DE GRUYTER OPEN

Applied Computer Systems doi: 10.1515/acss-2016-0005

2016/19

Analysis of Sequence Diagram Layout in Advanced UML Modelling Tools

Oksana Nikiforova[1], Sergii Putintsev[2], Dace Ahilcenoka[3]

[1, 3] *Riga Technical University, Latvia,* [2] *National Aerospace University – "Kharkiv Aviation Institute", Ukraine*

***Abstract* – System modelling using Unified Modelling Language (UML) is the task that should be solved for software development. The more complex software becomes the higher requirements are stated to demonstrate the system to be developed, especially in its dynamic aspect, which in UML is offered by a sequence diagram. To solve this task, the main attention is devoted to the graphical presentation of the system, where diagram layout plays the central role in information perception. The UML sequence diagram due to its specific structure is selected for a deeper analysis on the elements' layout. The authors research represents the abilities of modern UML modelling tools to offer automatic layout of the UML sequence diagram and analyse them according to criteria required for the diagram perception.**

***Keywords* – UML sequence diagram, layout, system modelling, UML modelling tool.**

I. Introduction

System modelling is the interdisciplinary study of how to use models for system construction as well as for model conceptualisation. With system modelling it is easy for software developers to understand system behaviour, separated parts of the system and system structure. Also system modelling provides understanding of problems by separating problem domain to models, which are based on real-world ideas. This approach gives benefits in communication for people involved in the project or modelling enterprises or documentation and designing. With modelling it is possible to promote understanding of the requirements and get more fine design and more retainable systems. Modelling language has so-called graphical aspect that consists of perceptible location of model elements and intuitive language semantics. In this way, modellers have to cope with two main tasks while creating the diagram: representation of system functionality by diagram elements and design of an optimal positioning of diagram elements.

Graphical models provide a common base at different levels of system domain and are used at different stages of system abstraction for system developers. This responsibility has been given to such standardised modelling mean as Unified Modelling Language (UML) [UML] [1]. Nowadays, UML is widely used to represent system specification at different levels of system abstraction. UML defines a notation for a set of diagrams used for modelling of different aspects of the system (i.e., static and dynamic ones). The central part of static modelling in UML is class diagram, which defines the general structure of the system and serves as a basis for the development of software architecture [1]. From the layout point of view, the UML class diagram can be specified as a graph and the algorithm for graph layout can be applied to place elements of class diagram, taking into consideration specific of class relationships, especially aggregation and inheritance.

The central part of system dynamic modelling is the presentation of object interaction, where UML sequence diagram plays an important role and is used to present the system behaviour. As far as the structure of the UML sequence diagram is not pure graph, but a specific presentation of objects, their lifelines and messages sent between objects, the research on the automatic layout of the UML sequence diagram became more interesting and complicated. This paper is continuation of the research initiated by the authors in 2011 [3], [4], where authors focused on the layout abilities for the UML diagrams. Then authors worked on the development of their own algorithms for the layout of the UML class and sequence diagrams to be able to implement transformations of the two-hemisphere model in BrainTool [5]–[8]. And now the goal of the research is to analyse the abilities offered by UML modelling tools to layout the UML sequence diagram and to resolve the issue whether modern CASE tools satisfy all the criteria defined for the perception of the UML sequence diagram in its automatic layout.

The paper is structured as follows. Section 2 reviews related studies. Section 3 discusses the depicted review of UML tools for the specific layout of the sequence diagram and provides a basis for selection of UML tools for further analysis. Section 4 covers criteria for the "good" layout of the sequence diagram. Section 5 depicts a deep analysis of how selected tools follow the criteria and discusses the way methods could be used to improve the layout of the sequence diagram. Conclusions and future research analysis are presented in Section 6.

II. Related Studies

Nowadays there is a considerable set of UML tool vendors that claim that their tools support the possibility to provide the best approach for UML modelling [9]. However, diagram layout is not an easily executable task. Manual layout of diagrams is time-consuming, especially layout of large diagrams.

One of the questions that should be raised is how good tools support the possibility to manage "good" diagram layout.

The focus of the paper is turned to the ability of UML editors to offer the layout of the UML sequence diagram with the correspondence to the criteria stated for the diagram layout.

©2016 Oksana Nikiforova, Sergii Putintsev, Dace Ahilcenoka. This is an open access article licensed under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), in the manner agreed with De Gruyter Open.

