Characterization of Techniques and Tools of Visualization Applied to Software Comprehension

A Systematic Mapping

Marllos Paiva Prado
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: marllosprado@inf.ufg.br

Auri Marcelo Rizzo Vincenzi
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: auri@inf.ufg.br

Fabrízzio Alphonsus A. de M. N. Soares
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: fabrizzio@inf.ufg.br

Felipe César
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: kamuie4ever@hotmail.com

Guilherme Pereira de Paula
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: guilhermepaula@inf.ufg.br

Hugo Alexandre Dantas do Nascimento
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: hadn@inf.ufg.br

João Carlos Silva
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: jcs@inf.ufg.br

Juliano Lopes de Oliveira
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: juliano@inf.ufg.br

Lucas Carvalho Lima
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: lucaslima@inf.ufg.br

Thiago Fernandes
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail:hidekownz4@hotmail.com

***Abstract*—This study aimed to make a characterization about how information visualization has been applied on software comprehension with relation to techniques and tools proposed by literature. Systematic Mapping was adopted as the method to guide the investigation process. The findings, although not being definitive, points direction over important questions such as what kind of artifacts and life-cycle phase have been more considered. It was also investigated how these studies have been evaluated, how they evolved in the main digital libraries over the last decade, and what points deserves further attention, through new research.**

Keywords – Systematic; Mapping; Visualization; Sofware Comprehension

I. INTRODUCTION

Information Visualization applied to Computer Science is committed to the visual representation of abstract data handled by computer and to interaction as a way of magnifying the cognition [1]. Data to be modeled does not necessarily need to have an intrinsic geometry shape previously associated [2]. This research field aims to develop and apply visual models to explore the human cognitive abilities of recognizing and deriving information from graphics of important data and their relationships [3].

One of the best ways to minimize the complexity of creating/maintaining a system is to simplify its understanding. At this point, an adequate visualization of the artifacts and information generated in the development process can be a favorable factor. The reason is that human vision has advantageous attributes such as the power to capture a high amount of information in parallel, considering a short period of time, and also the capacity to focus attention in a object's fragment of interest while not losing the attention of what happens at its surrounding [4]. Therefore, the characterization of manner and goals in which these proposals have been applied is a way of better knowing the existing solutions and what research opportunities are opened in this field.

Experimental Software Engineering provides guidance in a well standard and organized conduction of software engineering experiments. As mentioned by Wohlin et al. [5], experimental studies let one object of interest to be evaluated by different people and environments. The more an experiment is repeated, considering different contexts, the more information is obtained about the object of study, so that the results are more significant.

Two kinds of experimental studies have been disseminated in the research community: systematic review [6] and systematic mapping [7]. Both are secondary experimental studies and differ subtly. While the systematic mapping focuses at mapping the related research to some question of interest through a detailed categorization of primary related study, the systematic review is more restrictive, as a way to identify, evaluate and compare qualitatively all the relevant research to a research question.

The next five sections of this paper present the definition, execution and results obtained from a systematic mapping to characterize the visualization applied to the software comprehension. The adopted process on this work followed the definitions of Petersen, Feldt, Mujtaba and Mattsson [7] and Kitchenham [8] once these works target the definition of parameters for experimental studies applied to software engineering themes. Paper organization follows the sequence order adopted at StArt tool [9]. Section II describes the method and the parameters adopted in the planning and definition of a study. Description of the intermediate steps and results gotten from the investigation conduction as stated in the pre-established plan is defined and related at Section III. Section IV presents results reached after extraction step in the systematic review and gather relevant information observed from the analysis. Section V raises question about threats of validity concerning this study. Section VI concludes this paper and discusses future research.

II. PLANNING

Systematic Mapping planning is composed by goals definition, protocol, research strategy and inclusion/exclusion criteria to be adopted. Protocol consists of research question definition, population, intervention, control, results and application. These parameters were specified as follows.

A. *Research Goals*

This research consists of characterizing how visualization has been used at software comprehension, through the identification of papers which discuss tools and techniques applied to software comprehension.

B. *Main Questions*

Based on the defined goal, the following research questions were formalized:

- R.Q.1. How publications about visualization tools and techniques, applied to software comprehension, have been evolved in the main digital libraries?
- R.Q.2. How visualization tools and techniques, applied to software comprehension, have been evaluated?
- R.Q.3 What is the profile of visualization tools and techniques, applied to software comprehension, considering the artifacts represented, life-cycle phase and training?

