Requirements Visualization

Alfredo Teyseyre, Ricardo Orosco and Marcelo Campo

Univ. Nacional del Centro Prov. Bs. As. - Fac. Ciencias Exactas
ISISTAN - Grupo de Objetos y Visualizaci6n,
Paraje Arroyo Seco - Campus Universitario, (7000) Tandil, Bs. As., Argentina

E-Mail: {teyseyre, orosco, mcampo} @exa.unicen.edu.ar

1 Introduction

Meeting user requirements of a software system has been a major challenge to software developers. Experience in a number of large projects reveals that a very large percentage of errors were made at the earlier stages of their development. Therefore, it is a well-accepted fact that it is crucial to express user requirements completely, correctly and unambiguously as possible.

Many approaches have been applied to validate requirements, but most of them seem fit better to software developers than customers. In spite of their benefits, they often fail in the user validation process since they are based on formal notations not always comprehensible by users. So, in order to facilitate the comprehension by users, visualization techniques appear as an interesting alternative.

Visualization techniques are a powerful tool to facilitate the analysis and understanding of complex information. In spite of the their successful in numerous computing areas, little research has been reported in the area of requirements visualization.

In this work, we describe our current efforts towards the application of visualization techniques to achieve a more effective requirements validation process.

This paper is organized as follows. Section 2 explains visualization concepts and its benefits in the requirements validation domain. Section 3 describes the current project and identifies the different tasks involved in its development.

2 Requirements Visualization

Lets first state the notion of visualization, which is defined by Card [Card et al., 1998] as follows: ***"the use of computer-supported, interactive, visual representations of data to amplify cognition"***, where cognition is the acquisition or use of knowledge.

Although few works on requirements visualization have been reported (e. g. VIZ [Ozcan et al., 1998]), visualization techniques can be applied to requirements. The use of visualization techniques could reduce the communication gap between the customer and developer resulting in a more effective requirements validation process [Parry et al., 1998]. Visualization is a powerful tool to facilitate the analysis and understanding of complex information such as requirements. In general, graphical representations provide a closer match to the mental model of the users than textual representations and take advantage of their perception capabilities. For example, think about a lift control system: lift movements between floors, openings and closings doors, illumination of indicator lights, and user's requests for travel. Visualization could help the developer to analyze the correction of the specifications of such complex process and to validate them with the customer.

At the beginning most of the visualization systems display 2D graphics, but nowadays, more and more applications use 3D graphics in their visual presentations. Using this kind of presentations provides several advantages. The first and, perhaps the most clear one, is a greater information density than two-dimensional presentations as a consequence of a bigger physical space [Robertson et al., 1993]. Also, help to have a clear perception of the relations between objects by integration of local with global views [Mackinlay et al., 1991] and composition of multiples 2D views in only one 3D view [Koike, 1993]. However, the central advantage is their similitude with the real world that enables us to represent it in a more natural way. This means that the representation of the objects can be done according to its associated real concept, the interactions can be more powerful (ranging from immersive navigation to different manipulation techniques), and the animations can be even more real. This feature has a major incidence in the requirements visualization: ***it enables us to visualize the requirements closely to its real partner.*** For example, let's think again about the lift control system. An additional dimension enables us to present a view closer to reality, that also can be shown from different points of view. Moreover, a system composed by several lifts can be visualized in only one view and without loosing the context of the building that contains them. This representation gives a clear and comprehensive picture to the users in order to validate them. In this example and in other domains, the mapping from the requirements to a representation in the virtual world is direct, but depending on the nature of the information, finding this mapping could not be so direct as it was.

On the other hand, several drawbacks arise: intensive computation and more complex implementation than two-dimensional interfaces. This problems can be solved using powerful and specialized hardware and several tools like toolkits (IRIS INVENTOR [Strauss, 1993]), frameworks (GRAMS [Parris and William, 1992], GROOP [Larry and Wayne, 1993], and JAVA 3D [Sowizral et al., 1998]) , and modeling languages (VRML [ISO, 1997]).

However, one of the most serious problems is their ***usability*** which requires specific visualization techniques. The users often get lost in a virtual world due to a high level of freedom of the 3D interfaces. Moreover, interacting with a virtual world requires the adaptation of the user to non-conventional devices (head mounted displays and hand-gloves), that also produce weariness and discomfort.

Therefore, trying to take advantage of the benefits of 3D visualizations and to overcome their drawbacks, we have centered, one of the focus of our research, in improving the usability of these interfaces by developing ***useful and usable visualization techniques*** applicable to requirements. In order to do so, current visualization techniques could be adapted, or new entirely 3D techniques could be developed.