Similar research was conducted by Wan Husin in [10], where the author emphasised high importance of evaluation. The author proposed using the most popular tools for the investigation and classifying tools by functionality of their work, e.g., how it is more comfortable to end user to work with the tool. However, the research did not touch the layout criteria in detail, in other words, to evaluate user comfort by a set of different criteria, where the layout is one of them.

In research [11], the author proposed criteria to evaluate tools and suggested the rating system for tools selected. The research demonstrated a large variety of the case sets for evaluating the tools. But all the criteria again went through functionality of the tools and the author did not consider the criteria of the layout, but just stated the importance of performing a list of criteria needed for such analysis.

Based on this and other studies, we can conclude that evaluation of the layout is hard to make in the meaning of the proposed algorithm of layout used by tool. In [12], the author stated that most tools used graph layout for solving the layout problem of the UML and such approach was easy to be evaluated. The graph layout was also studied by [13]–[15]. But the graph layout algorithms did not match all the criteria for the layout of the UML sequence diagram, due to its specific presentation.

Also the author stated that the most common approach in the layout algorithms was made for UML layout of the class diagrams and other not complicated diagrams, even a lot of libraries build for this purpose. However, more complicated diagrams, such as sequence or activity diagrams, have drawbacks in layout because of complexity and difficulties in adopting them to standard graph layout algorithms. The author also stated that this evaluation was hard to perform because of the lack of a certain algorithm for such layout. Also this evidence was stated by [16].

The author also stated that for more easy implementation vendors of the tools tried to separate complex diagrams to layers, e.g., object layout was processed by one algorithm and labels of these objects and message connections for these objects were processed by another; however, this approach led to complex decisions and difficulties in evaluation of layout criteria for the diagram at all.

In the previous research, the authors provided evidence that the problem of algorithm for the automatic UML diagram layout still exists, and it was widely discussed with regard to sequence diagrams. The authors provided information that the main problem was that the algorithms for the automated layout playing the main role in the drawing diagrams had not been well studied. The authors mentioned several algorithms that allowed us to understand the roots of the automated layout and gave us a possibility to analyse the concept of the automated layout for the UML sequence diagram.

The understanding of how each algorithm follows the criteria for diagram layout will give better understanding, if it is easy for tool vendors to provide a possibility of automated layout.

The author of [17] gave a clear idea that there was no one straight way of accomplishing automated layout and several technics should be used. Also the author studied a topology shape approach. The core concept of this approach was based on enhancement of the original graph. As a result of this approach, orthogonal graphs were taken. In this approach, crossings in the graph were taken into consideration as the main aspect. Also the approach lied in the comparison of shapes of the elements of the graph and making decision ether deformation of the graph should be performed or changing the length of the edges without touching angles of the elements of graph.

Another approach mentioned in the previous study was hierarchical graph drawing. The idea of this approach was to represent the graph or the diagram in the view of layers with the edges generally directed downwards [17], [18].

Divide and conquer approach relies on the dividing of graph into subgraphs or decomposing the graph and as the next step positioning the subgraphs according to their types, e.g. linear, cyclic [17].

According to the force-directed approach, edges should be of more or less equal length and there are as few crossing edges as possible [17].

Multi-scale approach is the further development of the force-directed approach, also mentioned in the previous study, and it is based on the series of abstractions of the graph, performing relocations of vertically yield vertices, correcting global directions and performing fine-scale of all directions in the graph.

All the discussed studies do not provide an in-depth analysis and answer the question about how UML editors conform to the layout criteria stated for the UML sequence diagram. Therefore, the issue is still topical and studied by the authors of this paper.

III. Specific Layout of the UML Sequence Diagram

With UML sequence diagrams it is easily possible to describe interactions between system components and actors of its environment. It presents a sequence of messages that may occur during system start-up and monitors the messages exchanged during this run [19].

UML sequence diagram is a popular notation for the definition of scenarios of operations as its clear graphical layout processing gives instant intuitive understanding of system behaviour [19]. The UML sequence diagram is declared as one of the equivocal UML diagrams, with an implicit and unofficial semantics that designers can give to basic sequence diagram as a result of this conflict [19].

The UML sequence diagram shows the objects, their lifelines, and messages that are sent by objects-senders and received by objects-receivers. Sequence diagram is used to show the dynamic aspect of the system that in the object-oriented approach is indicated as message transfer between objects. The dynamics flow is defined by ordering of the message sending and receiving actions. It gives the ground for defining operations executed by objects to be grouped into classes, as well as showing and proving the dynamic aspect of class state transition.

