On the Use of Context Information for Supporting Software Visualizations

Renan Vasconcelos, Marcelo Schots, Cláudia Werner

Programa de Engenharia de Sistemas e Computação (PESC) – COPPE/UFRJ
Caixa Postal 68.511 – 21945-970 – Rio de Janeiro, RJ – Brasil
{renanrv,schots,werner}@cos.ufrj.br

Abstract. *Software visualization helps to identify issues and speed up the indications of solution. The efficiency of visualization interpretation varies according to the elements that compose the representations. The choice of such elements must comply with the context that comprises their use, possibly requiring or promoting visualization changes. This work analyzes some information that is part of this context, highlighting the importance and influence of each one in a visualization. Examples of use of this context information are presented in order to illustrate the influence on the choice of visualization techniques in different scenarios.*

1. Introduction

Interpreting the vast amount of information available becomes complicated in environments with varied types of information and tasks. Visualization may represent a suitable alternative in this type of situation, since abstractions of data can reveal patterns, clusters, gaps, or outliers [Shneiderman 1996]. However, some aspects that are closely related to how the graphical representation is displayed must be taken into account, such as the task at hand, the interactions performed by users, and the characteristics of data.

The purpose of using a given visualization is usually related to some *task* that will be fulfilled or aided by it. Indeed, tasks relate to visualizations' design and evaluation [Schulz et al. 2013]. The former combines data and tasks in order to seek the most appropriate representation. Visualization evaluation, in turn, combines data and visualization in order to investigate how appropriate is the visual result in supporting a task. Thus, tasks can be considered as input for visualizations, representing external context information concerning the visualization and the data, seeking to make a graphical representation meaningful.

Because of the intrinsic relation with the visualization field, *interaction* can also be considered one of visualization's main components. Although separated from the graphical result, these components are strongly related, since interactions may lead to changes in representations [Yi et al. 2007].

The process of information visualization is based on the data transformed into images that are more easily interpretable by humans [Spence 2000]. Thus, it is possible to note the importance of data for such domain. The data to be represented by a visualization should be a vital aspect to be considered. Accordingly, the viewed data can also be analyzed as contextual information.

This article aims at exploring the use of context information in order to support information visualization, indicating its use for software visualization as well. Particularly, three types of information are discussed: data, interaction and task. In order to illustrate these key elements on the visualization context, some motivating examples are also presented. The remaining of the paper is organized as follows: Section 2 introduces the concept of context awareness; the elements that may influence the visualization context are discussed in Section 3; Section 4 presents a practical example of use of context information in visualizations; related works are listed in Section 5; and the final remarks are presented in Section 6.

2. Context-Aware Applications

Context-aware applications are characterized by capturing information from the environment, including users and their behavior, in order to provide customized answers according to each context state [Chen & Kotz 2000]. In order to derive actions from the current state, this context must be well-defined. In this sense, context models help to standardize the representation of information.

A context model should support different levels of elements, such as: (i) entities, related to the context dimensions; (ii) context information, showing how each item collaborates to the description of a particular entity; (iii) context situations, which are composed by entities and their information with defined values; and (iv) rules, which allow associating actions to be performed on the occurrence of a given situation. A notation that follows this structure is UbiFEX [Fernandes et al. 2011].

With a defined notation, it is necessary to evaluate the aspects that should be included in the model, i.e., to examine the appropriateness and relevance of the information to compose the desired context model. In the context of this work, some information that is relevant for visualizations is discussed. This relevance is interpreted in terms of how modifications in the contextual information can lead to or require changes in visualizations.

3. The Role of Context Information in Visualizations

The most important aspect of using visualizations is that some meaning must be transmitted, regardless of the medium where they are displayed or the chosen data source. The featured representations should enable users to interpret the results and to obtain useful information. In this sense, besides the values of the selected data, techniques aimed at expressing the visualization context are also required to provide an interpretive meaning through data representation. It is important to relate data values with the phenomenon that they represent [Keller & Keller 1993]. With such characteristic, visualizations can make their users aware of different situations.

Some aspects can be considered as *contextual information* in order to define the state of a view, given their relevance. In other words, changes in these information values may result in adaptations in the view. These aspects are discussed as follows.

3.1. Interactions

Interactions allow a direct communication between users and systems [Dix et al. 2004]. When a system is represented through visualizations, interactions have an explicit relationship with the change on a view, since interaction techniques may enable different actions on this view. The variety of existing interaction elements is demonstrated in [Vasconcelos et al. 2014], where each interaction technique is identified as an information visualization domain feature.

