Ergonomic Assessment of the "ManPro" Software Application for the Computer Aided Generation of Documentation for Nuclear Facilities

Cristina Olaveri-Monreal*, Carsten Dlugosch, Klaus Bengler
Technische Universität München, Germany
Institute of Ergonomics
Department of Mechanical Engineering
{olaverri, dlugosch, bengler}@lfe.mw.tum.de

Abstract

Instruction manuals have a direct impact on the efficiency of processes within nuclear facilities directly affecting the productivity and safety of employees. Operation handling must therefore be specified in an accurate manner that avoids ambiguity and prevents errors. To support this process, a framework was implemented for the computer based creation of instruction manuals for the operation of technical systems in nuclear facilities ("ManPro"). The tool is based on a modeling specification in the Unified Modeling Language (UML), which defines the components of a system. This paper extends the "ManPro" approach examining the ergonomic aspects for its system components and interactions.

Keywords : Software Evaluation, Technical documentation, Nuclear Power Plants, Markup languages

1 Introduction

Recent unfortunate events related to safety systems within nuclear facilities have boosted worldwide concern about the disadvantages of using nuclear energy. Many control systems are obsolete and no longer comply with the current technological and safety standards. This also applies to the production of instruction manuals that have been, in some cases, developed over 40 years ago and rarely updated. These user manuals are directly responsible for the efficiency, productivity and safety within the facilities, and when left outdated could lead to dire consequences. For the safety of nuclear power plants operators it is crucial to make sure that the specifications that a plant has to meet, as well as its handling instructions, functionality and architecture is updated [1]. The "ManPro" framework for the Generation and Assessment of Documentation for

*Corresponding author

Nuclear Facilities has been recently developed to assuage the concern related to the safety systems of nuclear facilities. “ManPro” facilitates the generation of user manuals through a computer aided process that guides the technical writer through the different system components [2, 3]. 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 ability of such a new design concept as “ManPro” to meet the user needs as well as its ease of use was verified through the electronic reference system and evaluation tool “EKIDES”, that was developed by the institute of Ergonomics at the Technische Universität München as a modular framework to assist designers of technical systems to meet ergonomic requirements [4]. The electronic database provides ergonomic knowledge that has been compiled from industrial standards, scientific publications and research reports, as well as from laws and regulations regarding relevant topics concerning the study of interactions among humans and other elements of a system to optimize human well-being and overall system performance [5]. It allows for ergonomic assessment of workload, task or products through the selection of check lists that build the heuristics relevant to the tested product. The ergonomic tests can then be performed according to the selected requirements resulting in a test report generation. The rest of this paper is organized as follows. Section 2 revises related work in the area of the automatic generation of documentation. Section 3 describes the implementation process of the proposed software application, section 4 reports on the evaluation of the system. Finally, section 5 concludes the paper.

2 Related Literature

Markup languages support the representation of structured information. Therefore, they can be used to translate software engineering that is represented in a certain language into other markup languages. The authors in [6, 7] made use of this interoperability and introduced a software solution for a variable information structure display that was based on the combination of markup languages. The Extensible Markup Language (XML) was used for representing a document structure, which could subsequently and independently be transformed into new ones using style sheets in the Extensible style sheet Language (XSL). Basing their approach on markup languages, the authors in [8] introduced a portable, distributed approach that allowed browsing UML models on the internet using XML and related specifications, such as the Document Object Model (DOM), the XML Metadata Interchange (XMI) and the Vector Markup Language (VML).