The complexity of the structure of the sequence diagram raises the necessity of well-structured layout of this diagram. The right layout gives a possibility of easy interpretation and further implementation of this diagram.

Nowadays, UML diagram design is represented by the set of tools that allow formulating and structuring the layout of the diagrams. Such a variety of the tools could be explained by the interest of architects of using a uniform tool that will allow making diagrams more readable, which could be made by reaching high quality of the layout of the diagram [1].

Due to such a variety and the complexity of the UML sequence diagram, the first check point authors should investigate is the ability of the tool to perform diagram layout and to check either the tool provides an automatic layout or not (just, if any layout is built-in). The second point is if the tool can import model developed by another tool with the ability to keep the original layout. The third point is to check the correspondence of the layout algorithm offered according to the layout criteria stated for the UML sequence diagram. This step gives understanding if tools manage to provide good auto layout algorithms. Better auto layout algorithm can lead to reduction of the time consuming work on creation diagram and modification it to appropriate form that gives right position for the elements according to the layout criteria.

Table I shows the result of the research, how the tools are able to accomplish work on auto layout. In other words, Table I provides an analysis on the first and second check points, thus giving an ability to select the most appropriate tools for their analysis according to the third check point, i.e., according to the layout criteria, which are explained in detail in the next section of the paper.

The tools listed in Table I are selected as tools working with the UML sequence diagram and giving a possibility to perform automated layout. Also the argumentation to select tools is the tool presented as a standalone solution that does not require any additional software for the work. Moreover, the popularity of these tools is also taken into consideration based on the research results proven by [12], [16]. Table I demonstrates several tools that provide automatic diagram layout; moreover, the research has shown that not a big number of tools give this functionality. Some tools have a possibility to perform automatic layout, but do not have a possibility to perform the layout of the sequence diagram. The evaluation "YES/NO" means that specific characteristic is/is not supported.

TABLE I

RESULT OF EVALUATING TOOLS FOR UML LAYOUT SEQUENCE DIAGRAM

Tool	XML/XMI import/export	Auto layout	Compatible with other tools
Argo UML	XMI* import/export	NO	NO
Astah	XMI* import/export	NO	NO
Creatify	NO	NO	Visio
Dia	NO	NO	NO
Sparx Enterprise Architect	XMI import/export	YES**	Argo, Visual Paradigm
Visio	NO	NO	Creatify
MagicDraw	NO	NO	NO
Modelio	NO	NO	NO
OpenModelSphere	XMLS export	NO	NO
Papyrus	XMI export	YES**	NO
Rational software modeller	XMI import/export	YES	Argo
StarUml	XML import/export	NO	NO
Umbrello	XHTML export	NO	NO
Visual Paradigm	XML/XMI export/import	YES**	Argo, Enterprise Architect

* – only export of objects without layout

** – for not complex diagrams

From the research results listed in Table I, it can be seen that the variety of the UML tools does not completely propose auto layout and only few of the tools propose such functionality. Also we can see that tools give a possibility to make export of the result of the work, but it is impossible to import result of the work to another tool to provide better layout to the end user with the help of more advanced tools. In other words, it is impossible to use results of the work with another vendor. Another criterion that should be mentioned is that a huge part of the responsibility on diagram layout is given to the user that makes developing of the diagram complex and time consuming, also it could lead to a problem when the diagram could be designed in the hard-to-read and maintain format.

Still, the following tools are most appropriate for further analysis: Visual Paradigm, Papyrus, Rational software modeller, Sparx Enterprise architect. To analyse these tools, according to the third check point, the criteria for the UML sequence diagram are defined in the next section.

IV. DEFINITION OF THE CRITERIA FOR THE LAYOUT OF THE UML SEQUENCE DIAGRAM

Figure 1 depicts the examples of good and bad layout of the diagram and demonstrates the effects of bad layout of the diagram. The diagram on the left side of Fig. 1 demonstrates unreadable message flow; on the other hand, the right one represents an easy-to-read flow of the messages, structured using layout criteria.

As for sequence diagram specific criteria have to be generated in order to represent all interconnections between objects in a right way. Diagram is a convenient way of presenting information and much more clear than text information. While the scheme may be used to represent complex and difficult problems, they must be semantically and syntactically correct and well-layouted to give the desired result.

A good diagram needs to satisfy different criteria, e.g. aesthetic and layout criteria. For the representation of the diagrams the main role is readability of this diagram.