C. *Population*

The population considered was composed by researchers and developers, who use/propose information visualization at software comprehension and publish them at indexed electronic databases.

D. *Intervention*

The observed characteristic was the application of tools and visualization techniques to the software comprehension. Characteristic Observation was made from software engineering researchers' point of view.

E. *Control*

A total of four relevant papers [10] [11] [12] [13] were previously set by experts to be used as the control for the search string. The search result shall be considered adequate in case of returning all these papers in the considered databases.

F. *Results*

The expected result in the end of the systematic mapping is the characterization of relevant information about visualization tools/technique application at software comprehension.

G. *Application*

Systematic mapping results should collaborate to a better understanding of how visualization has been applied until the moment at software comprehension, so it lets the identification of weakness and opportunities in this research field.

H. *Research Strategy and Search String*

Most bibliographic studies, including systematic mapping and reviews, are made through an automatic search over digital libraries and using a pre-defined search string. This string is a set of combined key-words which reflects the search to be made, organized in a way to guarantee that the returned results of the search be closer as possible from the scope. According to the formulated research questions, the following search string was used:

- *Q0: ("technique*"OR "tool*") AND ("visuali*ation") AND ("comprehension"OR "understanding") AND ("software"OR "program")*

Following the recommendations from Kitchenham [8], besides key-words, specific synonymies and spelling variations to our research question were considered. This approach was employed to increase the number of returned papers and to avoid that important papers to our research question were neglected because of synonymy word, e.g., “understanding” instead of “comprehension”.

I. Tools and Instrumentation

The digital libraries adopted to do the search in this systematic mapping were: “IEEE Xplore”, “ACM” and “Science Direct”. Additionally, it was applied a manual search to complement it, that is, a search made at Google website, using some terms of the search string and applying a case by case analysis to decide if the paper would be pertinent or not.

During the execution step, all returned papers after applying the search string in the digital libraries should be first considered as relevant and added to the StArt tool [9], for classification aftermost. The same has occurred in the manual search. This approach was followed because StArt tool divides the execution process in three phases:

1. **Study Identification**: Phase to when the automatic and manual returned papers are added to the tool. Information such as name, title, author, abstract, publishing source and year of the paper are organized to make reviser work easier, so the study conduction can become a less costly work.
2. **Selection**: Phase when an initial search is made. At this step, revisers are responsible for the abstract reading and to approve or reject the paper to the next step. This step is of great relevance, since it is the first phase which eliminates from the whole paper set, those that clearly are not part of the research scope.
3. **Extraction**: Phase when the selected papers from previous step are read and decision to accept or not the article to the final step is definitive. Besides that, the tool lets pre-defined classification studies on the plan to be verified. These data are after used in the automatic generation of statistics which will serve as the bases to the final conclusions of the study.

J. Inclusion and Exclusion Criteria

The following restrictions were adopted to eliminate papers considered not relevant to the study, on selection step:

- R.1. Studies which are not written in English Language.
- R.2. Studies which were not published in conference proceedings or journals.
- R.3. Studies prior to the year 2003.

Besides restrictions described, the following inclusion criteria (I.C.) and exclusion criterion (E.C) were adopted on the selection and extraction phase, related to the papers content:

- I.C.1. Addresses the application of visualization tools in the software comprehension.
- I.C.2. Addresses the application of visualization techniques in the software comprehension.
- I.C.3. Experimental studies about visualization techniques or tools in the software comprehension.
- E.C.1. Does not address the application of visualization tools or techniques in the software comprehension.

To answer the research question it is needed to analyze data collected by study conduction. These data, in turn, are obtained observing the classification criteria. Therefore, these classification criteria should represent objective and coherent properties to the parameters intervention defined in the plan. To each criteria established, the value “not adequate” was created to classify papers which do not meet any of the pre-defined values to the referred criteria, or cannot be considered a primary study. The latter intended to avoid that properties of a specific classification criterion could count for a work whose technique has a secondary focus, for example. In this study, the following classification criteria (C.C.) were adopted:

