Evaluation of UML Modeling Tools for Clinical Pathways used in Health Information Systems

Christina Pahl[1,2], Eko Supriyanto[2,3,4], Vesselin Detschew[5]
Email: christina.pahl@tu-ilmenau.de, eko@utm.my, vesselin.detschew@tu-ilmenau.de
[1] Department of Biomechatronics, Ilmenau University of Technology, Ilmenau, Germany
[2] IJN-UTM Cardiovascular Engineering Centre, Institute of Human Centred Engineering, Universiti Teknologi Malaysia, Skudai, Malaysia
[3] Advanced Diagnostics and Progressive Health Care Research Group, Health and Wellness Research Alliance, Universiti Teknologi Malaysia, Skudai, Malaysia
[4] Department of Clinical Sciences, Faculty of Biosciences and Medical Engineering, Universiti Teknologi Malaysia, Skudai, Malaysia
[5] Institute of Biomedical Engineering and Informatics Ilmenau University of Technology
Ilmenau, Germany

Abstract — **For the development of clinical care pathways for the Brazilian hospital information system Pajé, a modeling tool is required. For the selection of this tool, some modeling standards were reviewed, applied and subsequently evaluated. Four modeling tools, zOOml, gmodeler, dia and UMLet were tested and validated for the process of modeling UML diagrams based on predefined requirements. Results show that all chosen modeling tools can be used for the representation of clinical pathways. However, only two of them being Dia and UMLet full filled the requirements of this study. It can be concluded that these tools can be used for E-health applications and cross platform modeling projects. Our results are of value for health information system developers and further e-health related projects.**

Keywords – clinical pathway; e-health; health information system; modeling; PACS; Pajé; UML.

I. INTRODUCTION

Currently, Brazil aims at following the state-of-the-art in E-health and promotes the digitalization of its health care system on all governmental levels. Therefore, approaches were made to integrate clinical pathways within a Health Information System (HIS) among existing public hospitals. This work involves various public institutions and universities working on the same concept. The main focus of this project lies on the documentation, administration, support of e-health research as well as the application of a Picture Achieving and Communication System (PACS) for clinical image data [1]. The system is called Pajé and is under development. There were two conflicting requirements predefined, which are (1) the architecture of the software has to be general so that institutions can use this product in larger scales like general hospitals as well as smaller health units and (2) the software should be specialized so that components can respond to specific needs of single health units.

A. Relation to "CloudMedWF"

The project "CloudMedWF" aims at developing a system architecture for the implementation of clinical pathways. Therefore, single architecture layers need to be clear and well documented, which is not the case among most of the existing clinical pathways [2], [3]. Furthermore, this project aims at the implementation of clinical pathways for the processing of medical images like ultrasound image data based on "Cloud Computing" [4], [5], [6]. For this purpose, server services are used to access resources via the World Wide Web. The objective is to find a suitable UML modeling tool.

II. MATERIALS AND METHODS

The solution to this problem is implemented in several phases, which are addressed in this work. This study is based on consultation with clinical professionals. The evaluation was conducted using Linux architecture.

A. Concept

Based on the achievements of related works [7], [8], suitable modeling tools have been reviewed [9]. First, a restricted selection of modeling languages has been made. Then, the decision on one modeling language was done based on required criteria [10].

B. Comparison of UML, BPMN and BPEL

In Table I, a comparison of the three modeling languages Unified Modeling Language (UML) and Business Process Model and Notation (BPMN) and (Business Process Execution Language) BPEL were made. BPMN addresses primarily business-oriented users such as business analysts and system architects, and was designed to support the complete development process for business processes, starting from the acquisition over the modeling and up to the implementation of clinical pathways. UML, however, helps developers to specify models of software systems to visualize, simulate and document pathways. It addresses primarily technically oriented users, such as system architects and software engineers and has been designed to support the complete development process for software systems. Clinical pathways in industry are often created with UML. BPEL is merely exemplary in the creation of models using BPMN.