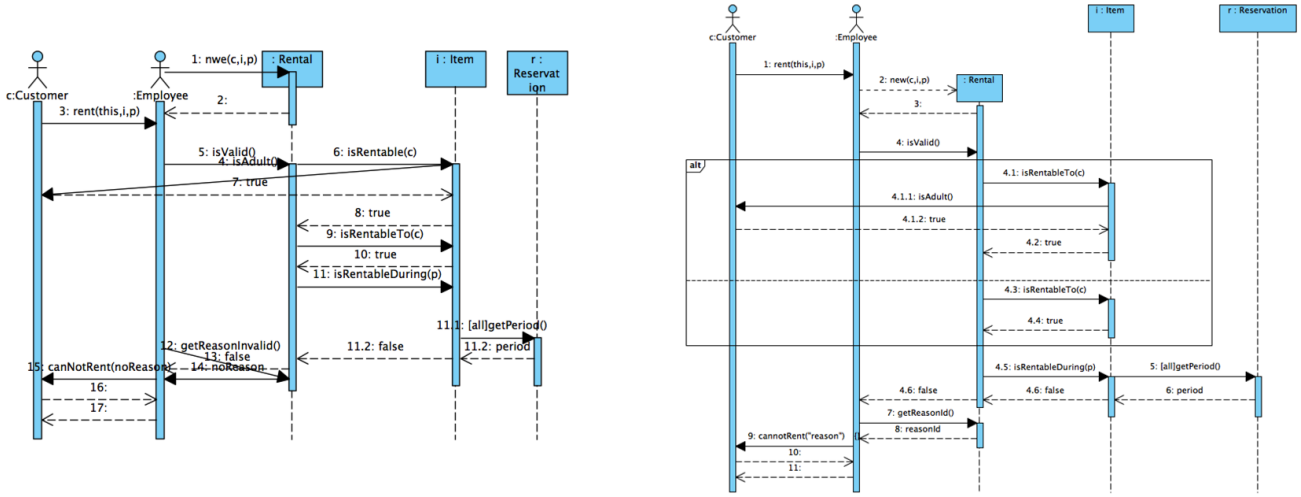

Fig. 1. Example of ill-structured and well-structured diagram.

Speaking about diagram layout criteria, we easily can say that diagrams have some common criteria that give the set of rules for their representation.

In previous research [20], the authors take a look at the research of the layout criteria. The authors mention that general diagram criteria and specific UML diagram layout criteria have also been studied by [5], [9], [21], [22] and others. All diagrams should conform to general graph layout criteria.

General layout criteria can clearly be formulated from the theory of perception [23]. In the theory of perception there are many criteria for unifying perception, but not all of them can be applied to all types of diagrams. Sequence diagram belongs to specific UML diagrams and this diagram has to cope with additional criteria, e.g., slidability. Organisation of the diagram elements has been studied in [5] and according to this study there are six perceptual principles, when the elements are considered to be a group. These principles are acquired from Gestalt theory [24] that formulates the set of laws.

Laws give restriction to the diagrams by simplifying them – the simpler the diagram, not containing too much information, the easier it is to understand. Elements of similar shape or colour often can be perceived as a group. Elements can be perceived more easily according to a smooth path. Those elements, which are connected with curved complex lines, are more difficult to perceive together. Grouping elements give a possibility to take these elements as a group, if elements are connected with a line or in other way physically. Elements that look similar are perceived together.

There are three more principles related to perceptual element segregation that explains that diagram's symmetric parts are perceived as separated. Horizontally or vertically oriented elements are more likely to be perceived as separated figures than shapes in an angle. One more principle of grouping the elements is that elements with contour are seen separately and all elements in this contour are perceived as a group.

All of these Gestalt theory principles are considered as aesthetic criteria. General aesthetic criteria are widely discussed in [5], [21], [25], [26], [27] and [20].

Speaking about visual presentation it is easy to conclude that the UML sequence diagram is specific. All the objects are positioned horizontally at the top of the diagram and the life lines are drawn vertically from the top to the bottom of the diagram. This specific of the diagram shows that the criteria for the UML sequence diagram should be selected from existing or even modified ones, so they could be applicable to the sequence diagram, e.g., one specific criterion for sequence diagram called "Correct sequence of messages", which is the core concept of this diagram. The authors of [21] and [5] have identified the criteria specific for sequence diagrams. Table II shows the list of these criteria. Criteria are marked with SD identifier.

