A Magnet-and-Spring Based Visualization Technique for Enhancing the Manipulation of Requirements

Ghazi, Parisa

DOI: https://doi.org/10.1109/RE.2015.7320459

Originally published at:
Ghazi, Parisa (2015). A Magnet-and-Spring Based Visualization Technique for Enhancing the Manipulation of Requirements. In: Doctoral Symposium of the 23rd IEEE International Requirements Engineering Conference (RE'15), Ottawa, Canada, 24 August 2015 - 28 August 2015. IEEE, 400-405.
DOI: https://doi.org/10.1109/RE.2015.7320459

A Magnet-and-Spring Based Visualization Technique for Enhancing the Manipulation of Requirements Artifacts

Parisa Ghazi
Department of Informatics
University of Zurich
Binzmühlestrasse 14, 8050 Zurich, Switzerland
http://www.ifi.uzh.ch/rerg/people/ghazi.html
ghazi@ifi.uzh.ch

***Abstract*—Requirements engineers model the system of interest from different points of view by creating numerous artifacts. Although they have to deal with a great amount of information, the display space of the devices is limited. This limitation leads to a time consuming navigation through the artifacts. Requirements engineers have to scroll through numerous pages and switch between multiple windows. However, they have to rely on their memory when there is no space left on the screen to view another piece of relevant information. In this research, we propose to develop a novel visualization technique that flexibly creates editable views of a linked set of elements or artifacts where the pieces show different levels of detail according to the user's demand for the current task. Thus, important parts are shown in detail, while the space taken for displaying unimportant parts is minimized.**

Our conceptual solution is a combination of the focus+context concept and a magnet-and-spring system. The focus+context concept is responsible for resizing and relocating objects to make space for more relevant information. The magnet-and-spring system is responsible for distributing the distortion caused by the focus+context concept throughout the workspace, such that the distorted view of the information looks more natural. Considering the artifacts of a software development project as a single hypothetical artifact enables us to manage the artifacts in the same way we deal with the objects inside an artifact. Our envisaged tool support should be embeddable in requirements applications and bring its benefits to the applications manipulating requirements artifacts.

***Index Terms*—requirements engineering, requirements artifacts, focus+context, visualization, navigation, physical metaphor, magnet, spring**

I. Motivation and Research Problem

Requirements engineers develop numerous artifacts to model their system of interest [1]. The complexity of the systems they deal with is the main reason for producing different models. They try to overcome the complexity by representing a system from different points of view and converting their understanding of the system into a large number of artifacts. Each artifact contains a set of features of the original system. The convenience of manipulating these artifacts is highly significant since the artifacts are created once, but are reviewed, compared and improved recurrently. These activities are not just performed in the requirements engineering phase, but almost throughout the entire software development project by designers, developers, testers and other stakeholders [2].

Today's display devices come in various sizes, from small displays on mobile devices to poster-sized screens or electronic whiteboards. However, they all share a common feature: the available space is limited [3]. Also, the larger a display device, the less handy and mobile it is and the more it costs. Consequently, the amount of information that can be displayed at a given moment is limited.

In comparison to the large range of senses requirements engineers use to look into problems in the real world, display screens are narrow windows through which they explore and manipulate artifacts that are stored in computerized systems.

To compensate for the narrow information bandwidth imposed by the limited size of display screens, appropriate visualization techniques are crucially needed for letting requirements engineers perform their tasks efficiently. Visualization becomes particularly important and challenging when (i) users need more than one piece of information at the same time, (ii) the artifact which contains the information they need is larger than the screen and (iii) users want to have their mental model of the information layout on the screen preserved when the visualization changes, e.g., for displaying some elements in more detail.

The goal of our work is to broaden the information exchange channel between requirements engineers and the stored artifacts that they are working on, not by just using larger displays, but by smart visualization solutions that use the available space efficiently. As none of the existing visualization techniques is fully satisfactory with respect to items (i)-(iii) above, we are exploring a new technique which is based on a physical metaphor of magnets and springs for creating an editable representation of requirements artifacts to display information in multiple levels of detail.

II. State of the Art

There are four main techniques that support users' navigation in information spaces: zooming, overview+detail, focus+context and cue based techniques [4].