Most approaches to generate software documentation use UML diagrams that include the system design information [12]. In a different context a further, XML-based approach that focused on the generation of architectural documentation for a software system from the implementation code was presented in [9]. 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. A related approach to an iterative reconstruction process that allows users to create a source code model in an database, and extract the reconstructed architecture through Structured Query Language (SQL) queries was proposed in [10]. An additional technology that addressed a related topic was introduced in [11]: The authors developed an authoring tool for automating the generation process of software architectures from program sources. In the same context of automatic information extraction from an existing system, in [13] it was shown that graphical input dialogs can be automatically created from text specification. Additionally, several model-based approaches that have been introduced in previous research [14, 15, 16, 17] allow to automatically generate user interfaces. Leaning on a combination of some technologies from these frameworks, we generate specific documentation for nuclear facilities semi-automatically through a computer-aided process. Making use of the interoperability characteristics of the markup languages, particularly of UML diagrams representing nuclear systems components, our approach transfers the required information from the software's functional specification described in the Unified Modeling Language (UML) into user documentation, analog to the process to generate software documentation. We display a correct user interfaces that map the different system components, allowing the user to enter the information that will later appear in the user manual.

3 ManPro Implementation Process

3.1 Requirements Analysis

The main goal of the "ManPro" framework is to achieve a platform that generates functionality instructions for a more flexible representation of documents ensuring that the information contained in the final user manual is searchable and understandable for the user, without ambiguity. Accuracy of document content must be ensured through predefined fields for data entry preventing errors and making the information accessible to multiple users.

3.2 System Design

The "ManPro" tool adheres to internationalization principles, allowing for multilingual content. 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. The tool implementation process is depicted in Fig. 1. To edit an instruction manual, the technical writer needs only access the appropriate Web forms to then create a document or add information or change contents for an existing document. The descriptive components of the instruction manuals are then stored in XML format for the further compilation of the final document.

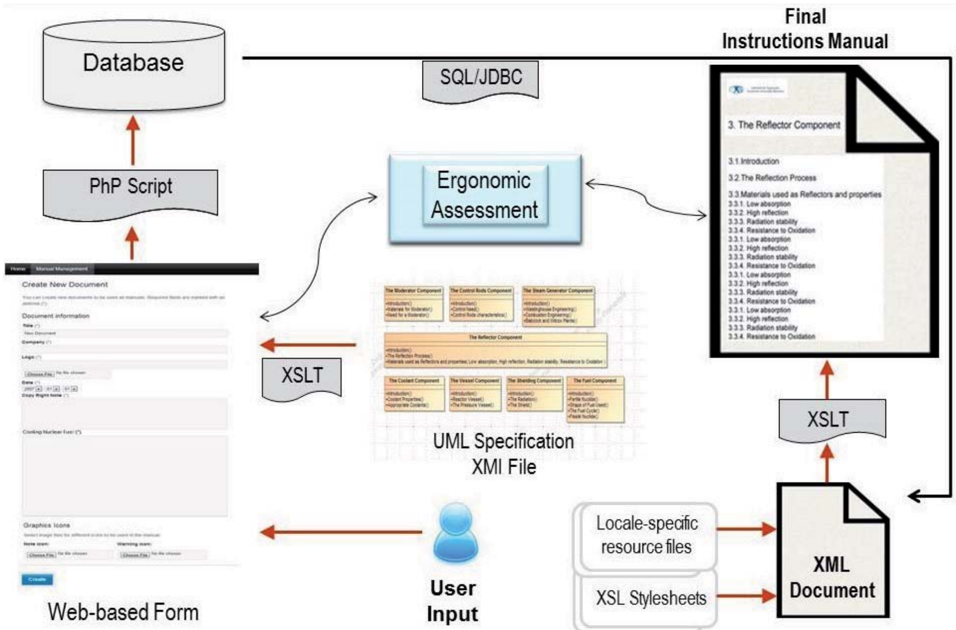

Figure 1: “ManPro” process to semi-automatically generate instruction manuals for nuclear facilities.

3.3 System Components

“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 a human-readable format such as PDF achieved through the XSL Transformation language XSLT;

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. The Web-based form 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 further compilation of the final document.