- C.C.1 – Source: Responsible to register the study source: (i) IEEE; (ii) ACM; (iii) Science Direct.
- C.C.2 – Focus: Responsible to register the study focus. Options: (i) Tools; (ii) Technique; (iii) Study – In the case that paper reports an experimental study about tools and/or technique application in software comprehension.
- C.C.3 – Evaluation Context: Responsible to register tests and results presented on the study. Options: (i) Tested in production context; (ii) Tested in academic environment; (iii) Does not show any evaluation result; (iv) Not Adequate.
- C.C.4 – Analysis Criterion: Responsible to register the way tool/technique generates the visualization. Options: (i) Static Analysis; (ii) Dynamic Analysis; (iii) Both; (iv) Not Adequate.
- C.C.5 – Object of Analysis: Responsible to register the object(s) represented in the visualization. Options: (i) Source Code; (ii) Execution Tracing; (iii) Documentation; (iv) Memory; (v) UML Diagram; (vi) Graphical Interface; (vii) Communication Register; (viii) Threads; (ix) Other artifacts; (x) Not Adequate.
- C.C.6 – Representation Type: Responsible to register the type of representation used in the visualization. Options: (i) 2-D; (ii) 3-D; (iii) Both; (iv) Not Adequate.
- C.C.7 – Life Cycle Scope: Responsible to register the knowledge focus of application of the visualization tool/technique. Options (i) Requirements; (ii) Construction; (iii) Test; (iv) Quality; (v) Maintenance; (vi) Design; (vii) Multiples; (viii) Not Adequate..
- C.C.8 – Learning Aided: Responsible to register if the tool/technique used the visualization to help the learning/training of beginners. Options: (i) Yes; (ii) No; (iii) Not Adequate.
- C.C.9 – Specific Platform or Language: (i) Name of the specific platform or language of the tool. (ii) Not Adequate

 ISBN: 978-1-61208-304-9

- C.C.10 – Tool Name: Responsible to register the tool name.

III. MAPPING CONDUCTION

The following subsections present the results of extraction and selection phases of systematic mapping based on the defined plan. This study was realized between November 2012 and February 2013.

During the Study Identification phase, the searches were applied and returned papers were added to the StArt [9] tool for analysis. Summing the total of returned papers by the automatic and manual search, it was obtained a total of 449 papers. From this total, 116 were considered duplicated and removed automatically by the tool. So, it was left a set of 333 papers to be evaluated.

A. Selection

At selection phase, 333 papers were submitted to analysis process. This process was defined in the following way: Initially, each paper's abstract, title and key-words were read by two revisers, in a way that determination made (inclusion and exclusion criteria) would be enough safe decided. Then, each reviser gives its advising. If both voted for the paper rejection, the paper was automatically rejected. If both voted for its preliminary acceptance, the paper was accepted until the next phase, when decisions made would be final. In case of divergent opinions, i.e., one reviser voted for its exclusion and the other for its inclusion, the paper was set as accepted, to be read on the next phase. Additionally, there were cases when two or more papers of the same author and describing the same technique were found. In this case, only the most recent paper was considered.

Applying the analysis process, the 333 papers were pre-classified. From this total, 228 (68%) of the papers were rejected (eliminated) and 105 (32%) of the papers were accepted and passed to the next step.

B. Extraction

During the extraction phase, each of the 105 papers identified on the previous phase were read and classified permanently according to the inclusion and exclusion criteria. As it can be observed in Table I, from the 105 studies, 3 were considered duplicated, 27 were rejected and 75 were accepted.

TABLE I. PAPER STATUS X MAPPING PHASE

Phase	**Status**		
	Duplicated	***Rejected***	***Accepted***
Identification and Selection	116	228	105
Extraction	3	27	75
Total	119	225	75
Percentage	26%	57%	17%

All the accepted papers were also evaluated to the classification criteria so it could be possible to make the study analysis, presented in the next section.

IV. DATA ANALYSIS AND CHARACTERIZATION

This section is dedicated to analyze data and answer the research questions based on the individual analysis of the mapping results and crossing data provided by classification criteria.