Overview+detail and zooming techniques are available in the mainstream diagram modeling tools used in software engineering [5]. In the overview+detail approach, the overview of the whole diagram is shown in a small window while a fraction of that diagram is shown in a larger window with more detail. The overview window usually indicates which part of the diagram is visible in the detail view. The users can interact with the overview window to navigate within the whole diagram.

Focus+context is a general concept which has been employed in various fields of visualization and has been implemented in different ways based on the context of the usage [6], [7], [8]. In this approach the focused area is shown in more detail than other areas [9]. In the graphical fisheye view approach, this concept is applied to graph visualization [10]. In that method the focused node receives a larger share of the screen space than other nodes. In the technique proposed for visualizing ADORA models [11], users can enlarge any part of the diagram individually. The space needed to fit in the enlarged object is provided by pushing other objects away from the enlarged object. In this technique, zooming in multiple objects may result in a large canvas again.

Cue-based techniques support navigation in large information spaces by providing additional information to the user's view such as highlighting certain information to deal with a dense information view, or displaying indicators of existing off-screen objects usually on the edge of the view [5]. These methods provide hints about what is not currently on the screen but these still do not suffice for the users: e.g., the overview is lacking.

In addition to these four categories, *semantic zooming* is a technique that can be combined with zoomable interfaces [12]. In this technique the representation of an object changes in different zoom levels. Semantic zooming in hierarchical structures means transforming an entity into its underlying entities. In a graphical environment, semantic zooming substitutes an object with its constituents, which results in a higher level of detail [13]. Many approaches combined semantic zooming with the focus+context concept such as onion graphs [8] where a novel method limited to visualizing UML class diagrams is proposed.

One way of having a more efficient representation of information is showing not more detail than needed [14]. How much detail is needed or in how many levels the detail should be distributed are the early decisions that requirements engineers have to make about their model [15]. They live with these decisions throughout the project. The focus+context approaches make it possible to have different levels of detail in a single snapshot of the information, hence providing the users with a visualization where every element is presented exactly in the level of detail that they need. They will be able to study an artifact while having a glimpse of where that artifact is located among the interconnected artifacts.

The information can be provided in different levels of detail automatically using the metadata hidden in it [13]. An example of such metadata is when the user groups some objects or binds a text to an object. These metadata can be used for abstraction. Abstract forms of the objects can be used instead of the objects in the representation when the original object is too small to contain details.

Although adjusting the level of detail according to users' demand results in efficiency, it produces an adverse effect known as distortion [16]. Any structure altered in one or more aspect non-uniformly is distorted. Therefore on the way to our goal, distortion is inevitable. However, understandability of the distorted representation of information depends on the method that caused the distortion.

Physical metaphors aid virtual environments to look natural to the users [17]. Graph visualization techniques have used metaphors such as magnet, spring and force to make their results understandable and consequently acceptable. Force directed graph visualization is a group of approaches that arrange the nodes of the graph in an aesthetically pleasing way by lowering the number of crossing edges and keeping the edges equal in length. Their mechanism is modeling the nodes and edges as magnetic elements and simulating the movement of these elements under influence of the forces they apply to each other [18]. One of these approaches named spring model [19] uses a metaphor of both magnets and springs to model nodes and edges of graphs respectively. Then this approach visualizes the graphs aesthetically by exposing the models to magnetic fields.

III. Research Goal

Our goal is to design, develop and evaluate a technique to alter the traditional way of visualizing information embedded in RE artifacts. We don't restrict our technique to specific notations. The artifacts can conform to a modeling language (e.g., UML) or multiple languages or even their structure may be user-defined such as in FlexiSketch [20]. We intend to create a view specifically for the needs of the user at a given point of time. This view includes the required information and excludes what is closely related to the required information but is not actually needed at that point of time. The created view should not distort the information in a way that the user is no longer able to recognize it and map it to his/her own mental image of the information. In addition, the created view should be editable. The users should be able to create and modify their artifacts regardless of the distortion caused. Moreover, a convenient interface is required to prevent imposing an excess effort to the user for customizing the view of the information.

