

3D Requirements Visualization

Alfredo Raúl Teyseyre

ISISTAN Research Institute, Facultad de Ciencias Exactas,
Universidad Nacional del Centro de la Pcia. de Bs. As.
Campus Universitario Paraje Arroyo Seco - (7000) Tandil - Bs. As., Argentine.
Phone/Fax: +54 - 2293 - 440362/3
email: teyseyre@exa.unicen.edu.ar

Abstract

The importance of correctly determining the requirements of a system at the very beginning of the development process it is a well known fact. Experience shows that the incorrect definition of the requirements leads to the development of deficient systems, increases the cost of its development or even cause projects to fail. Therefore it is crucial for the clients to verify that the planned system satisfies their needs. This means that the system must be described in a form that clients can clearly understand it. In this context, visualization techniques appear as a useful tool to help the users in the process of requirements understanding and validation.

This work proposes the use of 3D visualization techniques to validate the requirements of a system with the user. The use of these techniques can reduce the communication gap between the clients and the developers resulting in a much more effective process of requirements validation. The approach tries to take advantage of the benefits of the 3D visualization, complementing this with the advantages of formal specifications.

A tool, called ReqViz3D, that materializes the proposal was developed. This tool allows to specify the requirements in the formal language Z, define a graphical representation of them, and create a 3D animated visualization of theirs execution through which the users can validate them.

Index Terms: Requirements, Visualization, Requirements Visualization, 3D Graphics, Formal Specifications.

1 The Approach

Meeting user requirements of a software system is a major challenge to software developers. Experience in a number of large projects reveals that a very large percentage of errors were consequence of the imprecision in the earlier stages of the development process [11]. Therefore, it is a well-accepted

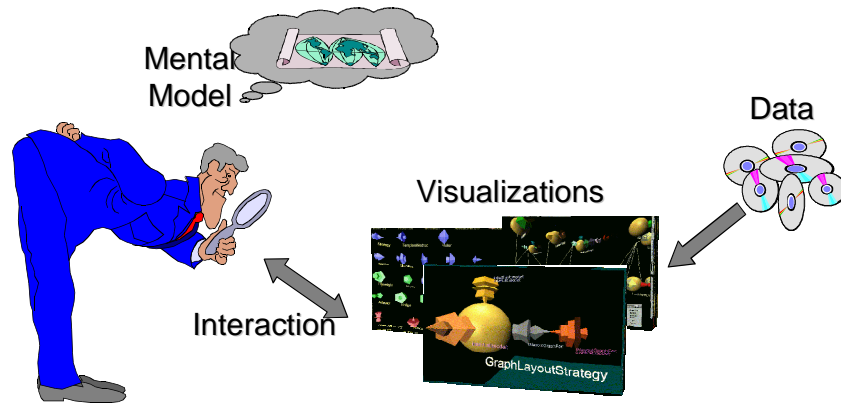


Figure 1: Visualization Process

fact that it is crucial to express user requirements as completely, correctly and unambiguously as possible. Moreover, it is vital for the customers to be able to confirm that the planned system meets their needs, and this means that the system must be described in a way that they can understand it [12].

Many conventional approaches have been applied to validate requirements, but, most of them, fail in detecting errors [6]. On the other hand, formal approaches, give clarity and precision at specification time. In that sense, formal specifications, enable us to denote unambiguously the meaning of a requirements specification document due to their formal syntax and semantics. However, except in safety-critical work, the cost of full verification is prohibitive [5]. Moreover, formal specifications often fail in the user validation process since they are based on formal notations not always comprehensible by users and hence they fit better to software developers than customers. Therefore, in order to overcome these difficulties visualization techniques appear as an interesting alternative to explore.

Visualization is a method to comprehend information by use of diagrams to represent it. Data are transformed into geometric representations that help users in the understanding process as figure 1 shows. 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.

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 [13]. Also, they help to have a clear perception of the relations between objects by integration of local with global views [8] and by composition of multiples 2D views in a single 3D view [7]. Moreover their similitude with the real world enables us to represent it in a more natural way than 2D. This means that the representation of the objects can be done according to its associated real concept, the interactions can be more powerful and the animations can be even more real.

In spite of their success in numerous computing areas, little research has been reported in the area of requirements visualization. The previous approaches enable developers to validate visually the specification of a system with the user, but their poor expressive graphics make difficult understanding [9, 4]. Moreover, neither of the works make use of current 3D graphics capabilities in order to present more real animations. However, 3D visualization techniques can be a powerful tool to