This study was sponsored by UTM VOT Nr. 03G12.

TABLE I. COMPARISON OF MODELING TOOLS; UML, BPMN AND BPEL

	UML	BPMN	BPEL
Notation	Graphic	Graphic	Textually
Objective	Specification, modeling and Documentation of work processes	Modeling and documentation of workflows and business processes	Description and execution of business processes
Target Group	Computer scientists and system developers	Process analysts and business analysts	Low level computing professionals
Attributes	Comparable with a toolbox	Technically detailed models	Executable model, allows automation of business processes
Scope	Figure workflows	Subsequent realization	Interaction based on "Work lists"
Focus	Graphical representation of business processes, Picture and development of processes	Graphical representation of business processes mapping and development of processes among human experts	Execution of mapped models
Approach	Object-Oriented Approach	Process-centered approach	-
Support of Patterns	Direct support	No support	No support

C. Requirements

Due to the research character of this project, it was required that the tool has to be open source. The open-source property implies the freedom to modification and further distribution of source code [11]. Such software may be used for any purpose, and be modified and distributed in its original form.

The idea was born in the UNIX world, where it was common to allocate source code. The concept, however, is based on General Public License (GPL), which is a license of the Free Software Foundation (FSF) [12], [13]. It ensures that the software is free for all users. Another major requirement is that the tool shall be web-based [14]. This is to ensure that the tool can be used within the concept "cloud computing" during all stages of development. The following requirements were defined for this study:

- ∞ Open Source
- ∞ Web-based
- ∞ Support of UML and its diagrams
- ∞ Import/Export of other formats
- ∞ Feasibility of models with one engine
- ∞ Model-driven approach
- ∞ Processes should be represented in details
- ∞ The modeling should be easy
- ∞ Good ergonomics
- ∞ Clear navigation
- ∞ Graphical Modeling

Web-based tools have the advantage of not requiring software installation, being cross platform interoperable and collaborative. However, this criteria depends on the internet connection. The support of UML and its diagrams is essential. Exchanging other formats supports interoperability. Feasibility of models with one engine and model driven approach contribute towards performance. Possibility to obtain the process in details, easy modeling, good ergonomics, clear navigation and graphical modeling are major usability criteria. Modeling tools were selected based on the requirements defined above [15].

III. APPLICATION OF WEB TOOLS

Most of the tools available are complex and offer many redundant options to generate models efficiently. This results in low productivity. The focus of this work was therefore on lightweight tools. Due to their nature, web based tools do not require any installation. The internet represents the basis for their execution. The advantages of this method lie in the saving of data storage space of the local institution [16]. Moreover, this platform independence does not only allow the usage of any PC equipped with a browser but also any personal digital assistant and smart phone. Another advantage of the Web as a communication medium is the support of group work over long distances. Thus, projects can be realized, in which different instances can participate and thus multinational contributions to a single model is possible in real time. The tools listed below were tested in their chronological arrangement listed here.

A. ZOOml

This tool offer a website with free services. It was published in 2009 and has been improved online since then. The tool was programmed in C ++. In order to use it, registration is required. In Fig. 1, a close-up view of zOOml can be obtained, whereas Fig. 2 shows a detailed view of the work surface. It can only create UML class diagrams. Its surface is clearly structured and easy to handle. Furthermore, diagrams are created in machine-readable XML code but can be converted. Because this format is platform independent, it can be used for the interoperable exchange of data between different computer systems.

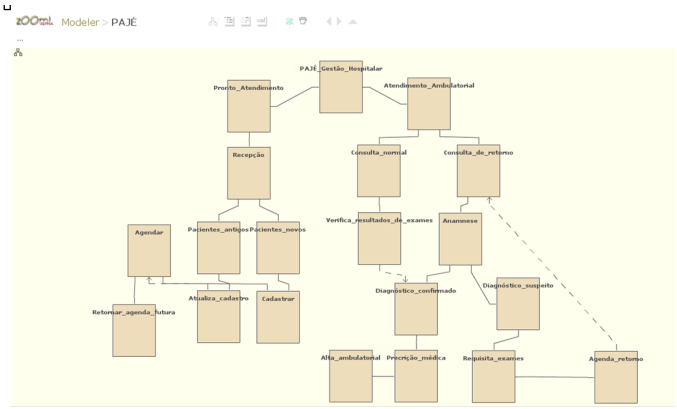

Figure 1. zOOml closeup view.