Our tool should support the requirements engineers in the following situations which requirements engineers encounter using current methods:

Viewing a large artifact. The user loses context when he/she zooms in a part of an artifact to view more details. The user has to rely on his/her memory about the context or switch between different zoom levels repeatedly.

Viewing pieces of information from different artifacts. The user has to switch between different pages (windows) in order to view information from different artifacts in the following cases: comparing artifacts, comprehending supplementary artifacts or creating a new artifact based on multiple other artifacts.

Managing a large number of artifacts. The requirements artifacts are related to each other like a connected network. Some artifacts are based on other artifacts, some artifacts are supplementary and some artifacts extend the concept of other artifacts. Displaying the artifacts in the form of a file list makes managing and searching cumbersome.

IV. Research Questions

Based on our research goal, we derived the following research questions (RQ).

RQ 1.1: How much do the existing visualization concepts support displaying different artifacts such as diagrams and documents with the amount of detail that user needs at the moment?

RQ 1.2: Which features of the visualization of requirements artifacts (e.g. size, position or amount of detail of the constituents) can be changed to create a more informative view of the information?

It is essential to our work that we first identify which existing concepts can be inspiring to develop our solution. Concepts such as focus+context will be investigated in this phase. By studying the beneficial concepts we can identify the parameters of the representation of the information that can be the subject of manipulation in order to fulfill the goals mentioned. Many researchers have already proposed methods that adjust parameters such as size, position and amount of detail to serve their goal of visualization. By alternatively changing these, parts of the information considered irrelevant at the moment can be hidden, thus giving space to highly demanded information. Many researchers have customized these parameters to find a mechanism for visualizing artifacts in a new way appropriate for their intended application.

We have already answered these questions when developing our conceptual solution [21].

RQ 2: To what extent can the conventional way of presenting artifacts be changed without causing disorientation in users or making the artifacts unrecognizable?

It should be emphasized that we are proposing the idea of modifying the information visualization method which users are widely familiar with. There is a risk that the users do not accept the new presentation of information.

At any moment different distortions fit the users' demands. Therefore the overall image of the artifacts changes over time. But the users have their own constant mental image of the artifacts which is an undistorted version. Hence, at an early stage, we will investigate what is the limit of altering the appearance of the information so that the users can still map their mental image of the information to the modified image on the screen. We are going to investigate whether requirements engineers are still able to find their way through the artifacts using their own mental image of the information and also how convenient it is to comprehend, create or modify the artifacts displayed in the new fashion.

RQ 3: How does the magnet-and-spring approach affect the performance of the requirements engineers and enables them to fulfill their tasks efficiently?

We are going to change the conventional way of presenting information which, according to the Section I, is used widely in software engineering tools. The main benefit of this change is the better performance of the requirements engineers. Therefore we need to measure the success of this approach. We will measure how efficiently requirements engineers perform when utilizing this new technique. Firstly, we have to define how the performance of requirements engineers can be measured and also the criteria of assessing them. Then, we will evaluate our technique according to these criteria while the requirements engineers perform their regular tasks.

RQ 4: How does the magnet-and-spring approach affect the existing solutions to other requirements engineering problems such as change management and traceability?

Topics such as traceability and change management have been the focus of many researchers. Although our approach is not a direct solution to these problems, it may enhance their solutions since it supports the management of artifacts and it makes use of the interconnection metadata of the artifacts.

V. Conceptual Solution

In this section, we introduce our conceptual solution named "FlexiView". Our approach is based on the focus+context concept. FlexiView partitions the working space into regions. In our approach, unlike other related approaches, the regions are the subjects of manipulation instead of the objects. In other words, FlexiView resizes and relocates regions according to the user's demand. The purpose of these changes is providing more space for the objects which the user is interested in. In order to distribute the responsibility of providing free space between regions, we propose to model the aforementioned regions with springs, as depicted in Figure 1.a. Changing the size or location of the regions will be possible by applying forces to the springs. A structure made of springs resists change and propagates the forces applied to the neighboring springs. In this way the distortion is distributed gradually and naturally.