The list of the criteria combined from the previous research made by the authors of the paper are divided into three groups for better manipulating, and perhaps for further implementation of a new layout algorithm:

1) Vertical alignment – describes how to better situate vertically positioned elements:
   - SD0 Precise sequence of messages. The Requirement of notational convention of the UML is to display messages in the order they are being sent.
   - SD1 Avoid object and lifeline overlapping. It makes difficult or sometimes impossible to read diagram when objects or lifelines are overlapping.
   - SD2 Minimize crossings. Understanding of the diagram is more complex with crossings of message arrows. This criterion says that message arrows should not cross at all.
   - SD3 Improve slidability. Slidability is an aesthetic criterion for better transparency, particularly it is important in bigger sequence diagrams, where the whole diagram fails to fit in one screen. Slidability means that a fixed size window can be placed on the diagram and slid over it in such a way that all senders and receivers of messages, which are in this window, fit in too.

2) Horizontal alignment - describes how to better situate horizontally positioned elements:
   - SD4 Message arrow length minimisation. To minimize message arrow length for making the diagram more comprehensible and the region smaller.
   - SD5 Reduction of number of long message arrows. It is better to limit message arrow length because of difficulty to follow long message arrows.
   - SD6 Minimize longest message arrow length. The elements of the diagram should be placed closer to make message arrows shorter as possible.
   - SD7 Uniform message arrow length. To make diagram more understandable it is better to perform message arrows with similar length. Similar arrow length is also needed to fulfil the slidability criteria.
3) Criteria applied for layout of the whole diagram – describe positioning elements in the appropriate way for better reading possibility and easier to maintain for the end user:
   - SD8 Elements need to be arranged orthogonally. Sequence diagram is an example of orthogonal diagram – message arrows are situated horizontally (typically) and lifelines – vertically.
   - SD9 Diagram flow. It is very important to layout elements by creating obvious flow. Diagram should have start and end that gives a possibility to follow the elements and read the diagram easier. Usually the first message is located at the top left corner of sequence diagram.
   - SD10 Subset separation. Sequence diagram has subsets if there is one such message eliminating which, two unconnected sequence diagrams are formed.
   - SD11 Employ symmetry. [5] and [21] believe that symmetry should also be considered in sequence diagrams. For this criterion there is no special definition, but this criterion is one of the most important aesthetic criteria. For implementation of this criterion a decision was made to look closer on the criteria formulated from graph drawing theory and related to common UML rules.

The final list of criteria includes twelve criteria. All criteria are collected for the evaluation of layout possibilities of the set of tools selected in Section 3, or in another words how each tool supports them.

V. Analysis of the Criteria Support by UML Modelling Tools for the Layout of the UML Sequence Diagram

The analysis of the tools made in Section 3 gives the evidence that automated layout is not so easy element of the tool to be implemented as not all tools accomplish to include it to their products.

To evaluate quality of the implemented automated layout it would be good to see if the implemented feature follows all the criteria described in Section 4.

In turn, to evaluate tools selected in Section 3 according to the criteria defined in Section 4, Table II depicts the result of the research on how tools follow the criteria for UML sequence diagram while performing automatic layout. Evaluation of the criterion as "YES" shows the possibility of the tool to follow the criterion and "NO" opposite. Evaluation of the criterion as "Partial" means that he tool could not follow the criterion in full. From these results it is possible to conclude that no one tool completes the task of the layout of the UML sequence diagram. As seen from Table II, most part of the criteria is not supported by the tools at all.

For more precise understanding of the problem the authors made a decision to make classification of existing algorithms for layout according to the criteria clarified in Section 4. The algorithms are not defined to accomplish the layout of the sequence diagram, but can be considered the potential solutions for constructing the algorithms for automated layout of the UML sequence diagram. The results of this investigation are shown in Table III.

BrainTool as a tool for generation of the UML diagrams, including the UML sequence diagram, is analysed among the algorithms according to the criteria, not among UML editors, because this tool is not the CASE tool for UML design, but the tool that creates sequence diagram. Therefore, the algorithm of the diagram layout is already built-in solution invented by the authors in their previous research. Thus, BrainTool is positioned not as a tool, but rather as an algorithm for the diagram layout.