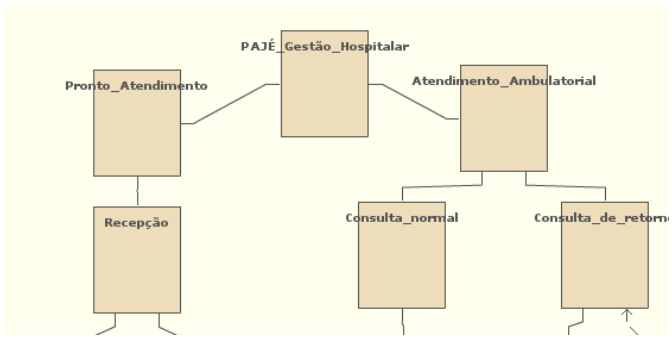

Figure 2. ZOOml detailed view.

The operation language is English. The services are free of charge and do only require internet access. Thus, the hardware requirements are reduced drastically. The tool is platform independent, and can be applied with any operating system, which avoids expensive licenses. Stored diagrams are saved chronologically. The user interface is plain and contains countable elements. The appropriateness of the placement of these controls cannot be objectively assessed, since it generally depends on the operating patterns of the user. The modeling process offers the option of documentation. There are illustrations provided on the website as well as easy-to-understand instructions, which are available in English. The navigation on the modeling interface is intuitive, which leads to the reduction of modeling training period. The nesting of elements depends on the frequency of use. However, the movement of elements on the working surface is affected by an interrupting time delay. This degrades the ergonomics considerably during the modeling process.

Moreover, relationships automatically follow changes of object positions. This dynamic attitude is beneficial in modeling large graphs. However, the size of classes cannot be varied. Longer class names are therefore only visible in its full length, when effected by double-clicking a highlight. For the designation of classes, not all special characters are accepted. “Space” is not considered as a valid character for naming classes. An alternative to that is the use of underscores. When using multiple objects, overview decreases rapidly. The work surface cannot be modified in size. The menu bar has a prominent symbol for storing the current diagram. This option was helpful during the creation of models, since there were occasional complications with the existing hardware. However, problems occurred during the printing of models created. The difficulty was that created charts did not adapt to DIN A4.

B. Gmodeler

The web based tool, gmodeler (see Fig. 3) offers another UML diagram next to the class diagram, which is a use case diagram. The first version was launched in 2003.

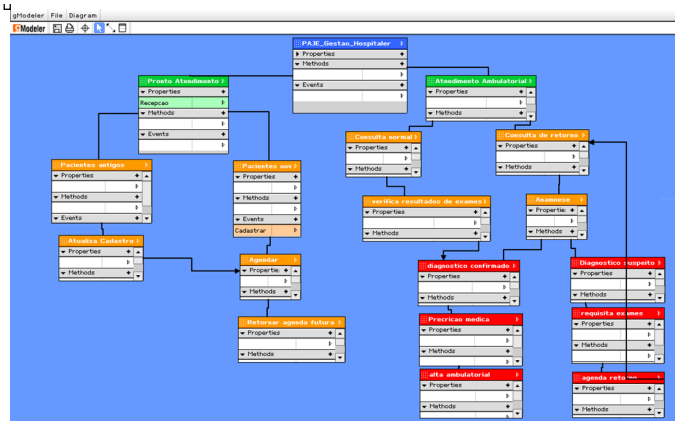

Figure 3. Gmodeler close-up view.