In order to identify what changes might occur in a view based on the applied interactions, it is necessary to check the user intentions on performing an action in a view. In this sense, Yi et al. (2007) present seven categories of interaction that indicate possible changes in visualizations, namely: Select, Explore, Reconfigure, Encode, Abstract/Elaborate, Filter and Connect.

It is worth noting that the context extraction involves registering past user interactions. The user interaction on a view denotes his/her behavior on the representation of a data set. This behavior can be considered a useful aspect for context analysis, since it allows performing changes in the visualization representation.

Interactions may reveal user preferences that can be useful to a recommended visualization change. For instance, a view is set to represent the most searched keywords in a software repository. Each element displayed in a view has its representation based on the number of each keyword's occurrences, so that the size of an element is proportional to the number of searches for that keyword. If only keywords with a number of occurrences above a certain threshold are selected, meaning that the user only selects the larger elements, the view can be updated to display just those keywords' elements. In this case, *removing* such non-interesting records would be more suitable. Appropriate context information for this case is summed up as the *minimum occurrence of selected items*.

3.2. Data

The importance of data in visualizations is recognized by the number of information visualization taxonomies that depend on the data type [Müller & Schumann 2003] [Chi & Riedl 1998]. Thus, it is necessary to assess how changes in the data may impact in visualizations. With this goal, it is important to check the data types and the transformations that occur in existing visualizations in the literature.

The study of Chi (2000) uses the *Data State Model* [Chi & Riedl 1998] to evaluate views with different techniques. This evaluation includes (i) a survey of data values, (ii) data transformations for composing an analytical abstraction), (iii) visualization transformations, turning an analytical abstraction into a visual abstraction, and (iv) visual mapping transformations, generating a graphical representation.

A survey of various visualizations allows observing similarities between techniques and assessing the necessary steps for representing the data. In this sense, it is worth noting that changes in data values should promote a number of specific transformations for each visualization technique until the graphical result.

It is worth emphasizing that often the change in the data set cannot be predicted. In this sense, event-based visualizations [Müller & Schumann 2003] must have special precaution mechanisms, treating possible restrictions on the application of certain visualization techniques, such as interpolation values.

Since changes in the data set usually influence the graphical result, an example can be seen in the selection of a very large data group. When one needs to represent many visual elements (e.g., in the case of the searched keywords on a software repository), behaviors that can be adopted in a view are: (i) the adjustment of scale, providing more space for a larger number of simultaneous records, and (ii) filtering mechanisms, in order to show only the (most) relevant items [Shneiderman 1996]. It is also important to check the data type in order to evaluate how the information can be displayed in a view.

Examples of appropriate context information to address these constraints are the *number of records* and the *data type*. For instance, when a certain number of records can impact negatively on the representation understanding, one can apply appropriate visualization features for updating the scale and filtering, such as *geometric zooming* and *removal*. This will help to avoid the overload of elements to be displayed on a small screen.

3.3. Tasks

In addition to the user interactions and data, the view focus can be assigned by the task to be supported by the visualization. In this case, the so-called visualization tasks allow for analyses in order to obtain a set of recurring tasks whose accumulated knowledge would optimize the visualization design and evaluation [Schulz et al. 2013].

An appropriate visualization for a task must consider elements that characterize such task. Schulz et al. (2013) identify dimensions that describe a task relating to its context, aiming to extract information about the user, such as the moment when a task is performed, the ordering in a sequence of tasks, and who will carry it out. Aiming at defining a set of views that would be important for a domain, it is necessary to define which tasks cover the main issues that should be represented.

For example, an important task in the software development domain concerns the possibility of reusing existing reusable assets (development *with* reuse), and some information can be visually represented for providing awareness on this possibility [Schots 2014], such as the number of reusable assets available in the organization for the current project domain. The number of finished projects in the domain of a current project also indicates if such domain has a significant number of projects, showing if there should be more reusable assets available. The number of searches in the reuse repository by the same keyword without results helps verifying if a particular feature is commonly expected/requested (and possibly present in several projects) in the organization. This would indicate interest in the existence of a reusable asset that meets this demand. A group of related context information can be comprised by the *number of available assets to the current domain*, the *number of legacy projects in the current domain* and the *number of keywords searches in the repository without results*.

3.4. Context Rules

From the aspects considered relevant for the context, which shall be treated as context information, it is necessary to directly relate the use of certain visualization techniques to the momentary state of the context. Context rules are responsible for this association.

With a defined context situation, through a set of information with certain assigned values, it is possible to define the configuration of a view. The occurrence of a specific context implies the domain feature selection [Fernandes et al. 2011] – in this case, the selection of visualization elements. Figure 1 displays the composition logic of a context rule using the UbiFEX notation [Fernandes et al. 2011].