TABLE II

Tools Criteria Evaluation for Layout of the UML Sequence Diagram

ID	Name of the criterion	Visual paradigm	Papyrus	Rational software	Sparx
SD0	Precise sequence of messages	PARTLY	YES	YES	PARTLY
SD1	Avoid object and lifeline overlapping	YES	NO	NO	YES
SD2	Minimize longest message arrow length	PARTLY	NO	PARTLY	NO
SD3	Improve "slidability"	NO	NO	NO	NO
SD4	Minimize crossings	NO	NO	NO	NO
SD5	Message arrow length minimisation	NO	NO	NO	NO
SD6	Reduction of long message arrow number	NO	NO	NO	NO
SD7	Uniform message arrow length	NO	NO	NO	NO
SD8	Elements need to be arranged orthogonally	YES	NO	PARTLY	PARTLY
SD9	Diagram flow	NO	NO	NO	NO
SD10	Subset separation	NO	NO	NO	NO
SD11	Employ symmetry	NO	NO	NO	NO

TABLE III

ALGORITHM CRITERIA EVALUATION FOR LAYOUT OF THE UML SEQUENCE DIAGRAM

Abbreviations used in the table are the following: TSMA – Topology-Shape-Metrics Approach, HA – Hierarchical Approach, VA – Visualisation Approach, DCA – Divide and Conquer Approach, FDA – Force-Directed Approach, MSA – Multi-Scale Algorithms, GA – Genetic Algorithms, BT – BrainTool, adj – adjustable.

ID	Name of criterion	TSMA	HA	VA	DCA	FDA	MSA	GA	BT
SD0	Precise sequence of messages	YES	YES	YES	YES	YES	YES	ADJ	YES
SD1	Avoid object and lifeline overlapping	YES	NO	NO	NO	NO	ADJ	ADJ	YES
SD2	Minimise longest message arrow length	YES	YES	NO	ADJ	YES	NO	ADJ	YES
SD3	Improve "slidability"	NO	NO	NO	YES	YES	NO	ADJ	NO
SD4	Minimise crossings	ADJ	ADJ	ADJ	ADJ	ADJ	ADJ	ADJ	YES
SD5	Message arrow length minimisation	ADJ	ADJ	ADJ	ADJ	ADJ	ADJ	ADJ	YES
SD6	Reduction of long message arrow number	ADJ	ADJ	ADJ	ADJ	ADJ	ADJ	ADJ	YES
SD7	Uniform message arrow length	NO	NO	NO	NO	NO	NO	ADJ	NO
SD8	Elements need to be arranged orthogonally	NO	NO	NO	NO	NO	NO	ADJ	YES
SD9	Diagram flow	YES	YES	YES	ADJ	YES	NO	ADJ	YES
SD10	Subset separation	NO		NO	NO	NO	NO	NO	NO
SD11	Employ symmetry	NO	NO	NO	NO	NO	NO	NO	NO

The results of the algorithm evaluation according the layout criteria give the possibility to say that not all algorithms can be taken as a full implementation of the universal algorithm for the automated layout of the sequence diagram. Also it is seen that algorithm made by the authors support the most set of the criteria defined while following all criteria for the right layout of the diagram. However, there are still criteria that are not followed by even this algorithm.

VI. CONCLUSION AND FUTURE RESEARCH

Nowadays, UML is widely used to represent system specification at different levels of system abstraction. A big set of CASE tools gives the possibility to accomplish modelling of the diagrams with their means; also, vendors show a variety of functions to the end user. Two of the most important functions of the CASE tools are: to accomplish automated layout; to perform import and export possibilities. Modelling diagrams is time-consumptive work as stated in the paper and it is a very complex task to the user to follow all the criteria of the right layout of the diagram. As for import and export possibilities it could be very comfortable to have a possibility of moving all work to another software presented by another vendor as not all companies use the same software.

UML sequence diagram is a popular notation for the definition of scenarios of operations as its clear graphical layout processing gives instant intuitive understanding of system behaviour.

The research object of this paper is the quality of the layout of the UML sequence diagram in modern CASE tools also as possibility of automated layout of the UML sequence diagram.

The research was divided into three main steps. The first step was to collect a set of tools and define if they propose to the end user working with sequence diagram and accomplishing work with automated layout. The paper describes the set of chosen tool and shows that not all tools manage to perform automated layout for the UML sequence diagram.

The second step was defining if tools managed to perform import and export possibilities. Paper shows evidence that most part of tools managed to work with export and import possibilities but for most of the tools migration of the performed work from one tool to another did not give successful result and this possibility was designed only for internal work.