It exports charts created not only in an XML, but also in HTML format. It was designed by Grant Skinner. It was written in the programming language JavaScript and the operation language is English. This tool implicates no purchase costs. It is as well a platform-independent tool. As for its ergonomics, the complexity of the tools, a wide-ranging menu bar as well as a very good navigation on the work surface were experienced to be highly usable. The arrangement of its elements provides a good overview. The look and feel of object selection is designed to be intuitive. One can choose from a given amount of symbolic relations and chart objects. The menu offers seven tools to choose from. There is an opportunity to open a dialog to search for the latest information on gmodeler. One can save, print, and center current diagrams. Moreover, three further tools are available in the menu bar. The select-tool allows to select an item from the chart to manipulate its position. The link tool creates relations between objects and the element tool provides the option to choose different charts that can be drawn by double-clicking the desktop.

The creation of models starts on a centered work surface. A cross indicator serves as a starting point, which proved to be useful for to draft the larger models. The class diagrams can be divided in properties, methods and events.

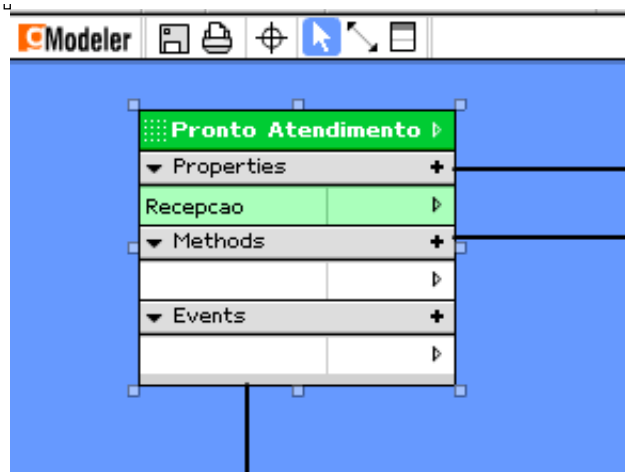

Figure 4. Gmodeler detailed view.

Furthermore, all objects can be navigated on the work surface without any delay in real time. As a further degree of design freedom, the size of these objects can be adjusted.

However, the designation of objects does not include special characters (see Fig. 4), so that letters, which do not exist in English language like Portuguese characteristic characters, cannot be used. Arrows can be adapted to the architecture of the chart, which promotes useful clarity. To further increase the plurality of individual classes and their properties, the assignment of colors may be used. This visual tool was experienced to be quite useful for the modeling of complex systems. Unfortunately, for the selection of relationships, there was no dotted arrow, so that such a relation could not be displayed. Furthermore, the ability to document the present work is enabled. As for the printing of developed diagrams the layout is automatically matched to the chosen paper size. To use this tool, it requires no registration. The diagram generated can be saved. For storage purposes, information about file name, creation date and date of the last modification are displayed. Moreover, data size are displayed.

C. *Dia*

This tool stands under the license of GPL and gets therefore regularly maintained. For the usage of this tool, an installation is required. It allows the creation of many types of charts.

This product supports class, component, object, package, use case, communication and many other chart types. The free of charge software runs on three operating systems being UNIX, Linux and Windows. The tool attracts by its engineer-friendly look and feel, which is characterized by a checkered background as well as a selection window in a calculator format. The usage of Dia reminds as already mentioned above about Microsoft Visio. Although it was not meant to be primarily an UML modeling tool, the most important UML diagrams can be created easily and quickly. The working surface is checkered and can be adjusted by the manipulation of the height of objects as it can be obtained from Fig. 5. However, the format of object-diagrams cannot be set manually. As for the usage of package-diagrams one can choose between large and small packages. In addition, object names are constantly high lightened. One advantage of the use of Dia is the opportunity to combine all provided chart types. However, the disadvantage of this tool lies in the lack of the ability of code generation. This includes the export and import from XML formats, which is not possible with Dia.