3.3.1 Web Form Generation from UML

The Unified Modeling Language (UML) is a standardized modeling language. It contains a system for notating graphics and, thus, it enables the creation of visual models for object-oriented software applications. For these reasons, it has become the industry standard for modeling software systems [18]. UML provides a standardized way to write a system’s blueprint, for example classes written in a specific programming language, database schemes, and reusable software components. A component diagram depicts how a software system is split up into components, and also shows the dependencies among these components [19]. Therefore it provides a perfect framework to develop the specifications functionality and architecture that a nuclear plant must meet.
To generate the user manuals from the different system components, we used in our framework a source class diagram in UML. Each class represented a chapter in the document; the methods contained in the classes represented the chapter sections. Fig. 2 shows a sample UML to illustrate our method based on the Nuclear Power Reactors Components specified in [20]. An XMI file was exported from the diagram. Since the XMI input file is XML-based, it was accessed through a XSLT file to generate the input fields of the Web form and be able to display the object-oriented design in UML in a web browser.

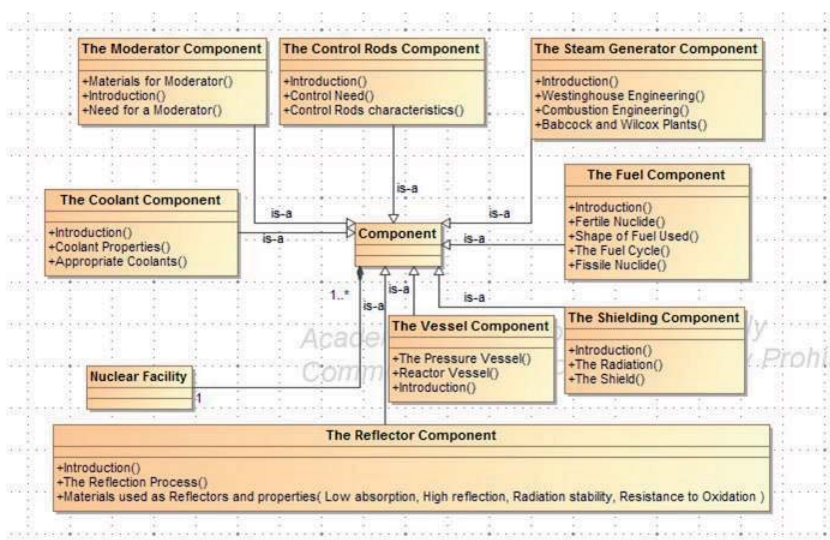

Figure 2: UML Sample Class Diagram.

The form fields reflected the class and interface UML elements as well as the relationships between objects (i.e. inheritance). The XSLT style sheet contained the patterns to match the relevant elements within the XMI document and the transformations rules to apply when a match was found. To build the Web form for data entry, we used the HyperText Markup Language (HTML). 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.

3.3.2 Database

A database is a collection of data that can be shared by several users. In "ManPro", access to the database was restricted to only authorized users requesting a login for the password protected web server. In the relational database model, data is organized in the form of tables that contain information which can be accessed through the Structured Query Language (SQL). To develop the relational database platform, we used the XAMPP PHP-Apache-MySQL solution. The "document" table was connected to the rest of the tables containing document parts. This architecture allowed for reuse of such data for multiple documents. Many-to-many relationships were saved in separate tables, which referred to primary keys with foreign keys of further tables to easily join information for a later data extraction based on queries.

3.3.3 XML Extraction from the Database

The Java application programming interface (API) and data access technology called Java Database Connectivity (JDBC) provides methods for querying and updating data in relational databases [21]. To be able to access database information in an XML format, we built our approach upon the JDBC data access technology. Through a mapping process, the information desired for retrieval was selected leaning on an intermediate xml mapping file used by java code to acknowledge the database structure [22]:

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, the overall structure of the new XML document was specified, describing the tree elements and their attributes, namely the root and the rest of elements that represented the database rows.