A. Use of Papers and Evaluation of Selection Quality

Figure 1 (a) presents results of the number of occurrence along the defined period of time, between before-selection and after-extraction phases. Observing the dark gray portion of the graphic it is possible to observe that the amount of papers selected after mapping kept fairly constant over the years. The observation of the difference between the light and dark area allows it to be noted that there was a follow-up between the evolutions of returned items and selected articles in the period analyzed and the difference was maintained between approximately 30 and 60 occurrences. Two observations can be made based on this information: (i) the Search String was compatible with the defined inclusion and exclusion criteria. That is, even before the application of the criteria, the oscillation of total papers returned along the years is almost similar to the oscillation of papers which relate to the research question in fact; (ii) the use of articles at the end of the mapping was 16% compared to the initial total. However, over the 449 initial papers, 116 were duplicated, that is, about 25%. Therefore, the actual use is approximately 22%.

B. Research Questions

The answers to the established research questions presented at Section II are discussed bellow.

R.Q.1. How publications about visualization tools and techniques, applied to software comprehension, have been evolved in the main digital libraries?

Figure 1 (b) presents a comparison of papers separated by digital library obtained from search string application before selection phase and after extraction phase. Analyzing the solid lines of the graphic, it can be observed that both before (light gray) and after (dark gray) criteria application, IEEE is the digital library with the largest amount of publications related to the search string and research questions along the years. It is interesting to notice the small difference between both solid lines, once it indicates the search string and criteria were well adjusted to this digital library. Another interesting feature to be observed is the oscillation along the years between IEEE and ACM, that is, periods when IEEE has a growth on the number of publications related to the research question, ACM had a decrease and vice-versa, and that this cycle recurred year after year.

The study identified 56% (42/75) papers as related to visualization tools which help at software comprehension. One was classified as experimental study. The others 32 relate to techniques, some with prototypes already implemented others still in conception stage. Crossing "Source" classification criterion (C.C.1) data with the "Focus" (C.C.2), it can be noted, at Figure 2 (a), that IEEE

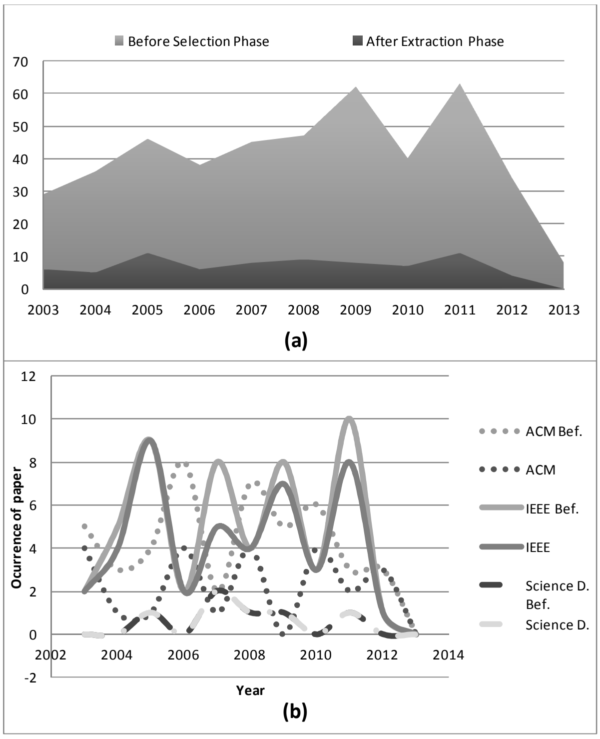

Figure 1. Graphic (a): Total ocurrence of paper before selection and after extraction phase; Graphic (b): The same information separated by digital library.

and ACM tool papers prevailed over those about technique, and Science Direct obtained the same quantity for both tool and technique. The only experimental study identified related to the research question [14] was published at IEEE.

R.Q.2. How visualization tools and techniques, applied to software comprehension, have been evaluated?

A total of 27% (20/75) of the tools/techniques does not demonstrate any kind of results of a practical evaluation, i.e., they do not indicate if the tool/technique proposed was tested to verify its efficacy/viability. From the others 73% (54/75), only 15% (11/75) were tested in a real development environment. Only 1.33% (1/75) paper was considered "Not Adequate". The other 58% (43/75) were tested in academic environment and had documented results.

Figure 2 (b) shows the values for "Evaluation Context" (C.C.3) along the years. Observing the prevailing area in light gray, it can be noted that evaluation context in academic environment (EAE) corresponds to the predominant form of evaluation, being followed by the non-evaluation of the proposal (NE). The evaluation in production context (EPC) remained modest, not exceeding 20% of the work until 2011.