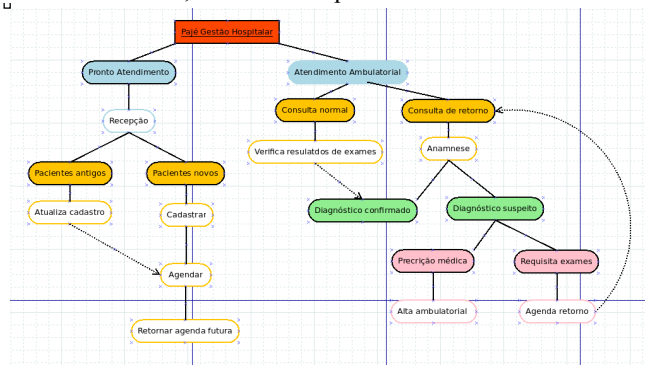

Figure 5. Dia close-up view.

All in all, this tool shows a high degree of usability based on a two-layered work surface. Moreover, special characters often used in Portuguese language were accepted without problems.

D. *UMLet*

UMLet has been download without any problems using a Linux based operation system. Using the terminal option one can automatically search for the latest UMLet version and download it. It provides a clear overview on its work surface and short training time.

Criteria enclosing platform independence and state-of-the-art features were the main factors for the choice of this tool. Its latest version was released in 2010. The online list of available updates, indicates that UMLet is regularly maintained. It stands under GPL and therefore its source code is available free of charge. UMLet was written in the computer language Java. In 2002, UMLet was made public. The tool provides a wide range of UML diagrams, which include class, activity, use case, sequence, and further types of charts [17], [18].

The user interface is divided into three sub-windows. This includes the work surface, a window for diagram components and a further window in which one enters the properties. The modeling begins by selecting symbolically represented objects with a double click from the palette. Selected objects appear subsequently in the upper left corner of the desktop. One can easily change the chart type. Already established relationships between objects retain even when changing the present position of objects. During the process of model creation, a UXF file is generated. This format provides a very simple form of the respective XML code, which enables textual sharing of self-created UML modeling, which is a strong advantage of this tool. However, it is not the intention of the UMLet working group to develop an environment for executable models or code generation. Fig. 6 shows a section of the care pathway that has been created with UMLet. Red circles indicate design issues being the attachment of connections and overwriting of circles. Relationships can be made visible at different angles. Those relationships, which are created between the objects do not lie edge to edge of the objects, but on an invisible quadrangle, which leads inconsistent connections.

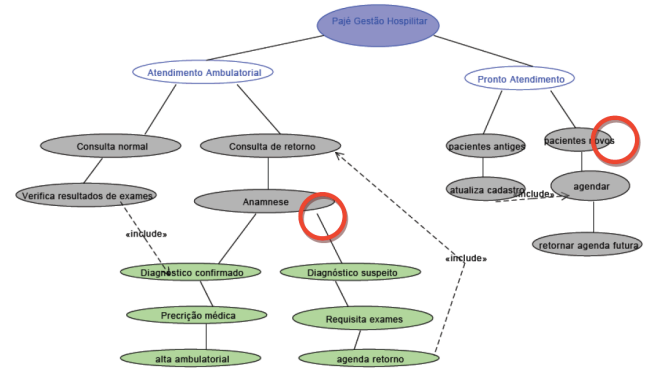

Figure 6. UMLet close-up view.

During the naming of objects in development mode, the borders of the objects enclose the position of names. However, after naming long names appear outside the object borders as it can be obtained in Fig. 6 in the red mark on the right side of the model. Another optional but not necessary feature of UMLet is to send the current model directly from the work surface by email. This enables the facilitated sharing of the development state of the model.

IV. Results

Table II shows main criteria for assessing the modeling process. Binary decision indicates the full filling of specific attributes. "Yes" indicates that the respective tool meets the criterion and "No" indicates that the requirement was not met.