3. Additionally, the structure information was completed with the names and contents of data elements, creating new elements and attributes.

An XSL style sheet 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 intended for manipulation. 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 [23].
To load the locale-specific resource from the appropriate resource file for the current user locale, we proceeded according to the following steps: Text elements for each language were stored in external XML documents or dictionaries. The source XML documents as well as the dictionaries contained an ID attribute for defining the node, or the text element that had to be inserted into the node, respectively. Finally, an XSLT file was used to compare the ID attributes of both documents and insert the corresponding translation into the target XML document through variables that stored the whole dictionary and the ID attribute of the matched element.

3.3.4 PDF Generation

The Extensible Style Sheet Language Formatting Objects (XSL-FO), enables formatting XML data for output to different media. The “ManPro” framework contained several XSL style sheets for the different representation of the XML content. PNG images were generated from the style sheets to enable a preview of the format in which the PDF would be created. We proceeded according to the following method: an image of the first page and a chapter page were saved on the server. Their URIs were stored in the database as XSL style sheets metadata. As a consequence, every time the user selected a different template from the drop-down menu option to generate a PDF file, each style sheet and its correspondent image URI was read and loaded using JavaScript. Using XSLT, the final XML document transformation into a PDF was performed. 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.

3.4 Graphical User Interface

For consistency reasons, all the pages in the application were based on the same structure and contained a Header element with the “ManPro” application title and logos from the Technische Universität München, Institute of Ergonomics and Venus 2 project. The Navigation menu contained the menu items for navigation, the side bar, instant help on the current page and the page content, elements such as input fields and other interaction buttons. To create a new manual the user had to select the ”New Manual” from the ”Manual Management” option. To edit an existing manual, the ”Edit Manual” option from the ”Manual Management” menu needed to be selected. Finally, to generate a PDF file from an exiting manual, the user had to select the ”Generate PDF” option. The ”Home” button allowed navigation back to the main page. To give the reader a concrete idea of page layout, Fig. 3 shows the menu options to create a PDF file from a selected manual. Fig. 4 illustrates the navigation process within the manual management menu option from the “ManPro” online form.

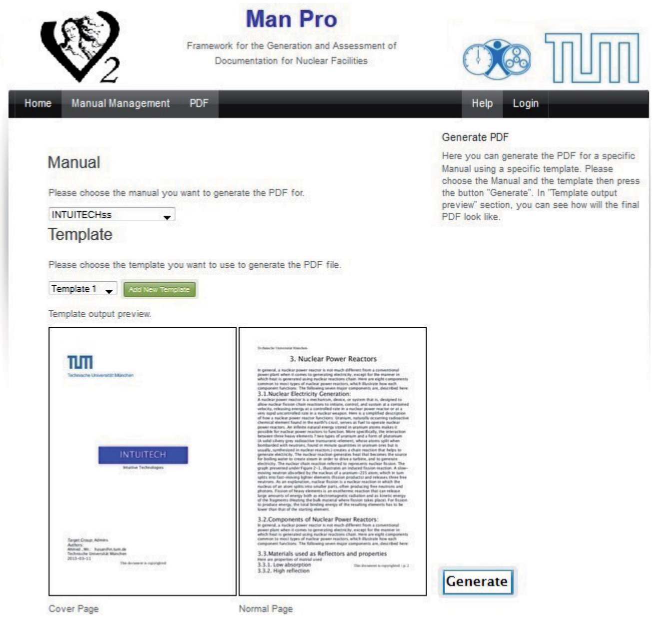

Figure 3: Screenshot of the "ManPro" online tool to create a PDF file from a selected manual.