The third step was to create and group criteria of "good" layout of the sequence diagram. During the research, twelve criteria of the layout of the sequence diagram were summarised. Also within the scope of research evaluation of layout was performed evaluation of the layout algorithms according to the criteria, was define how they could follow the criteria described in the paper. The investigation showed that all tools but one did not follow the criteria of the layout of the sequence diagram. The only tool that followed nine criteria of twelve was BrainTool designed not for modelling of the sequence diagram but for generating it.

These results of the research give background and show high importance of further research. Further research has to raise questions on developing algorithm of automatic layout of the sequence diagram that follows all criteria for better layout of the sequence diagram.

Also there is an issue on a possibility to provide import/export possibilities of tools. The research should provide better knowledge on how XMI standard is followed by tools while accomplishing import and export possibilities. The research should also focus on finding possible solutions for developing a tool as a middle layer between CASE tools to accomplish migration of the work from one tool to another.

ACKNOWLEDGMENT

The research presented in the paper is supported by Accenture Latvian Branch, project No. L7950 "Development of Model Transformation Tool Prototype", and by Latvian Council of Science, No. 342/2012 "Development of Models and Methods Based on Distributed Artificial Intelligence, Knowledge Management and Advanced Web Technologies".

REFERENCES

[1] UML, *Unified Modeling Language*. [Online] Available: http://www.uml.org/

[2] *W3C XML Schema*. [Online]. Available: http://www.w3.org/XML/Schema [Accessed: Mar. 21, 2016].

[3] A. Galapovs and O. Nikiforova, "UML Diagram Layouting: the State of the Art," *Scientific Journal of Riga Technical University. Computer Science. Applied Computer Systems*, vol. 47, pp. 101–108, 2011, [Online]. Available: https://ortus.rtu.lv/science/lv/publications/ [Accessed: Mar. 21, 2016].

[4] A. Galapovs, O. Nikiforova, "Several Issues on the Definition of Algorithm for the Layout of the UML Class Diagram," in *3rd Int. Workshop on Model Driven Architecture and Modelling Driven Software Development In conjunction with the 6th Int. Conf. on Evaluation of Novel Approaches to Software Engineering*, June 8–11, 2011, Beijing, China. SciTePress Digital Library 2011.

[5] O. Nikiforova, N. Pavlova, "Development of the Tool for Generation of UML Class Diagram from Two-hemisphere model," in *Proc. of The Third International Conference on Software Engineering Advances*, ICSEA, *International Workshop on Enterprise Information Systems*, ENTISY, Mannaert H., Dini C., Ohta T., Pellerin R. (Eds.), Published by IEEE Computer Society, Conference Proceedings Services (CPS), pp. 105–112, 2008. http://dx.doi.org/10.1109/icsea.2008.37

[6] O. Nikiforova *et al*. "BrainTool. A Tool for Generation of the UML Class Diagrams," in *Proc. of the Seventh Int. Conf. on Software Engineering Advances*, Mannaert H. et al. (Eds), IARIA, Lisbon, Portugal, Nov. 18–23, 2012, pp. 60–69.

[7] O. Nikiforova *et al*. "BrainTool v2.0 for Software Modelling in UML," *Scientific Journal of Riga Technical University: Applied Computer Systems*, Grundspenkis J. *et al*. (Eds), vol. 16, 2014, pp. 33–42. http://dx.doi.org/10.1515/acss-2014-0011

[8] O. Nikiforova, N. Pavlova, K. Gusarovs, O. Gorbiks, J. Vorotilovs, A. Zaharovs, D. Umanovskis, and J. Sejans, Eds., "Development of the Tool for Transformation of the Two-Hemisphere Model to the UML Class Diagram: Technical Solutions and Lessons Learned," in *Proc. of the 5th Int. Scientific Conf. „Applied Information and Communication Technology*, April 26–27, 2012, Jelgava, Latvia.

[9] Visual Paradigm. (2011, May). *Generate Sequence Diagram from Use Case Flow of Events*, [Online]. Available: http://www.visual-paradigm.com/product/vpuml/tutorials/gensdfromfoe.jsp [Accessed: Mar. 21, 2016].

[10] Wan Hashira Wan Husting *et al*. "Investigation of diagrams layout," *Faculty of Computer Science & Information Technology*, University of Malaya, 50603 Kuala Lumpur, MALAYSIA, 2007.

[11] S. M. Thomas, "Evaluation of UML tools using an end-to-end application," Ph.D. dissertation, Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, 2004.