3 The Project

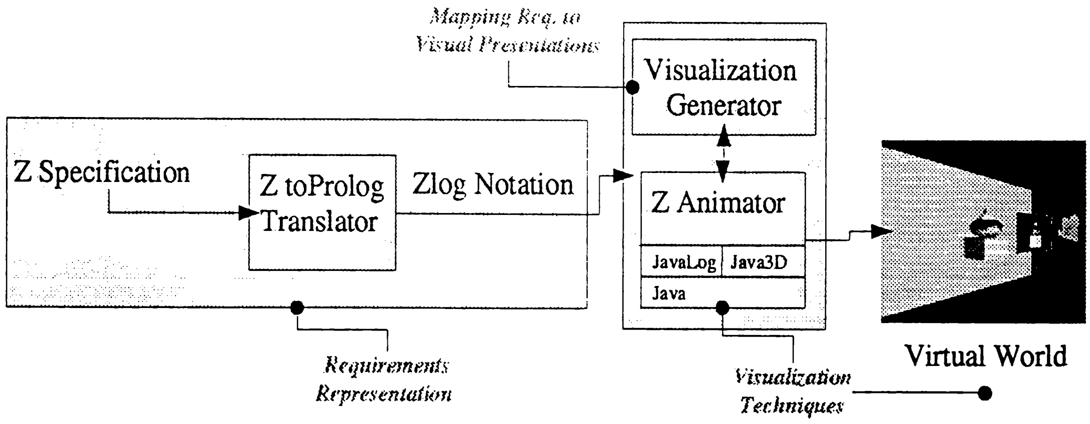
Our main objective is the visualization and animation of requirements in a virtual world for user validation. In order to do so, we must choose a requirement representation, find a mapping from the requirement representation to its visual presentation and study visualization techniques to represent requirements (figure 1 presents the system architecture). These problems are discussed in the following subsections.

3.1 Requirement Representation

We have decided to start the validation process from a formal specification. Formal specification languages have a formal syntax and semantics which makes it possible to unambiguously denote the meaning of the requirements. The best know formal specifications languages are Z [Hayes, 1987] and VDM [Jones, 1990], among others. In our case, we have chosen Z as our specification language.

However, in order to present an animation a Z specification is not enough, it must be translated to an executable version. A Z specification, can be translated and executed relatively easily to languages such as Prolog or Lisp [Ozcan et al., 1998][Traynor et al., 1997]. In our case we have chosen Prolog and we named the translated Z specification Zlog.

Figure 1 Architecture



3.2 Mapping Requirements to Visual Representations

In order to validate the requirements visually, a mapping from specification concepts to a suitable 3D representation of them must be done. This mapping could be done manually, but it would be preferably to make it by combination of user-assistance and semi-intelligent strategies. Several works have been reported about the automatic generation of presentations, but most of them deal with specific domains, such as statistical graphics, graphs, bar-charts, and scatter plots.

These systems can be classified according to its approach for the generation of the presentation: presentation generated from data description (APT [Mackinlay, 1986]), from task description (BOZ [Casner, 1991]), and mixed approaches (Descartes [Andrienko and Andrienko, 1998]). Although no report of similar tools in the area of virtual environments exist, similar approaches to generate the virtual environments can be taken.