One can clearly obtain the trend that Dia and UMLet fulfill most requirements. All tools are web-based and enable the representation of clinical pathways. Moreover, all products were found to be lightweight tools because their equipment was limited to essential functions. In contradiction to that, installation requiring tools are more powerful compared to the web-based. These usually offer a larger selection of UML diagrams, so that a higher level of detail representation can be achieved during the modeling process.

From Table II, it can be obtained that the UML diagram variability not satisfying in the case of zOOml as well as gmodeler. Moreover, the depth of details also poor when using the latter. As for the navigation on the work surface, gmodeler was significantly better than zOOml. In all cases, documentation of the modeling process was possible. Usability, however, was meeting the level of satisfactory in two cases only being Dia and UMLet.

In general, the zOOml online tool showed difficulties to navigate objects without delay. This was accompanied by less efficiency, and worse results. It was observed that zOOml is best for the simplified online representation of complex models. With the usage of gmodeler, a highly precise representation of simple models can be achieved. Furthermore, Dia is suitable for descriptive models. Finally, UMLet was observed to be highly suitable for the fast generation of models.

TABLE II. Evaluation of UML modeling tools

	zOOml	gmodeler	dia	UMLet
UML diagram variability	No	No	Yes	Yes
Depth of details	No	No	Yes	Yes
Navigation on work surface	No	Yes	Yes	Yes
documentation	Yes	Yes	Yes	Yes
Usability	No	No	Yes	Yes
Suitable for ...	Simplified representation of complex models online	Highly precise representation of simple models online	Descriptive models	Fast generation of models

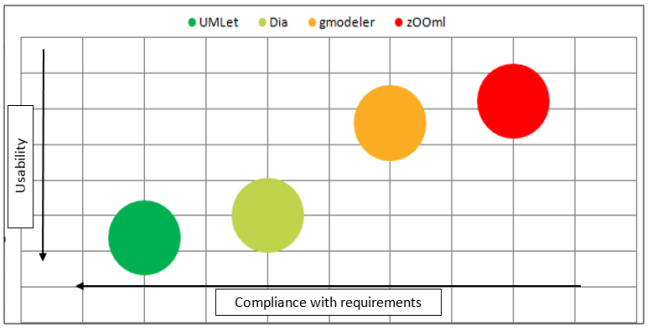

Figure 7. Evaluation of UML modeling tools.

The diagram displayed in Fig. 7 shows clearly the relation of selected tools based on their modeling properties. UMLet has proved to be the best tool. It offers numerous functions without falling into the category of a heavyweight tool. Diagrams created with UMLet, provide the required level of details. UMLet convinced by its handling characteristics. It offers various tools, which require some personal training in order to exploit the complexity of this tool.

V. Conclusions

This study provides an application based insight into web-based modeling tools and their requirements for the purpose of representing health data. The vague look and feel of zOOml consists of the UML class objects, which are suitable to illustrate a complex system, as well as a working process. However, for the modeling of complex charts, zOOml is not suitable. The real benefit of this tool lies in its simplicity. The handling of gmodeler is characterized by an intuitive operation and target-oriented work. The classes are shown in their full entity and properties. However, this simultaneously leads to deficits in the overview. This characteristics are useful for experienced user. Unfortunately, this tool is not updated regularly. Dia is a very handy tool to create accurate diagrams. One can use several types of charts in order to achieve a good level of details. Compared to the other tools a longer training time is required due to the variety of options. UMLet is in its complexity, a lightweight tool and shows strengths in its clarity and ease of its use. The elements on the work surface are not deeply nested and the menu is designed very intuitive. It's not just reduced to the bare essentials, but provides at an acceptable level a variety of extra options. Furthermore, it enables changes to UML templates or the source code. This result is useful for the future integration of clinical pathways within HIS of public hospitals. Therefore, UMLet represents the best candidate to become part of a system to support the documentation, administration and support of e-health research.

