Knowledge-oriented Refactoring in Visualization

Dong Hyun Jeong, Wenwen Dou, William Ribarsky, and Remco Chang

Abstract—In the past, many visualization applications are designed by visualization experts. Although most of them are useful and well designed to address people's needs in visualization, a refactoring process is often considered to enhance their visualization applications. However, the refactoring process in visualization is different from the process used in software-engineering. In this paper, we explain the refactoring process in visualization. Based on understanding the process in visualization, we propose a new approach (knowledge-oriented refactoring) that focuses on adopting user's personalized knowledge when redesigning visualization applications.

1 Introduction

Many successful visualization applications in recent years have been developed by visualization experts collaborating with domain experts. With a well-designed visualization application, users are able to understand and analyze complex datasets, and as the need of the end users changes or increases in using the visualization tools, the visualization designers would modify and update the system accordingly. In cases when the foundations of the visualization are inadequate in supporting the users' needs, the visualization would need to be redesigned and re-implemented all together. Typically, such a redesigning process emphasizes supporting additional features (visualization techniques), speed-up calculations (performance improvements), new hardware systems (graphics cards), etc.

In software-engineering literature, however, researchers have studied restructuring and refactoring the existing software for more than a decade [7]. The restructuring is defined as "the transformation from one representation form to another at the same relative abstraction level, while preserving the subject system's external behavior (functionality and semantics) [3]" and the refactoring is regarded as the object-oriented variant of restructuring. Specifically, the refactoring focuses on altering not the external behavior, but the internal behavior of the code [4]. The main idea of the refactoring is to redistribute classes, variables, and methods to facilitate future adaptations and extensions [7].

Although both the restructuring and refactoring are important and should be considered, these processes have not been major research areas in the visualization community. Instead, researchers in visualization typically focus on redesigning existing visualization applications in an efficient and useful way. In this paper, we propose a knowledge-oriented refactoring process that performs the refactoring process based on adopting users' personalized knowledge or domain-specific knowledge.

In recent, several researchers consider adopting knowledge in visualization. Chen et al. [2] propose knowledge-assisted visualization based on the differentiation of data, information, and knowledge in visualization, in which both existing and users' personalized knowledge are used to design a visualization application. While their system has been shown to be novel and effective, it is not clear how such a design paradigm could be broadened to the refactoring process in visualization In this paper, we extend the spirit of incorporating knowledge into visualizations and propose such a knowledge-oriented approach to refactoring visualization systems. We propose that with our refactoring method, existing visualization application(s) could be extended in useful ways.

- *Dong Hyun Jeong, Wenwen Dou, William Ribarsky, and Remco Chang are with Charlotte Visualization Center at UNC Charlotte, E-Mail:{dhjeong, wdou1, ribarsky, rchang}@uncc.edu.*

2 Refactoring Process in Visualization

Designing a visualization application is similar to the software-engineering process. However, the goals of two research domains are distinctive in that visualization designs primarily focus on the external structure (visual representation) of the application, and software-engineering concentrates on the internal structure of the software (the structure of the source code). Because of this distinction, the refactoring process in visualization needs to be considered differently. In this section, we propose the features that should be supported when redesigning a visualization application.

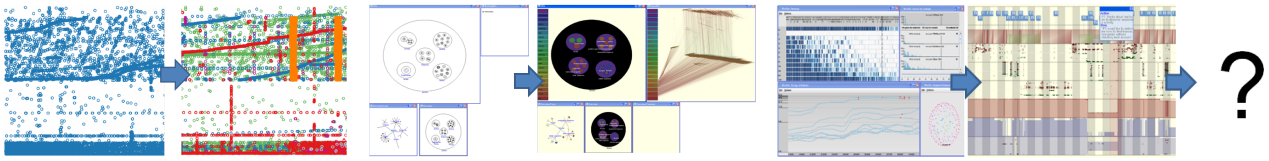
Fig. 1. A diagram that represents visualization pipelines.