<RULE> ::= <CONTEXT-SITUATION> ***implies*** <FEATURES>

Figure 1. Context rule model

The information visualization domain features and the constraints between them (defined by composition rules) are described in [Vasconcelos et al. 2014]. These rules take into account a set of identified recommendations for the application of visualization techniques listed in [Vasconcelos et al. 2013], among others.

4. Example of Use

Aiming at recognizing how context information can influence visualizations in practice, we present a unified example comprising the aforementioned aspects. A context rule is proposed to illustrate the use of context for the selection of visualization elements.

4.1. Context Situation

In order to map a context to the selection of visualization features, it is necessary to define context situations, which may provide a wider reach for the visualizations. Because such visualizations are intended to be context-aware, the occurrence of a situation should bring, through changes in a view, the necessary attention to a task.

With the occurrence of some values for the context information, the task of identifying the necessity of reusable assets may rise as the focus for the final view. A related context situation for this purpose is presented in Figure 2. It is noteworthy that the thresholds of each kind of information should be customizable by the organizations.

Develop a new reusable asset
Number of records > 30
Data type = String
Number of available assets to the current domain < 10
Number of legacy projects in the current domain > 5
Number of keywords searches in the repository without results > 7

Figure 2. Context situation

Another context situation can encompass changes that can be promoted through interactions in a view. Since the user behavior upon such view should be recognized as context information, an appropriate context situation holds the same information of the previous situation and incorporates the interaction information, as shown in Figure 3.

Filter reusable asset development candidates
Minimum occurrence of selected items > 20
Number of records > 30
Data type = String
Number of available assets to the current domain < 10
Number of legacy projects in the current domain > 5
Number of keywords searches in the repository without results > 7

Figure 3. Context situation with an interaction behavior

4.2. Context-Aware Visualization

After the appropriate context situations have been defined, the context rules will map the features that a visualization should have in order to better support the user carrying out the task of identifying the necessity of new reusable assets.

A visualization that matches the task goal can organize the keywords (searched in the repository without results) in a bubble structure (pack layout), representing each item as an overview. With a large data set (e.g., comprising more than 30 items), a geometric zooming mechanism can be helpful to adjust the scale. Colors can be used to highlight different domains, and sizes can represent the number/frequency of searches. Finally, the items' names and metadata can be shown as tooltips within each bubble, after a simple selection. A context rule that corresponds to these observations is illustrated in Figure 4. A visualization that matches this rule is shown in Figure 5.

Context Rule 1 (CR_1)
Develop a new reusable asset ***implies*** (Pack Layout AND Geometric Zooming AND Highlighting AND Tooltip AND Overview AND Selection)

Figure 4. Context rule for the necessity of reusable asset

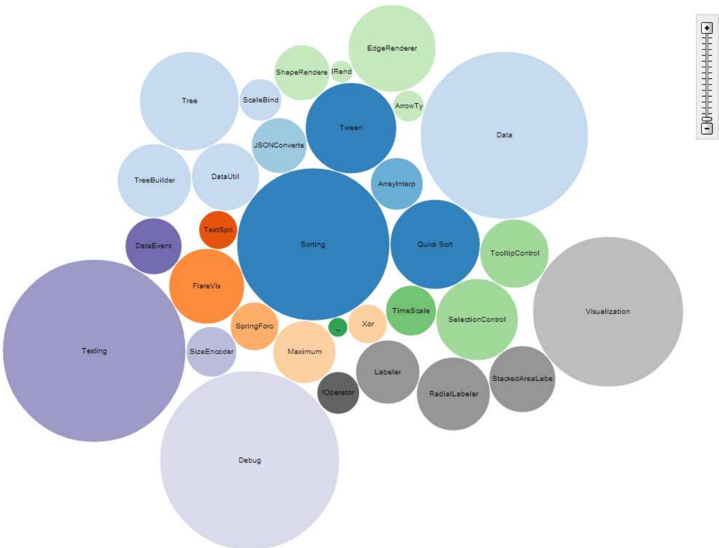

Figure 5. Visualization of search terms without corresponding assets

Another rule can be proposed to comprise the interaction aspect, in order to remove items with less than 20 occurrences based on the user selection. Such context rule is presented in Figure 6.

Context Rule 2 (CR_2)
Filter reusable asset development candidates ***implies*** (Pack Layout AND Geometric Zooming AND Highlighting AND Tooltip AND Overview AND Selection AND Removal)

Figure 6. Context rule for updating view for the necessity of new reusable asset

5. Related Work

Some existing approaches in the literature also discuss the role of data, interactions and tasks in selecting the most appropriate visualizations. Beck et al. (2013) propose a methodology for choosing a graph-based visualization according to application profiles.