We add semantic zooming to our approach to support a high range of sizes of regions. The size of the objects depends on the size of their respective regions and the level of detail depends on the size of the objects. Therefore, the objects of the artifacts can be resized without having to worry about making them too small and unreadable.

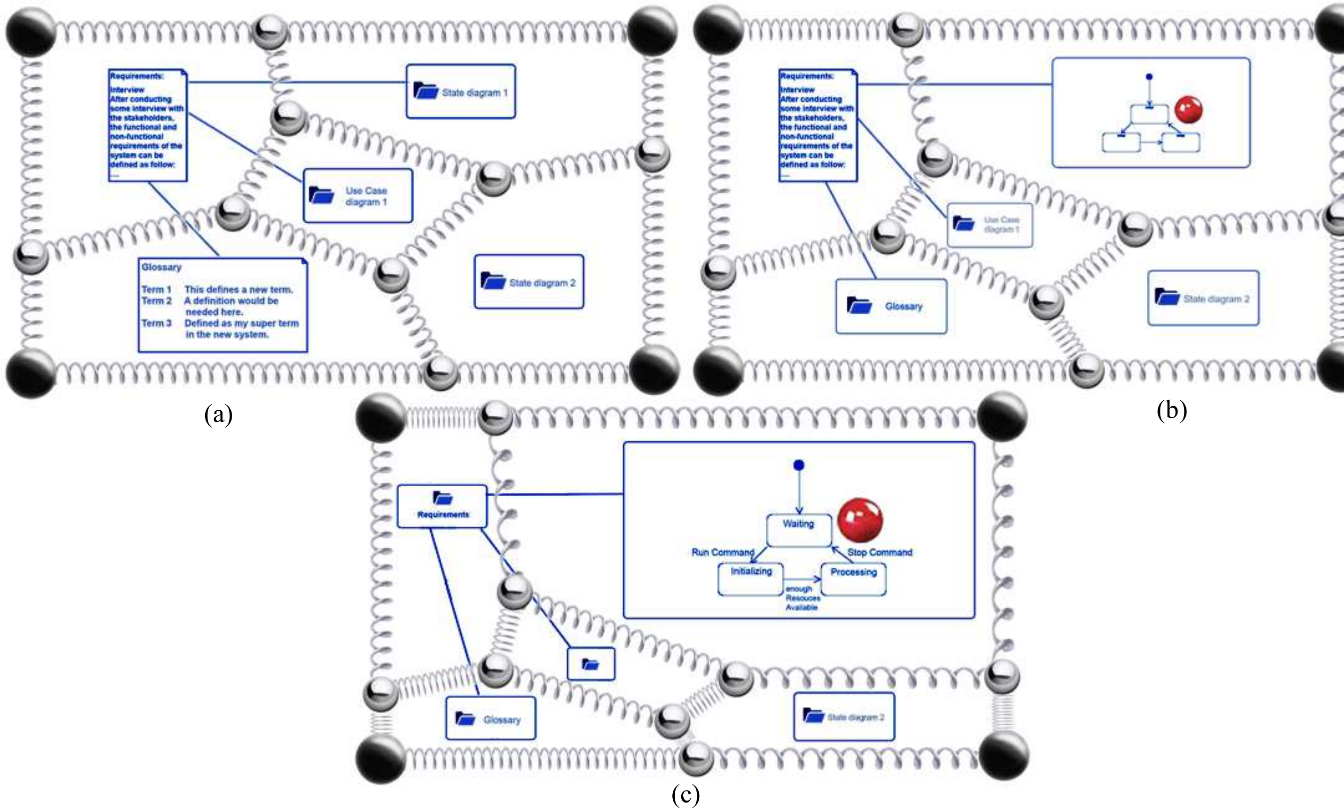

Fig. 1. (a) How FlexiView partitions the workspace and models it with magnetic balls (silver balls) and springs as the connectors. (b), (c) How the creation of a virtual magnet affects the position and the size of the regions [21]. Note that the balls and springs in this figure are shown for illustration purposes only. They are not visible in the views created by FlexiView.

Creating and modifying artifacts on a distorted canvas is nearly impossible. As mentioned in Section III, our proposed solution is not just for displaying information, but also for providing an environment for creating and modifying the requirements artifacts. To serve this goal, we maintain a uniform detail level inside each region. Therefore, each region is editable individually. We call this kind of canvas *partially distorted*.