3.3 Visualization Techniques

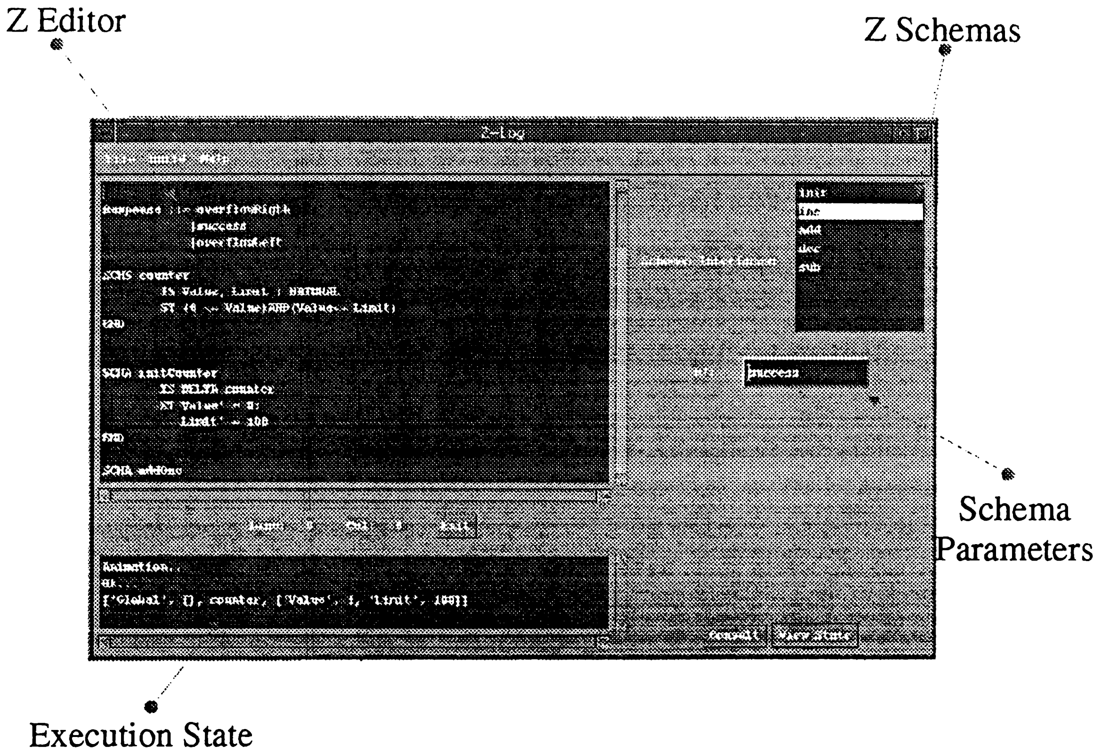
The goal of this stage is integrate and extract the advantages of current visualization techniques (VT). For doing so, we describe and identify features and advantages of current visualization techniques, that could be useful for the requirements visualization in a virtual world.

- Specialized VT based on the information structure: A visualization technique specific for each information structure allows us to choose the most suitable representation for each particular data, for example Xerox Parc [Robertson et al., 1993] studied the applicability of visualization techniques to information structured in different ways: lineal (Perspective Wall), hierarchical (Cone Tree), continuous data (Data Sculpture) and spatial data (Office Floor Plan).

- VT combining 2D views into one 3D view: several 2D diagrams are integrated into a single 3D view. Therefore only one presentation is given to users, so there is very little need for users to reconstruct their mental models. For example, Koike [Koike, 1993] and Wen [Wen, 1995] integrated 2D presentations of abstract data and software in a single 3D view.

- VT integrating different interaction techniques: usable interactions techniques are needed to access the information easily, for example navigation, semantic zoom, and panning, among others. But these techniques must be used according to the user experience, finding also an adequate degree of freedom in order to not disorient him. For instance, the system Galaxy of News [Rennison, 1994] embodies an approach to visualizing large quantities of independently authored pieces of news stories providing several interaction techniques adapted to this kind of information.

Figure 2 ZLog System



- VT adapting successful 2D presentations to 3D: much work has been done in developing tools for 2D spaces which allow for detail in context views. Much of this effort can be reused and applied to 3D visualizations taking advantage of the experience we have obtained, for instance, Carpendale [Carpendale et al., 1997] examined the extension of 2D distortion methods (fish-eye [Furnas, 1986]) to 3D and Robertson extended trees presentations to 3D (Cone Trees) [Robertson et al., 1991].

- VT integrating focus and context: integrating focus + context allows us to view in detail without loosing context. Several techniques have been done using a focus + context scheme: bifocal [Spence and Apperley, 1982], Perspective Wall [Mackinlay et al., 1991], DOCUMENT LENS [Robertson and Mackinlay, 1993], and MAGIC LENS [Bier et al., 1994].

3.4 State of the project

We have developed a compiler from Z to Prolog. This compiler generates a Prolog executable version of the Z specification. Also we have developed an environment to edit and execute specifications (figure 2). At the moment we are able to execute specifications for a subset of Z, and we are also studying the execution of object oriented extensions of Z .

The Project is been developed in JAVA. The integration between Java and Prolog is done using JAVALOG [Iturregui and Zunino, 1998]. JAVALOG allows us to mix JAVA and PROLOG, taking advantage of the benefits of each paradigm. In order to build 3D visualizations, we are using JAVA3D, a toolkit to develop 3D Graphics Applications developed by SUN MICROSYSTEMS.

References

[Andrienko and Andrienko, 1998] Andrienko, G. and Andrienko, N. V. (1998). Interactive maps for visual data exploration. GMD - German National Research Center for Computer Science.