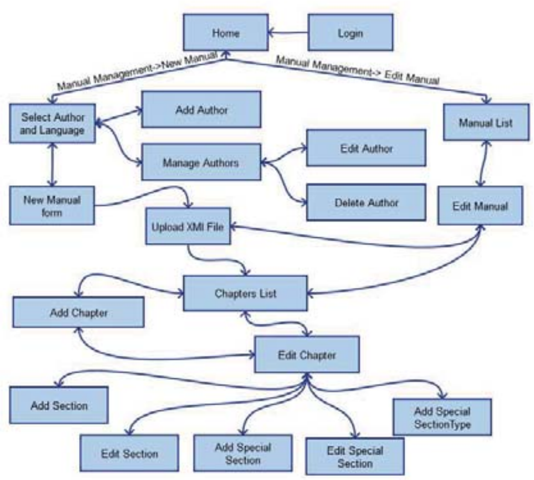

Figure 4: Navigation within the "manual management" menu option of the "ManPro" Tool.

4 ManPro Evaluation Process

4.1 Formative evaluation

During the continually-evolving development of the application, we performed several formative evaluations to detect and fix any potential operating problems. The first inspection-based evaluation during the development phase was carried out by two independent usability experts. They performed typical processing steps with the program (the tasks were allocated by the developer). The evaluation was carried out on the basis of predefined usability principles (heuristics). Additionally, 2 further experts performed a further evaluation selecting 142 items from the computer-aided ergonomic testing and evaluation module of the EKIDES tool [24] that were relevant to the “ManPro” framework. A section of the selected heuristics corresponding to the general rules for software design is shown in Fig. 5. For example some of the relevant heuristics corresponding to data input were:

- Prevention of unintended actions (unintended data loss);
- Delete authorization (deletion must be confirmed by the user);
- Immediate feedback (time to give feedback to the user must be immediate, otherwise the processing time must be shown);
- Clear indication of input field formats (for example for dates).

After selecting the relevant components for the testing of the application, the processing test report section of EKIDES was run and prompted a user interface that showed: a) an icon for the effect of a non-fulfillment of a certain system heuristic on health (it did not apply to the “ManPro” tool) safety, performance, reliability and comfort; b) its description and c) a question regarding the completion of the requirement. An input field from 1 (minor) to 3 (high) to enter the level of severity of the problem completed the system. Usability experts analyzed the “ManPro” tool according to these criteria. The most significant results are commented on the results section.

4.2 Summative evaluation

To verify the almost finalized “ManPro” system according to the criteria specified in the requirements phase a summative evaluation was performed by 14 subjects. They analyzed the criteria described in Table 1 classifying the ManPro Tool on a scale from 0 (very bad) to 10 (very good). Results are discussed in the next section.

Visual Design	Error Feedback	Functionality / Efficiency	Feedback	Navigation and Content	Learnability
Colors and fonts match	Warning messages for actions that can not be reversed	System works according to user expectations	Useful feedback	User guidance and input instructions	No colloquial language, abbreviations or acronyms contained
Color discrimination	Spell grammar check tools	System features work properly	Next steps feedback	Higher level menus buttons available	Standard icons available
Indication of active window	Auto correct tools	Tasks can be performed in a reasonable time	Feedback by processing time <2 secs	Easy and intuitive navigation	Intuitive system usage
Information groups definition	Error messages description for problem solving	Sequential menu selection available	Entries control feedback	Information easy to find	Terminology consistency

Table 1: Defined subjective evaluation criteria for the "ManPro" assessment.

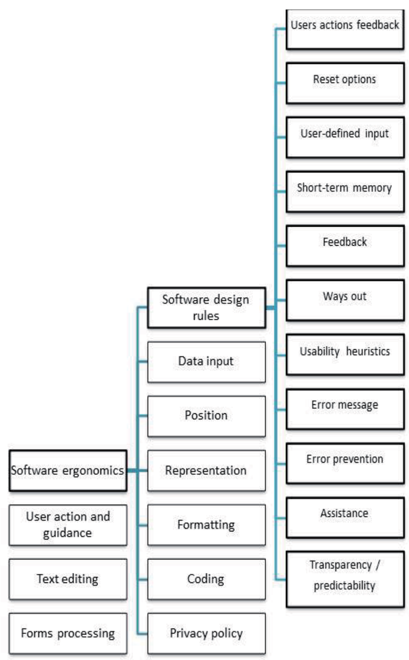

Figure 5: Section of the relevant components for the "ManPro" evaluation process.