We need to define a method of user interaction. We require a mechanism that allows users to indicate their regions of interest. Then the interaction of the user should be transformed into forces and the forces should be applied to the springs. In our conceptual solution, there is a magnetic ball at the intersection of boundaries of the regions. Users create virtual magnets in the regions that contain objects they would like to analyze. The virtual magnets repel the magnetic balls on the boundaries of the regions, resulting in the enlargement of the regions containing the interesting objects. The forces applied to the magnetic balls are transferred to the springs. The network of connected springs propagates these forces. Figure 1.b shows the effect of a virtual magnet that has been placed in the top right region. Consequently, this region has been enlarged while other regions have become smaller. Figure 1.c shows the effect of intensifying the strength of the virtual magnet. The objects are zoomed not only geometrically but also semantically, resulting in the emergence or disappearance of details.

More details about how the workspace is partitioned, how the regions are modeled with springs and magnetic balls and how the virtual magnet affects the user's view are provided in [21].

Finally, we need to manage multiple artifacts. In the hierarchy of detail levels, artifacts themselves are children of the highest node "project". If we suppose that the "project" is an artifact itself and its children are the elements inside this artifact, then the users can manage the artifacts in the same way that they deal with the elements of an artifact.

VI. Research Methodology

The first step of our research was classifying the needs and requirements of engineers in requirements engineering tools. Based on the requirements found, we started a *systematic literature review* about alternative visualization techniques. Each kind of information has its own specifications and a specific visualization technique suits it. We were looking for a visualization technique suitable for requirements artifacts. The literature review convinced us that there is no visualization technique tailored to requirements artifacts that satisfies the requirements we have identified in the first step. As described in the Section II, focus+context techniques enable us to have different detail levels on the screen simultaneously. Considering that detail level and scale level are closely related, we reached the basis of our *conceptual solution* and answered RQ1. Focus+context techniques cause distortion by their nature. So in the second step we continued the literature review searching for how to employ a focus+context method in a way that the distortion caused looks natural. In other words, users should be

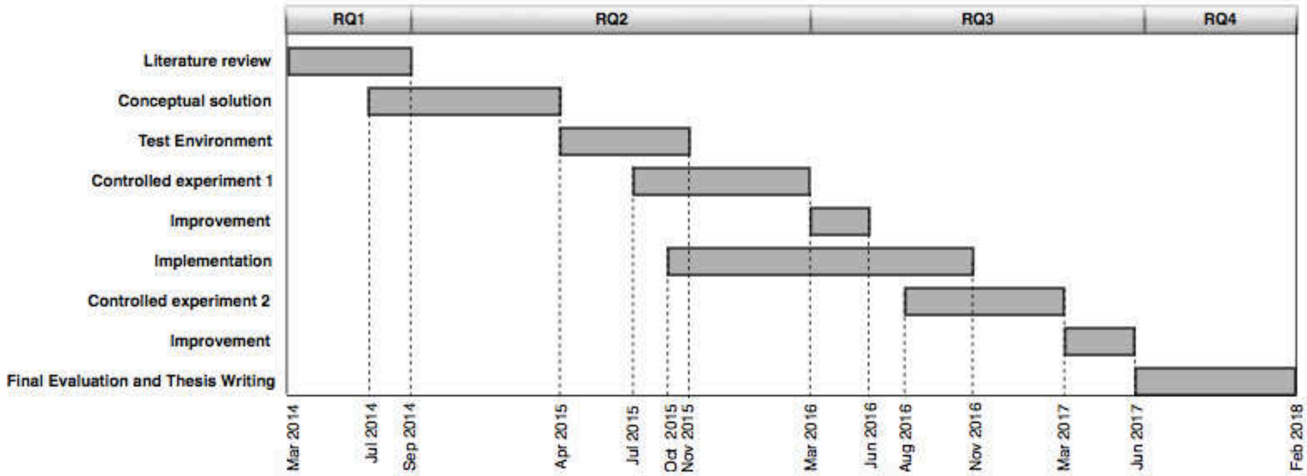

Fig. 2. The plan of this research project. The vertical axis shows the stages of the project. The horizontal axis represents the time. The bar at the top indicates where our research questions are being investigated.

able to follow the changing process of the view in order to be able to match the distorted view with the original undistorted information. We discovered that the spring model has been used in graph visualization, and the usability of the spring model is already proved [18]. Users are familiar with the outcome of their interaction with the spring model since it is a simulation of a real world phenomenon. Therefore, we expanded our conceptual solution to use springs and magnets. This partially addresses RQ2.