This scenario reveals two important findings: (i) the studies need further validation to its practical viability; (ii) the test in academic environment has been the evaluation environment adopted in more than half of the studies but is not enough to represent the reality of a production environment which can mean a limitation on the external validity [15] of these proposals.

R.Q.3 What is the profile of visualization tools and techniques, applied to software comprehension, considering the artifacts represented, life-cycle phase and training?

More than half of the studies, 55% (41/75), considered the "Source Code" as the artifact used to generate the visualization. The Crossing between "Object of Analysis" (C.C.5) and "Analysis Criterion" (C.C.4) – Figure 2 (c) – allows identifying that "Static Analysis" corresponds to the kind of analysis most used, and the type of object that employs more this criterion in its representation is the "Source Code". The "Execution Tracing" corresponds to the object which most applies the "Dynamic Analysis" criterion. Also, few proposals of visualization apply both analyses.

A percentage of 81% (60/75) of tools/techniques defined corresponds to two-dimensional visualization, 18.7% (14/75) corresponds to tree-dimensional representations, 2.67% (2/75) employ both kinds of representation and one study was classified as "Not Adequate".

A total of 90.6% (68/75) of tools/techniques does not show any functionality to help the learning/training of new developers. Among other works, 8% (6/75) have focused on learning and one paper, 1.33%, is "Not Adequate" to any of the past options.

Observing the Graphic 6, generated by the crossing between "Representation Type" and "Learning Aided", it is possible to verify that the more used representation is the two-dimensional (2-D), which is about five times more employed than the tree-dimensional (3-D). Few visualization models adopt both types of representation, and regardless of the representation considered the learning aid remains little employed.

During the reading of the papers it could be observed that several tools/techniques identified do not allow the adequate scalability of the visualization. This problem attenuates when the tree-dimensional representation is used. The third dimension solves one of the problems of two-dimension visualization which is the lack of space. However, it seems to add a lot of complexity to the adopted visualization. Many were the times which was done the question "would this really help at software comprehension?" since exhibited examples are, generally, of hundreds of thousands of lines of code visualized in a small window. This may seems to be only a detail, considering the gains that these tools/techniques brings by its own existence, but the absence of this detail also is a limiting factor for an effective experience in user interaction.

 ISBN: 978-1-61208-304-9

These information can be used to delineate the profile of the papers: great part of these tools/techniques makes the static analysis of the source code, visualized in two-dimensional graphics with focus at help maintaining software. The reasons for the predominance of this pattern over others reveal an opportunity of research in this question, considering the causes involved. The few incidences of works with learning aid, reveals a deficiency and at the same time a potential research field. Aside it consists of a key factor for the acceptance of the new proposal by the starter user, the learning aid can accelerate and facilitate the training and conduction of experimental studies which evaluate, for example, the effectiveness of the same proposals, enabling their evolution.

V. THREATS OF VALIDITY

There are some threats of validity to our study. One of them is the fact the searches were made at only three digital libraries, which can restrict, in some aspects, the results. Some relevant digital libraries to the software engineering [16] were not consulted, due to time constraints imposed for the preliminary conduction of this mapping. In addition, the searches considered only results published in the last 10 years, and important papers may have been ignored even assuming these work evolutions were published inside the time window adopted. This approach was followed because the objective were to search tools/techniques which are being used currently, and it was considered that 10 years would be time enough to reach this objective.

It is also possible that we have chosen a search string which does not cover the whole set of relevant works. The software visualization field is somewhat wide and some important works, such as [17] [18], were not observed, due the fact of these papers do not use the key-words “comprehension” or “understanding” directly, for instance. The choose of classification criteria, although has been thought in a way to cover the high number of properties which characterizes this theme, can had ignored another characteristics as important and which collaborates to decrease threats related to the validity construction of the results [15]. It is also understandable that some papers, even adopting these terms, consider software visualization as a secondary focus on the software comprehension. Anyway, it was decided to follow with this string, once it returned the papers of control used, and so to guarantee that the results reflect our expectancy: papers about visualization focusing directly at software comprehension.