4.3 Evaluation Results

In this section we summarize the main results from the performed formative evaluation. We indicate the problems related to usability and if they were corrected.

- Optical appearance and color choice: the combination of grey, black, blue and white makes a good impression. The color contrast is also good.
- Font size: the font is easy to read. Highlighting important information through bold printing facilitates orientation.

- Layout (size and arrangement of the elements): the layout is largely intelligible and clear. The texts that describe each menu are located top right on each page for each function, resulting in a beautiful design impression. The symmetrical arrangement to the title of the page that did not exist in previous versions was corrected. Additionally, explanatory texts were shown to the user after placing the mouse over the buttons.
- Input modalities: the program is easy to use with the mouse and the mouse areas are reasonably large. Functions are not yet accessible using the keyboard, shortcuts do not exist.
- Functionality: the functions within the main views "Create New Manual" and "Edit Manual" are clearly structured. The individual fields are also clearly labeled and the input functions react as expected. Input is enforced for fields marked with asterisks.
- Fault tolerance: feedback was given to the user about actions performed and right or wrong interactions.
- Help function: the help function in the menu tab "Home" includes a description about how to use the program and where to find the functions.

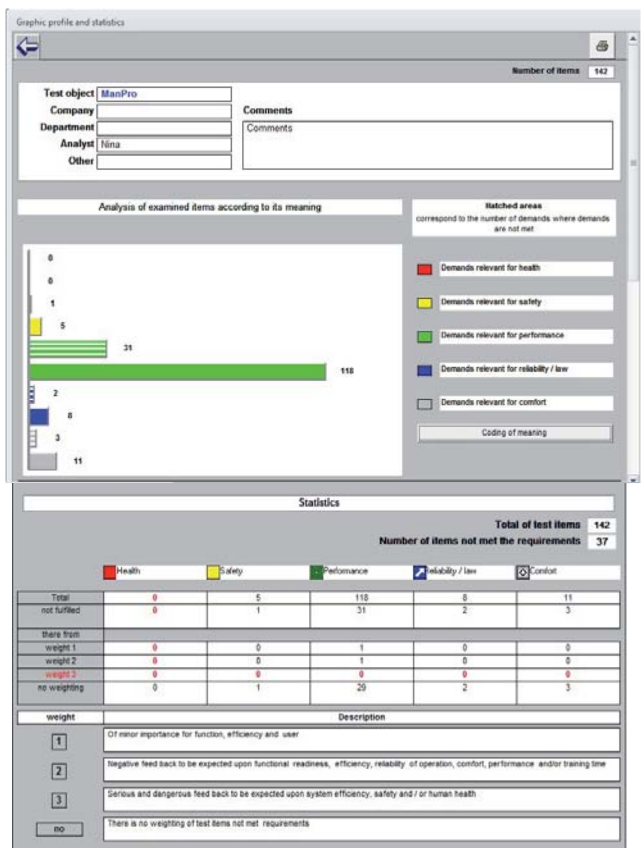

Figure 6: "ManPro" graphic evaluation analysis.