In visualization, creating a visual imagery is a complicated process. Figure 1 shows this process as a diagram that represents the visualization pipelines. Once data are entered into the visualization pipeline, algorithms are applied to the data in order to either (or both) filter out unnecessary information or extract important and useful information. With the processed data, visualization techniques are then applied to create visual representations as images.

In most cases, when a visualization application is redesigned, the changes take place in the "visualization techniques" and the "visual representation" parts of the pipeline (Figure 1) to support additional features, remove unnecessary features, improve visual representations, etc. The refactoring process therefore often involves adopting previously designed visualization techniques and visual representations. This refactoring process works well in improving visualization applications, but it is potentially limiting because the refactoring is performed without a clear understanding of the "data" or the "filtering and extracting" steps. In visualization, most visualization applications are well designed by visualization experts, but understanding the data and the domain knowledge necessary for the appropriate filtering and extracting algorithms are often beyond the expertise of the visualization designer. Collaboration with domain experts alleviates some of these shortcomings, but often the domain experts' knowledge cannot easily be transferred and directly integrated into the visualizations (sometimes referred to as the "communication gap" [8]). Therefore, the refactoring process in visualization is only limited to support either additional visualization techniques or redesigning visualization representations. We propose that with a more conscious focus on the other two steps in the visualization pipeline during a redesign, the resulting visualization will be enhanced not just visually, but also in actual use for solving domain specific tasks. In next section, we explain our approach, the knowledge-oriented refactoring process.

3 Knowledge-oriented Refactoring in Visualization

In general, the refactoring process in visualization is only considered when a visualization application has a limitation to represent data effi-



(a) (b) (c)

Fig. 2. Examples of knowledge-oriented refactoring in visualization. (a) A network traffic analysis system [10] is enhanced by adopting users' discoveries. In here, a color representation technique is used to represent each network traffic pattern such as mail (green), DNS (blue), scan (red), etc. (b) A genomic visualization application (GVis) [5] is extended based on adopting existing biological classification that mapped with each individual color. (c) From a financial visualization application (WireVis [1]), users' analyzing processes (e.g. semantic-level interactions) are captured and used to redesign the existing visualization application.

ciently. But the knowledge-oriented refactoring process is applicable to not just imperfectly designed visualization applications, but also well formed visualization applications. Some researchers consider applying knowledge into visualization applications to enhance their visual representations. Xiao et al. [10], for instance, present a network traffic analysis system that reuses users' knowledge (discoveries - the meanings of each network traffic pattern) to change the visual representation in a meaningful way (Figure 2(a)). With limited knowledge about each network traffic pattern, the visually represented network traffic patterns do not have enough information explaining what each pattern means. If the application is modified by adopting users' knowledge, detecting and understanding the meanings of each network traffic pattern become a trivial but meaningful task. Also our genomic visualization application (called GVis [5]) shows how the existing visualization application can be redesigned to incorporate domain knowledge and enhance the understanding of complex genomic data and finding useful information (Figure 2(b)). The initial version of the application has been designed to represent genomic data and to address biological analysis processes. It has been extended with known biological knowledge structures such as general biological classification and biological taxonomy structure from NCBI (National Center for Biotechnology Information) to enhance understanding biological categorization.