VI. CONCLUSIONS AND FUTURE WORK

As presented, this mapping work was divided in three parts: Planning, Conduction and Analysis. Conduction in turn, was subdivided in Study Identification, Selection and Extraction steps. The searches considered three important databases: ACM, IEEE and Science Direct and have identified initially 449 studies, from which only 75 were taken after applying the restrictions and selection criteria from Selection and Extraction phases. The data analysis lets the verifying of important information regarding how this research field has positioned with relation to the investigated

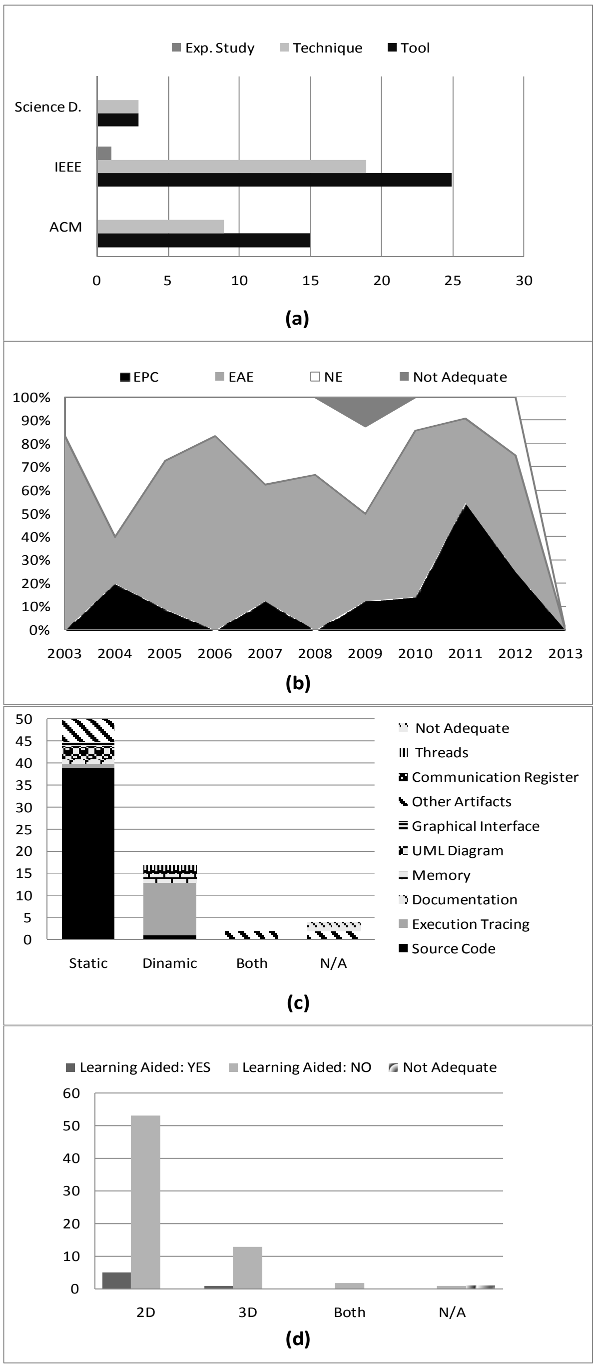

Figure 2. Criteria crossing for each graphic: (a) C.C.1 x C.C.2; (b) C.C.3 x Year; (c) C.C.4 x C.C.5; (d) C.C.6 x C.C.8.

theme. It was possible to identify, for example, a lack of tools/techniques of these works to support the training as well as the experimental validation of the pre proposes in the production environment. The preference for two-dimensional representation against three-dimensional and the lack of interest to questions related to interaction, considering the three-dimension, such as the question of the scalability, was another important point of this characterization. In addition, it was observed that the great majority of the proposes of visualizations adopts static analysis combined to source-code as software artifacts to be represented, which motivates the achievement of new experimental studies to investigate factors which influences these characteristics.

This work relates similarly to Bassil and Keller [13] in the sense that both are quantitative study, considering visualization tools related to software comprehension. Both works characterize some similar gaps and benefits in the field such as the importance of interaction issues and the visualization of source code aspects as being majority. The main differences are related to the considered source of information and the method applied. While the present work considers a systematic mapping, applied by software engineering researchers over scientific papers, the other was conducted as a survey with more than 100 participants (among researchers/users from industry) and considered a set of approximately 40 pre-defined tools.

This characterization study does not intend to be conclusive about the research questions investigated, considering the large extent of the covered theme and because we understand that some parameters of the protocol adopted such as the number of databases considered, researchers point of view, time windows established and other factors may let bias to the results. However, it can provide characteristic which still opened or are already consolidated and raises clues about cause-effect relations to the theme which deserves to be investigated.