The results regarding the completion of the requirements on safety, performance, reliability and comfort and the severity level of the problem from 1 to 3 are illustrated in Fig. 6. From the 142 items that applied to "ManPro", 37 did not meet the requirements. None represented a serious problem that prevented usability of the system (scored with 3) and only 2 items were scored as a minor (1) or medium (2) severity problem for the system performance, meaning that users quickly adapted to the issues and fixing them should be given low to medium priority.
Of the 5 items that affected safety of the systems, only 1, protection or encryption of stored data was not fulfilled. This issue was corrected.
From the 8 items that affected the reliability, the following two were not accomplished by "ManPro": 1) confirmation of actions by the user before occurring and 2) some unintended actions could lead to lose of data. Both were corrected.
From the 11 items that affected the comfort of using the system, 3 were not fulfilled by the system. Two of them were related to the help function and one concerned the position of the cursor on the first input field. All three were corrected.
Concerning the performance of the system, 118 heuristics were relevant for the tool and of them 31 were not met. They were related to feedback from the system by data change and layout problems such as paging and scrolling (not enough space to visualize information). All of them were corrected.
Results regarding the "ManPro" classification on a scale from 0 to 10 according to the criteria in Table 1, are summarized in Fig. 7. The graphic displays the score given to the application by a specific subject (green or external line) compared with the mean value obtained from all the 14 subjects (blue or internal line). The mean values for the system's functionality (0.61), feedback (0.71), navigation and content (0.64), learnability (0.68) and visual design (0.71) show that the users were satisfied with the tool.

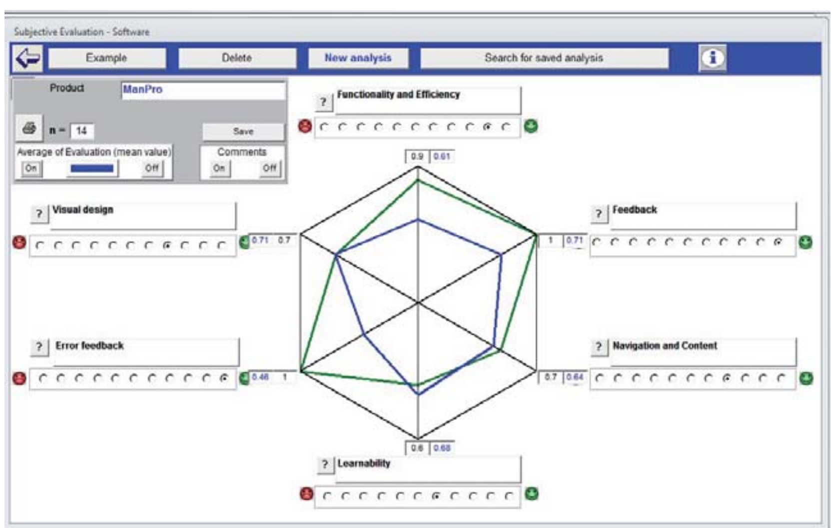

Figure 7: "ManPro" subjective evaluation analysis.

Only the mean value of the error feedback parameter was classified as insufficient by all the subjects with a score of 0.46 .

5 Concluding Remarks

This paper extended the content of earlier research in the field of instructions manuals for nuclear power plants [2, 3, 12] and examined the “ManPro” approach for the semiautomatic generation of instructions manuals from an ergonomic point of view. It showed that a complete software analysis can be achieved by making use of the EKIDES tool. The main “ManPro” qualities can be summarized as follows: “ManPro” is a novel framework for the computer based creation of instruction manuals for the operation of technical systems in nuclear power facilities. The approach is language and platform independent and ensures accuracy of documentation content through predefined fields for data entry, preventing errors. Additionally the information is accessible to multiple users. The tool also allows for multilingual content. “ManPro” 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. The performed “ManPro” ergonomic assessment allowed the gathering of knowledge to learn and improve future functionality and interaction design and implementation.

6 Acknowledgements

This work was supported by the VeNuS 2 Project - Vorgehen zum effizienten Nachweis der Benutzbarkeit und Sicherheit rechnergesttzer Leittechniksysteme (Approach to the Efficient Assessment of Safety and Usability of Computer-Based Control Systems), Number 1501282/2011, funded by national funds through the Bundesministerium für Wirtschafts und Technologie, BMWi. It contributes to the research on reactor safety promoted by the Federal Ministry of Economics and Technology, Germany.

