systems. Localisation Focus (2010) 4
18. Kazman, R., O'Brien, L., Verhoef, C.: Architecture reconstruction guidelines. (2001)
19. Nentwich, C., Emmerich, W., Finkelstein, A., Zisman, A.: Box: Browsing objects in xml. Software-Practice and Experience 30(15) (2000) 1661–1676
20. Zöllner, R., Patakui, K., Rosenthal, K.P.: Generation of the user documentation by means of software's specification in uml. In: Proceedings of the International Symposium Facteurs Humains et Conception des Systèmes de Travail. Optimiser les Performances de lEnterprise. Nizza. Comit AISS Recherche (2006)
21. Yu, A., Steele, R.: An overview of research on reverse engineering xml schemas into uml diagrams. In: Information Technology and Applications, 2005. ICITA 2005. Third International Conference on. Volume 2., IEEE (2005) 772–777

22. Jensen, M., Møller, T., Pedersen, T.: Converting xml dtds to uml diagrams for conceptual data integration. Data & Knowledge Engineering 44(3) (2003) 323–346
23. Zanden, B., Myers, B.: Automatic, look-and-feel independent dialog creation for graphical user interfaces. In: Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, ACM (1990) 27–34
24. Koo, S., Son, H., Seong, P.: A method of formal requirement analysis for npp i&c systems based on uml modeling with software cost reduction. Journal of Systems and Software 67(3) (2003) 213–224
25. Chase, N.: Using jdbc to extract data into xml (November 2012) http://www.ibm.com/developerworks/xml/tutorials/x-extract/.
26. XMLDOM: Xml dom the document object (November 2012) http://www.w3schools.com/dom/dom_document.asp. 57–66
27. Inaba K, Parsons SO, Smillie RJ, "Guidelines for developing instructions." CRC; 2004.