Acknowledgment

The authors would like to gratefully acknowledge the technical support of Universiade Federal do Paraná (UFPR). Moreover, we are thankful for the cooperation between UFPR and Ilmenau University of Technology (TUIL) and information supplement of TUIL. We gratefully acknowledge University of Technology Malaysia (UTM) for the financial support under

the VOT Nr. 03G12. Our appreciation also goes to anonymous reviewers, whose comments helped to improve this paper.

REFERENCES

[1] Cerami, Ethan G., et al. "cPath: open source software for collecting, storing, and querying biological pathways." BMC bioinformatics 7.1 (2006): 497.

[2] Philbin, James, Fred Prior, and Paul Nagy. "Will the next generation of PACS be sitting on a cloud?." Journal of digital imaging 24.2 (2011): 179-183.

[3] Davies, Jim, et al. "Model-driven engineering of information systems: 10 years and 1000 versions." Science of Computer Programming 89 (2014): 88-104.

[4] Daniyal, Ali, and Syed Sibte Raza Abidi. "Semantic Web-based modeling of Clinical Pathways using the UML Activity Diagrams and OWL-S." Knowledge representation for health-care. Data, processes and guidelines. Springer Berlin Heidelberg, 2010. 88-99.

[5] Pahl, C., et al. "Cervix Segmentation in Ultrasound Data Based on Numerical Technique.", (2013)

[6] Pahl, C., et al. "Performance Observation of Gabor Filter for Wall Thickness Measurement of Human Colon Based on Ultrasound Image.", (2013)

[7] Li, Weizi, et al. "Integrated clinical pathway management for medical quality improvement–based on a semiotically inspired systems architecture." European Journal of Information Systems 23.4 (2014): 400-417.

[8] Jalil, Noraini, et al. "Design and implementation of a telehealth platform for prenatal management in rural and remote areas." Biomedical Engineering and Sciences (IECBES), 2014 IEEE Conference on. IEEE, 2014.

[9] Mackay, Daniel, James Noble, and Robert Biddle. "A lightweight web-based case tool for UML class diagrams." Proceedings of the Fourth Australasian user interface conference on User interfaces 2003-Volume 18. Australian Computer Society, Inc., 2003.

[10] Kobayashi, Marina, et al. "Work coordination, workflow, and workarounds in a medical context." CHI'05 Extended Abstracts on Human Factors in Computing Systems. ACM, 2005.

[11] Wheeler, David A. "Why open source software/free software (OSS/FS, FLOSS, or FOSS)? Look at the numbers!." (2005): 514-520.

[12] Stallman, Richard. "Gnu general public license." Free Software Foundation, Inc., Tech. Rep (1991).

[13] License, GNU General Public. "Free Software Foundation." License used by the Free Software Foundation for the GNU Project. See http://www. fsf. org/copyleft/gpl. html (http://www. fsf. org/copyleft/gpl. html) (1991).

[14] Pino, Carmelo, and Roberto Di Salvo. "A Survey of Cloud Computing Architecture and Applications in Health." International Conference on Computer Science and Electronics Engineering. 2013.

[15] Pahl, Christina, et al. "Identification of Quality Parameters for an E-Health Platform in the Federal State of Thuringia in Germany." Journal of Soft Computing and Decision Support Systems 1.1 (2014): 17-23.

[16] Hewson, Claire, and Dianna Laurent. "Research design and tools for internet research." Sage Internet Research Methods 1 (2012).

[17] Auer, Martin, Thomas Tschurtschenthaler, and Stefan Biffl. "A flyweight uml modelling tool for software development in heterogeneous environments." Euromicro Conference, 2003. Proceedings. 29th. IEEE, 2003.

[18] Rouly, Jean Michel, Jonathan D. Orbeck, and Eugene Syriani. "Usability and Suitability Survey of Features in Visual Ides for Non-Programmers." Proceedings of the 5th Workshop on Evaluation and Usability of Programming Languages and Tools. ACM, 2014.

[19] Pahl, Christina, et al. "Role of OpenEHR as an open source solution for the regional modelling of patient data in obstetrics." Journal of biomedical informatics, 55, (2015): 174-187.