References

[1] Pyzdek, T., Keller, P.: Quality engineering handbook. Volume 60. CRC (2003)

[2] Olaverri-Monreal, C., Dlugosch, C., Bengler, K.: Manpro: Framework for the generation and assessment of documentation for nuclear facilities. In: Advances in Information Systems and Technologies. Springer (2013) 849–859

[3] Nelke, T., Sachse, K., Thring, M., Olaverri Monreal, C., Dlugosch, C., Bengler, K.: An approach to the efficient assessment of safety and usability of computer-based control systems (venus 2). In: 7th meeting in the series of Enlarged Halden Program Group meetings, Storefjell, Gol, Norway 2013. (2013)

[4] Jastrzebska-Fraczek, I., Bubb, H.: Software design and evaluation by ergonomics knowledge and intelligent design system (ekides). PsychNology Journal **1**(4) (2003) 378–390

[5] IEA: International ergonomics association: What is ergonomics (April 2013) http://www.iea.cc/01$_{w}hat/What\%20is\%20Ergonomics.html$, .

[6] 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

[7] Olaverri-Monreal, C., Draxler, C., Bengler, K.J.: Variable menus for the local adaptation of graphical user interfaces. In: Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on, IEEE (2011) 1–6

[8] Nentwich, C., Emmerich, W., Finkelstein, A., Zisman, A.: Box: Browsing objects in xml. Software-Practice and Experience **30**(15) (2000) 1661–1676

[9] Riva, C., Yang, Y.: Generation of architectural documentation using xml. In: Reverse Engineering, 2002. Proceedings. Ninth Working Conference on, IEEE (2002) 161–169

[10] Kazman, R., O'Brien, L., Verhoef, C.: Architecture reconstruction guidelines. (2001)

[11] Johnson, W., et al.: Dynamic (re) generation of software documentation. In: Proceedings of the Fourth Systems Reengineering Technology Workshop. (1994) 57–66

[12] 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)

[13] 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

[14] da Cruz, A.M.R., Faria, J.P.: Automatic generation of user interface models and prototypes from domain and use case models. In: ICSOFT (1). (2009) 169–176

[15] da Silva, A., Saraiva, J., Ferreira, D., Silva, R., Videira, C.: Integration of re and mde paradigms: the projectit approach and tools. Software, IET **1**(6) (2007) 294–314

[16] Pastor, O., Molina, J.C.: Model-driven architecture in practice: a software production environment based on conceptual modeling. Springer (2007)

[17] Molina, P.J., Meliá, S., Pastor, O.: Just-ui: A user interface specification model. In: Computer-Aided Design of User Interfaces III. Springer (2002) 63–74

[18] UML: Uml forum (July 2012) http://www.uml-forum.com/FAQ.htm.

[19] UKEssays: Uk essays.com generation of uml component diagrams based on user specification (November 2012) http://www.ukessays.com/essays/computer-science/generation-of-uml-component-diagrams-based.php.

[20] Mir, F.A.: Intuitech nuclear considerations (July 2013) http://intuitech.biz/nuclear-power-reactors-components/.

[21] JDBC: Trail jdbc tm database access (November 2012) http://docs.oracle.com/javase/tutorial/jdbc/index.html.

[22] Chase, N.: Using jdbc to extract data into xml (November 2012) http://www.ibm.com/developerworks/xml/tutorials/x-extract/.

[23] XMLDOM: Xml dom the document object (November 2012) http://www.w3schools.com/dom/dom$_{d}ocument.asp$.

[24] Jastrzebska-Fraczek, I., Schmidtke, H., Bubb, H., Karwowski, W.: Ergonomics knowledge and intelligent design system (ekides)-software tool for design, assessment and ergonomics teaching. International Encyclopedia of Ergonomics and Human Factors **2** (2006) 1613–1625