[12] H. Fuhrmann, M. Sponemann, M. Matzen, "Automatic Layout and Structure-Based Editing of UML Diagrams" Department of Computer Science Christian-Albrechts-Universitat Kiel, Germany, 2010.

[13] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, "Graph Drawing," in *Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[14] M. Kaufmann and D. Wagner, Eds., "Drawing Graphs" in *Methods and Models*, ser. LNCS. Berlin, Germany: Springer-Verlag, 2001, no. 2025.

[15] M. Junger and P. Mutzel, "Graph Drawing Software," Springer, Oct. 2003.

[16] G. Hoops, "Automatic Layout of UML Sequence diagram," Diplomarbeit eingereicht im Jahr 2013 Christian-Albrechts-Universität zu Kiel 2013.

[17] G. Di Battista, P. Eades, R. Tamassia, I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall, Upper Saddle River, 1999.

[18] P. Healy, S. N. Nikolov, "Hierarchical Graph Drawing," R. Tamassia eds., *Handbook of Graph Drawing and Visualization*, CRC Press, pp. 409–453.

[19] C. Sibertin-Blanc, N. Hameurlain and O. Tahir, "Ambiguity and structural properties of basic sequence diagrams," in *Innovations Syst. Softw. Eng.* vol. 4, pp. 275–284, 2008. http://dx.doi.org/10.1007/s11334-008-0063-2

[20] O. Nikiforova, D. Ahilcenoka, D. Ungurs, K. Gusarovs, L. Kozacenko, "Several Issues on the Layout of the UML Sequence and Class Diagram," in *Proc. of the 9th Int. Conf. on Software Engineering Advances*, ICSEA 2014, France, Nice, 12–16 Oct., 2014. Wilmington: IARIA, 2014, pp. 40–47.

[21] Visual Paradigm, *Drawing activity diagrams*. [Online] Available: http://www.visualparadigm.com/support/documents/vpumluserguide/94/200/6713_drawingactiv.html [Accessed: Mar. 21, 2016].

[22] Sparx systems, *Enterprise Architect*. Available: http://www.sparxsystems.com.au/ [Accessed: Mar. 21, 2016].

[23] H.C. Purchase, J-A. Allder and D. Carrington, "Graph Layout Aesthetics in UML Diagrams: User Preferences," *J. of Graph Algorithms and Applications*, vol. 6, no. 3, pp. 255–279, 2002. [Online]. Available: Universitat Trier. http://dx.doi.org/10.7155/jgaa.00054

[24] A. A. A. Jilani, M. Usman, Z. Halim, "Model Transformations in Model Driven Architecture," *Universal J. of Computer Science and Engineering Technology*, vol. 1, no. 1, pp. 50–54, October 2010. [Online] Available: UNICSE, http://www.unicse.org/. [Accessed May 28, 2012]

[25] M. Kardos, M. Drozdova, "Analytical Method of CIM to PIM Transformation in Model Driven Architecture (MDA)," *J. of Information and Organizational Sciences*, vol. 34, no. 1, pp. 89–99, May 2010. [Abstract]. Available: http://jios.foi.hr/index.php/jios/article/view/163. [Accessed: Mar. 21, 2016].

[26] A. Kleppe, J. Warmer and W. Ba, *MDA Explained: The Model Driven Architecture: Practice and Promise* USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

[27] K. Hamilton and R. Miles, *Learning UML 2.0*, USA: O'Reilly, 2006.

Oksana Nikiforova received the Doctoral degree in Information Technologies (system analysis, modelling and design) from Riga Technical University, Latvia, in 2001.
She is a Professor at the Department of Applied Computer Science, Riga Technical University. Her current research interests include object-oriented system analysis, design and modelling, especially the issues in model driven software development.
E-mail: oksana.nikiforova@rtu.lv

Sergii Putintsev received the Bachelor's degree in Computer Systems and Networks from *National Aerospace University – "Kharkiv Aviation Institute", Ukraine,* in 2014.
He is the second-year Master student at the Faculty of Computer Science and Information Technology, Riga Technical University.
His current research interests include problems and solutions for layout of the UML sequence diagrams.
E-mail: putintsev.sergii@gmail.com

Dace Ahilcenoka received the Bachelor' degree in Computer Systems from Riga Technical University, Latvia, in 2012.
She is the first-year Master student and Research Assistant at the Department of Applied Computer Science, Riga Technical University.
Her current research interests include UML diagram layouting, algorithms of diagram layout.
E-mail: dace.ahilcenoka@rtu.com