In order to evolve the present work to future qualitative investigation, it should be considered visualization techniques to understand the importance and relationship of selected contributions to the research questions. Examples are the visualizations of co-authorship networks, citation graph, and impact ranking of author/paper.

Finally, this is a study based on an experimental process, yet established which lets the magnification/consolidation of the results through its replication. Artifacts generated in this study can be accessed in the following address [19].

ACKNOWLEDGMENT

The authors would like to thank the Instituto de Informática - INF/UFG, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES - Brasil, and Fundação de Amparo à Pesquisa do Estado de Goiás - FAPEG - Brasil, which support this work.

REFERENCES

[1] S. K. Card, J. Mackinlay, and B. Shneiderman, Readings in Information Visualization: Using Vision to Think. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

[2] R. Spence, Information Visualization: Design for Interaction, 2nd ed. London, England: Pearson Education Limited, 2007.

[3] P. R. G. Luzzardi, C. R. Andrade, and C. M. D. S. Freitas, " An Extended Set of Ergonomic Criteria for Information Visualization Techniques," in 7th Conference on Computer Graphics and Imaging, Kauai,Hawaii, 2004, pp. 236-241.

[4] C. Ware, Visual Thinking: For Design. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.

[5] C. Wohlin, et al., Experimentation in Software Engineering. Springer, 2012.

[6] J. Biolchini, P. G. Mian, A. C. C. Natali, and G. H. Travassos, "Systematic Review in Software Engineering," PESC-COPPE, UFRJ, Rio de Janeiro, Tech. Rep. ES-679/05, 2005.

[7] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in 12th International Conference on Evaluation and Assessment in Software Engineering, Bari, Italy, 2008, pp. 68-77.

[8] B. A. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," Keele University, Tech. Rep. EBSE-2007-01, 2007.

[9] E. Hernandes, A. Zamboni, and S. Fabbri, "Using GQM and TAM to evaluate StArt a tool that supports," CLEI Electronic Journal, vol. 15, no. 1, Apr. 2012, pp. 13-25.

[10] R. Wettel, M. Lanza, and R. Robbes, "Software systems as cities: a controlled experiment," in Proceedings of the 33rd International Conference on Software Engineering (ICSE '11), New York, NY, 2011, pp. 551-560.

[11] M. Lanza, S. Ducasse, H. Gall, and M. Pinzger, "CodeCrawler: an information visualization tool for program comprehension," in Proceedings of the 27th international conference on Software engineering, St. Louis, MO, USA, 2005, pp. 672-673.

[12] B. Cornelissen, A. Zaidman, A. van Deursen, L. Moonen, and R. Koschke, "A Systematic Survey of Program Comprehension through Dynamic Analysis," Software Engineering, IEEE Transactions on, vol. 35, no. 5, 2009, pp. 684-702.

[13] S. Bassil and R. K. Keller, "Software visualization tools: survey and analysis," in IWPC 2001. Proceedings. 9th International Workshop on, Toronto, Ont. Canada, 2001, pp. 7-17.

[14] A. R. Teyseyre and M. R. Campo, "An Overview of 3D Software Visualization," IEEE Transactions on Visualization and Computer Graphics, vol. 15, no. 1, Feb. 2009, pp. 87-105.

[15] T. D. Cook and D. T. Campbell, Quasi-experimentation: design & analysis issues for field settings, 3rd ed. Rand McNally College Pub. Co., 1979.

[16] P. Breretona, B. A. Kitchenham, D. Budgenb, M. Turnera, and M. Khalilc, "Lessons from applying the systematic literature review process within the software engineering domain," Journal of Systems and Software, vol. 80, no. 4, Apr. 2007, pp. 571-583.

[17] U. Erra, G. Scanniello, and N. Capece, "Visualizing the Evolution of Software Systems Using the Forest Metaphor.," in Information Visualisation (IV), 2012 16th International Conference on, Montpellier , 2012, pp. 87-92.

[18] M. Risi, G. Scanniello, and G. Tortora, "MetricAttitude: a visualization tool for the reverse engineering of object oriented software," in AVI '12, New York, NY, USA, 2012, pp. 449-456.

[19] INF/UFG. (2013, Oct.) Systematic Mapping Files. [Online]. http://www.inf.ufg.br/~auri/icsea-mv/