Adopting knowledge into visualizations is important and useful for finding important information. However, identifying relevant and appropriate knowledge that could be applied to a visualization application is difficult. One possible source of knowledge could be extracted from experts' analysis. For example, work by Dou et al. have shown that analysts' analytical processes, strategies, methods, and findings, can be recovered through only the examination of analysts' interaction log [9]. In the study, 10 analysts' interactions of analyzing financial fraud detection tasks were captured in a financial visual analytics application (called WireVis [1]) and analyzed by four coders (Figure 2(c)). The studied showed that up to 80% of the findings could be recovered from the interaction logs through the use of specifically designed visual analytical tools [6]. This finding suggests that users' interaction logs (e.g. semantic-level interactions such as keywords, accounts, and transactions) embed users' reasoning processes. The knowledge-oriented refactoring process therefore may include the users' reasoning processes to enhance understanding and analyzing data through a visualization application that considers and incorporates users' interactions. The financial visualization (WireVis) is viewed as a well designed application that preserves the analytical procedures in the financial data. However, with a limited understanding about the financial fraud analysis, such a useful visualization cannot be efficiently used to solve complex analytical tasks. In such a complicated visualization application, the knowledge-oriented refactoring should be performed by adopting experts' reasoning processes.

The knowledge-oriented refactoring process can be applied to visualization applications in two possible ways: integrating knowledge into the applications or completely redesign the applications using a knowledge-oriented design. Visualization applications such as the network traffic analysis system and the genomic visualization are redesigned by integrating knowledge into the existing visualization applications. In this case, the overall visual representation does not need to be greatly altered. Instead, existing visualization techniques are simply adopted to emphasize the importance within the visualizations. For the financial visualization, we are in the process of redesigning the application based on users' reasoning processes. The two knowledge-oriented refactoring processes need to be carefully considered because the redesigning process is as difficult as building a new one in complicated visualization applications.

Of course, the knowledge-oriented refactoring looks somewhat similar to knowledge-assisted visualization (KaV) [2]. However, the main focus of the knowledge-oriented refactoring is to redesign any existing visualization applications by adopting experts' reasoning processes.

4 Conclusion and Future Work

The question "what is the best visual representation?" is typically difficult to answer because visualization applications are designed by adopting different visualization techniques depending on the types of the datasets. However, the final goal for many visualization applications, in particular visual analytical systems, should be the usefulness of the system to perform analysis. In these types of systems, a goal-oriented refactoring process should be considered as an important step when upgrading or refining the visualization. In this paper, we propose a new approach, knowledge-oriented refactoring, which focuses on redesigning existing visualization applications by adopting domain specific knowledge. Our proposed approach is based on the examination of the visualization pipeline, and we show that visualizations that are redesigned or refactored with integration of knowledge in mind, the resulting systems demonstrate significant improvement.

References

[1] R. Chang et al. Wirevis: Visualization of categorical, time-varying data from financial transactions. In *VAST 2007. IEEE Symposium on*, pages 155–162, 30 2007-Nov. 1 2007.

[2] M. Chen et al. Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl.*, 29(1):12–19, 2009.

[3] E. J. Chikofsky and J. H. Cross II. Reverse engineering and design recovery: A taxonomy. *IEEE Softw.*, 7(1):13–17, 1990.

[4] M. Fowler. *Refactoring: Improving the Design of Existing Programs*. Addison-Wesley, 1999.

[5] J. Hong et al. Gvis: A scalable visualization framework for genomic data. In *EuroVis 2005*, pages 191–198. Eurographics Association, 2005.

[6] D. H. Jeong et al. Evaluating the relationship between user interaction and financial visual analysis. In *VAST '08. IEEE Symposium on*, pages 83–90, Oct. 2008.

[7] T. Mens and T. Tourwe. A survey of software refactoring. *IEEE Trans. Softw. Eng.*, 30(2):126–139, February 2004.

[8] A. J. Pretorius and J. J. van Wijk. Bridging the semantic gap: Visualizing transition graphs with user-defined diagrams. *IEEE Comput. Graph. Appl.*, 27(5):58–66, 2007.

[9] D. Wenwen et al. Recovering reasoning processes from user interactions. *IEEE Comput. Graph. Appl.*, 29(3):52–61, 2009.

[10] L. Xiao, J. Gerth, and P. Hanrahan. Enhancing visual analysis of network traffic using a knowledge representation. In *VAST 2006 IEEE Symposium On*, pages 107–114, 31 2006-Nov. 2 2006.