The application characterization is accomplished through a classification of data and the tasks involved for their representation. This study supports experts during the selection of visualization techniques for a given application.

In a task-focused approach related to visualizations, Schulz et al. (2013) present a visualization tasks design space, with shared taxonomy and models. This study allows the evaluation of the suitability of a task to a specific case, and its selection to compose views for end users. The design space assists in the orientation and differentiation between tasks, having as output the recommendation of views to meet a selected task.

Also seeking to organize a taxonomy, Chi (2000) examines characteristics of visualization techniques at different levels. By checking the similarities and differences between techniques in various data domains, it was possible to check requirements of interest for a proper application of each technique.

From the presented studies, it is worth noting the focus on elements related to tasks and data for the use of visualization techniques. Although the work of Beck et al. (2013) considers data and tasks, it gears its analysis to graph-based visualization layouts. The other works focus on proposing taxonomies for common visualization applications, considering aspects of data [Chi 2000] and tasks [Schulz et al. 2013]. The current study aims at analyzing the context in a broader way, considering data, interactions and tasks.

6. Final Remarks

The proper use of visualizations, with a careful choice of techniques, can contribute to a more efficient interpretation of results. Such proper use can accelerate the decision making and allow carrying out different activities with a higher level of awareness. In scenarios with a complex information flow, such as software development, visualizations that meet the needs of different tasks can ease the information processing.

In this sense, an analysis of the most relevant aspects of a context can provide indications about what should be graphically represented. Also, in order to treat the whole data as a context, the data values should be periodically checked, so that a visualization can be adapted/updated or completely changed.

The context information analyzed in this study integrates to the APPRAiSER approach [Schots 2014] through CAVE, a tool under development that seeks to provide context-aware visualizations for the software reuse scenario. Data, interactions and reuse tasks will serve as input in order to recognize the context and select the appropriate visualization elements for the current state. An experiment will be carried out for evaluating the pros and cons that context-driven visualization may provide.

References

Beck, F., Burch, M., & Diehl, S. (2013). Matching application requirements with dynamic graph visualization profiles. In *17th International Conference on Information Visualisation*, London, UK, pp. 11-18.

Chen, G., Kotz, D. (2000). A Survey of Context-Aware Mobile Computing Research. Dartmouth College Technical Report TR2000-381.

Chi, E. H. H., & Riedl, J. T. (1998). An operator interaction framework for visualization systems. In *IEEE Symposium on Information Visualization*, Durham, USA, pp. 63-70.

Chi, E. H. (2000). A taxonomy of visualization techniques using the data state reference model. In *IEEE Symposium on Information Visualization (InfoVis)*, Salt Lake City, USA, pp. 69-75.

Dix, A., Finlay, J., Abowd, G. D. and Beale, R. (2004). Human-computer interaction, 3rd ed: Pearson Prentice Hall.

Fernandes, P., Teixeira, E. N., Werner, C. (2011). An Approach for Feature Modeling of Context-Aware Software Product Line. J. Univers. Comput. Sci., v. 17, n. 5, pp. 807-829.

Keller, P. R., & Keller, M. M. (1993). Visual cues: practical data visualization (p. 6). Los Alamitos, CA: IEEE Computer Society Press.

Müller, W., & Schumann, H. (2003). Visualization methods for time-dependent data-an overview. In *Proceedings of the 35th Winter Simulation Conference*, New Orleans, USA, Vol. 1, pp. 737-745.

Schots, M. (2014). On the Use of Visualization for Supporting Software Reuse. Ph.D. Qualifying Exame, Federal University of Rio de Janeiro, Brazil.

Schulz, H. J., Nocke, T., Heitzler, M., & Schumann, H. (2013). A design space of visualization tasks. In *IEEE Trans. Visual. Comput. Graphics*, 19(12), 2366-2375.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *IEEE Conference on Visual Languages*, Columbus, USA, pp. 336-343.

Spence, B. (2000). Information Visualization. Pearson Education Higher Education Publishers.

Vasconcelos, R. R., Schots, M., & Werner, C. (2013). Recommendations for Context-Aware Visualizations in Software Development. In *10th Workshop on Modern Software Maintenance*, Salvador, Brazil, pp. 41-48.

Vasconcelos, R., Schots, M., & Werner, C. (2014). An information visualization feature model for supporting the selection of software visualizations. In *22nd International Conference on Program Comprehension, Early Research Achievements track,* Hyderabad, India, pp. 122-125.

Yi, J. S., ah Kang, Y., Stasko, J. T., & Jacko, J. A. (2007). Toward a deeper understanding of the role of interaction in information visualization. In *IEEE Trans. Visual. Comput. Graphics*, 13(6), 1224-1231.