Currently we are building a *test environment* to evaluate our conceptual solution at a preliminary stage. In this environment, users can interact with a graph. The nodes of the graph are hypothetical artifacts. It is crucial to know how employing a magnet-based approach affects the ability of users in matching the altered view of their artifacts to the overview they have in their minds. We are going to find out whether the combination of the focus+context and spring models guarantees that users accept the created view as an overview of their artifacts. The envisaged test environment demands implementing a graph manipulation algorithm based on our definition of physics. We are exploring the existing algorithms to form the foundation of our algorithm.

The next step is to validate our answer to RQ2 using the described test environment. For this purpose, a *controlled experiment* will be conducted. The users will be assigned similar tasks and the following two attributes will be measured during their performances: (1) how precisely they can recognize a distorted version of given graphs, and (2) how fast they can distort a given graph to a desired state. The results will be compared with similar tasks in a conventional environment featuring zooming and panning. We may add other attributes to this list when we are designing the details of the experiment.

Figure 2 shows the progress of our project. After controlled experiment 1, we will *improve* our conceptual solution and tool support according to the results. Toward answering RQ3, we will *implement* our technique in a real modeling application. We intend to use FlexiSketch [20] as our platform for this purpose. Then, we will conduct our *controlled experiment 2* to evaluate our solution. First we have to define the common challenging operations that requirements engineers perform while working with RE artifacts. Information gathering techniques such as interviews or ethnography will be a part of this phase. The result is expected to be operations such as juxtaposing two artifacts or creating an artifact based on another one. Based on these operations, we will design different tasks. Test users will be asked to perform the tasks in two ways: (a) using a traditional navigation interface featuring scrolling and zooming, and (b) using our implemented solution. The performance of the users will be measured in terms of speed, precision and quality.

Although Figure 2 shows the phases of the project sequentially, the two experiments and improvement phases may be carried out iteratively depending on how much modification is imposed by the improvement phase. In case of high amount of modification, another experiment is needed which may be followed by an improvement phase.

The *final evaluation* provides the answer to RQ4. In this stage, the application of FlexiView will be investigated in other areas of requirements engineering. This phase of our project has the potential to lay the basis for further research.

VII. Progress

Having identified the requirements for the envisaged requirements artifact visualization tool, we defined the conceptual solution for FlexiView, described in Section V. At present, we are simultaneously creating a test environment to evaluate our conceptual solution during an early stage and working on the existing algorithms to form the foundation of our envisaged tool's algorithm. We will then continue with answering RQ2 and RQ3 in an iterative fashion (implementation and evaluation), and plan to evaluate and complete the research project in 2018.

We have a first publication [21] where we answer RQ1 and present our conceptual solution.

VIII. Conclusion and Contribution

In this paper, we proposed a method to gather the information that a requirements engineer needs for his/her current task from multiple artifacts and unify them in a single view.

The generated view features heterogeneous levels of abstraction in accordance with users' demand. The impact of this approach on requirements engineering is enabling the requirements engineers to accomplish their tasks with shorter scrolling distance, fewer windows open and memorizing less, and consequently increasing their performance and precision. We foresee that our tool support will make the artifacts more comprehensible and will accelerate the knowledge transfer. Moreover, when the final implementation is completed, we will search for more applications of this approach in specific domains of requirements engineering such as traceability and change management.

In addition to the tool support, our contribution will be the algorithms that serve the implementation of our tool, the result of our first experiment showing how much the distortion of focus+context algorithms is apprehensible, and the result of our second experiment showing to what extent a visualization technique can influence the process of requirements engineering.