[Bier et al., 1994] Bier, E. A., Stone, M. C., Pier, K., Fishkin, K., Baudel, T., Conway, M., Buxton, W., and DeRose, T. (1994). Toolglass and magic lenses: The see-through interface. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 2 of *VIDEOS: Part II – Techniques for Improved Human-Computer Interaction*, pages 445–446.

[Card et al., 1998] Card, S., MacKinlay, J., and Shneiderman, B., editors (1998). *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers.

[Carpendale et al., 1997] Carpendale, M. S. T., Cowperthwaite, D. J., and Fracchia, F. D. (1997). Extending distortion viewing from 2D to 3D. *IEEE Computer Graphics and Applications*, 17(4):42–51.

[Casner, 1991] Casner, S. M. (1991). A task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics*, 10(2):111–151.

[Furnas, 1986] Furnas, G. W. (1986). Generalized fisheye views. In Mantei, M. M. and Orbeton, P., editors, *Proceedings of the ACM Conference on Human Factors in Computer Systems*, SIGCHI Bulletin, pages 16–23. Association for Computer Machinery, New York, U.S.A.

[Hayes, 1987] Hayes, I., editor (1987). *Specification Case Studies.* International Series in Computer Science. Prentice-Hall, Inc.

[ISO, 1997] ISO (1997). Vrml97, international specification. Technical report, ISO.

[Iturregui and Zunino, 1998] Iturregui, R. and Zunino, A. (1998). *Una Arquitectura de Software Híbrida para Coordinación de Agentes Inteligentes.* UNICEN, Tandil, Buenos Aires, Argentina. Trabajo Final de la Carerra de Ingeniería en Sistemas.

[Jones, 1990] Jones, C. B. (1990). *Systematic Software Development Using VDM.* Prentice-Hall International, Englewood Cliffs, New Jersey, second edition. ISBN 0-13-880733-7.

[Koike, 1993] Koike, H. (1993). The role of another spatial dimension in software visualization. *ACM Transactions on Information Systems*, 11(3):266–286.

[Larry and Wayne, 1993] Larry, K. and Wayne, W. (1993). Groop: An object oriented toolkit for animated 3d graphics. In *OOPSLA'93*, pages 309–325.

[Mackinlay, 1986] Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141.

[Mackinlay et al., 1991] Mackinlay, J. D., Robertson, G. G., and Card, S. K. (1991). The perspective wall: Detail and context smoothly integrated. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, Information Visualization, pages 173–179.

[Ozcan et al., 1998] Ozcan, M. B., Parry, P. W., Morrey, I. C., and Siddiqi, J. I. (1998). Visualisation of executable formal specifications for user validation. *Lecture Notes in Computer Science*, 1385.

[Parris and William, 1992] Parris, E. and William, K. (1992). Application graphics modeling support through object orientation. *IEEE Computer*, (10):84–90.

[Parry et al., 1998] Parry, P. W., Ozcan, M. B., and Siddiqi, J. I. (1998). The application of visualization to requirements engineering. Technical report, Computing Research Centre, Sheffield Hallam University, England.

[Rennison, 1994] Rennison, E. (1994). Galaxy of news: An approach to visualizing and understanding expansive news landscapes. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Visualization I, pages 3–12.

[Robertson et al., 1993] Robertson, G., Card, S. K., and Mackinlay, J. D. (1993). Information visualization using 3D interactive animation. *Communications of the ACM*, 36(4):57–71.

[Robertson and Mackinlay, 1993] Robertson, G. G. and Mackinlay, J. D. (1993). The document lens. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Visualizing Information, pages 101–108.

[Robertson et al., 1991] Robertson, G. G., Mackinlay, J. D., and Card, S. K. (1991). Cone trees: Animated 3D visualizations of hierarchical information. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, Information Visualization, pages 189–194.

[Sowizral et al., 1998] Sowizral, H., Rushforth, K., and Deering, M. (1998). *The Java 3D API Specification.* Addison-Wesley.

[Spence and Apperley, 1982] Spence, R. and Apperley, M. (1982). Data base navigation: An office environment for the professional. *Behaviour and Information Technology*, 1(1):43–54.

[Strauss, 1993] Strauss, P. (1993). Iris inventor, a 3d graphics toolkit. In *OOPSLA '93*, pages 192–200.

[Traynor et al., 1997] Traynor, O., Hazel, D., and Strooper, P. (1997). Possum: An animator for the sum specification language. Technical Report 97-10, Software Verification Research Centre, Schhol of Information Technology, The University of Queensland.

[Wen, 1995] Wen, J. (1995). Exploiting orthogonality in three dimensional graphics for visualizing abstract data. Technical Report CS-95-20, Brown University - Department of Computer Science.