REFERENCES

[1] M. Petre, "Insights from expert software design practice," in Proceedings of the 7th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2009), pp. 233-242, 2009.

[2] B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," Communications of the ACM, vol. 31, pp. 1268–1287, 1988.

[3] N. Mangano, T. D. LaToza, M. Petre, and A. van der Hoek, "Supporting informal design with interactive whiteboards," in Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '14), pp. 331-340, 2014.

[4] A. Cockburn, A. Karlson, and B. B. Bederson, "A review of overview+detail, zooming, and focus+context interfaces," ACM Computing Surveys, vol. 41, pp. 1–31, 2008.

[5] M. Frisch and R. Dachselt, "Off-screen visualization techniques for class diagrams," in Proceedings of the 5th International Symposium on Software Visualization (SOFTVIS '10), pp. 163-172, 2010.

[6] G. W. Furnas, "Generalized fisheye views," ACM SIGCHI Bulletin, vol. 17, pp. 16–23, 1986.

[7] T. Reinhard, S. Meier, R. Stoiber, C. Cramer, and M. Glinz, "Tool support for the navigation in graphical models," in Proceedings of the 30th International Conference on Software Engineering (ICSE '08), pp. 823-826, 2008.

[8] H. Kagdi and J. I. Maletic, "Onion Graphs for Focus+Context Views of UML Class Diagrams," in Proceeding of the 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis, pp. 80-87, 2007.

[9] I. Herman, G. Melancon, and M. S. Marshall, "Graph visualization and navigation in Information visualization: A survey," IEEE Transactions on Visualization and Computer Graphics, vol. 6, pp. 24–43, 2000.

[10] M. Sarkar and M. H. Brown, "Graphical fisheye views of graphs," in Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '92), pp. 83-91, 1992.

[11] M. Glinz, S. Berner, and S. Joos, "Object-oriented modeling with ADORA," Information Systems, vol. 27, pp. 425–444, 2002.

[12] B. B. Bederson and J. D. Hollan, "Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics," in Proceedings of the 7th annual ACM symposium on User Interface Software and Technology (UIST '94), pp.17-26, 1994.

[13] M. Frisch, R. Dachselt, and T. Brückmann, "Towards seamless semantic zooming techniques for UML diagrams," in Proceedings of the 4th ACM symposium on Software Visuallization (SoftVis '08), pp. 207-208, 2008.

[14] G. W. Furnas and B. B. Bederson, "Space-scale diagrams: Understanding Multiscale Interfaces," in Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '95), pp. 234-241, 1995.

[15] H. Fuhrmann and R. von Hanxleden, "Taming graphical modeling," In Proceedings of the 13th international conference on Model Driven Engineering Languages and Systems (MODELS 2010), LNCS vol. 6394, pp. 196-210, 2010.

[16] Y. K. Leung and M. D. Apperley, "A review and taxonomy of distortion-oriented presentation techniques," ACM Transactions on Computer-Human Interaction, vol. 1, pp. 126–160, 1994.

[17] J. M. Rzeszotarski and A. Kittur, "Kinetica: Naturalistic multi-touch data visualization," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14), pp. 897-906, 2014.

[18] A. S. Spritzer and Carla M. D. S. Freitas, "A physics-based approach for interactive manipulation of graph visualizations," in Proceedings of the working conference on Advanced Visual Interfaces (AVI '08), pp. 271-278, 2008.

[19] P. Eades, "A heuristic for graph drawing," Congresses Numerantium, UtilitasMathematica Publishing Inc., Winnipeg, vol. 42, pp.149–160, 1984.

[20] D. Wüest, N. Seyff, and M. Glinz, "FlexiSketch: a mobile sketching tool for software modeling," in Proceeding of the 4th MobiCASE Conference, vol. 110, pp. 225–244, 2013.

[21] P. Ghazi, N. Seyff, and M. Glinz, "FlexiView: a magnet-based approach for visualizing requirements artifacts," in Requirements Engineering: Foundation for Software Quality, Lecture notes in Computer Science, S. A. Fricker and K. Schneider, Eds. Springer International Publishing Switzerland, vol. 9013, pp. 262-269, 2015.