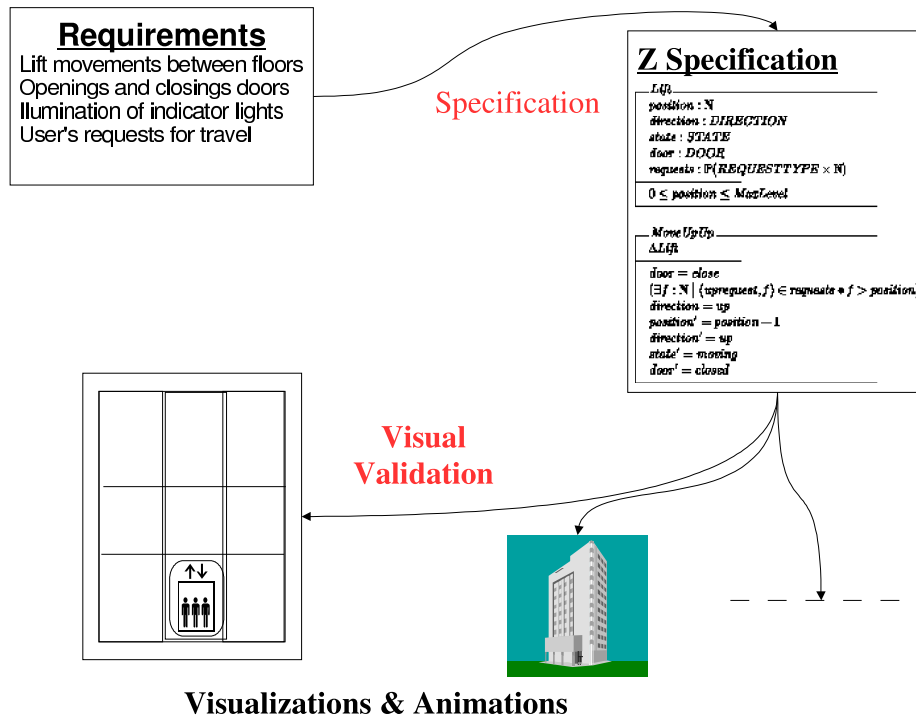


Figure 2: Requirements Validation Process

facilitate the analysis and understanding of requirements.

In this context, the main objective of this work is using 3D visualization and animation techniques to validate requirements with the user. This work proposes the use of 3D visualization techniques to validate the requirements of a system with the user. The use of these techniques could reduce the communication gap between the customer and developer resulting in a more effective requirements validation process [10]. The approach tries to take advantage of the benefits of the 3D visualization, complementing this with the advantages of formal specifications.

A tool, called REQ VIZ3D, that materializes the proposal was developed. This tool allows to specify the requirements in the formal language Z [15], define a graphical representation of them, and create a 3D animated visualization of their execution through which the users can validate them (figure 2 presents an informal model of the requirements validation process).

The main contribution of this work is the use of 3D visualization techniques to validate the requirements of a system and help, consequently, to avoid the propagation of errors in the requirements to the last stages of the development process, where its correction is much more expensive. In addition, the incorporation of the third dimension contributes to represent the requirements in a more natural way than using bidimensional graphics because of its similarity with the real world.

Also, a prototype tool was developed. This tool assist the developer in several stages in the development process: from requirements specification in Z and definition of graphical objects, to animation and execution of requirements in a 3D world. This prototype was developed based on architectural design driven by patterns. This process brings benefits that impact directly on understanding, reuse, evolution, analysis and documentation management of the system.

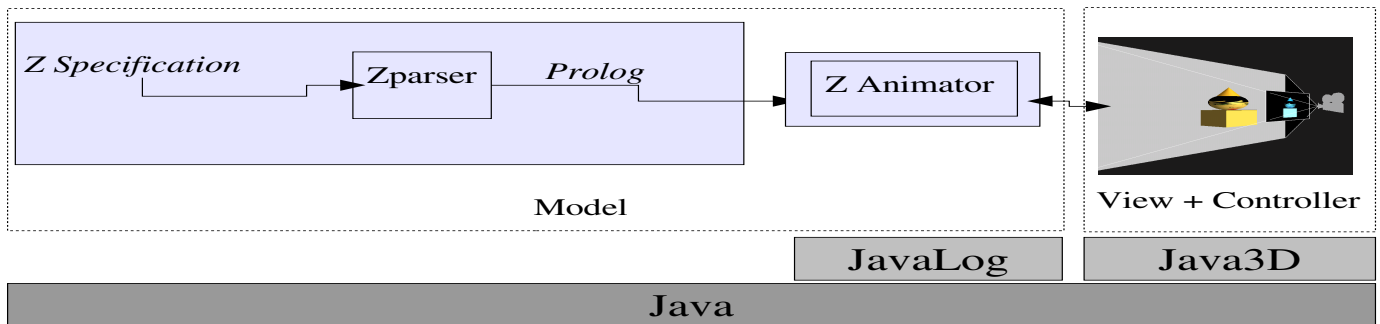


Figure 3: Global System View

2 The Tool

The figure 3 presents a global system view of REQVIZ3D that defines a blueprint of the overall structure of the application and corresponds to the architectural model *Model-View-Controller* [3]. This model prescribes the division of an interactive application in three parts, the *Model* that represents the application functionality, the *View* responsible for the output interface and the *Controller* responsible for the input handling.

In order to animate a Z specification it is translated to PROLOG so executed. A logic programming language is a very interesting choice for translating a specification language as Z which is based on first order logic. The conceptual gap between a logic programming language (which is a subset of a first order logic) and an imperative language is significantly less than a specification based on logic and an imperative language [2].

As we developed REQVIZ3D in JAVA, a way to integrate JAVA and PROLOG was needed. This integration was done using JAVALOG [1]. JAVALOG is a PROLOG interpreter written in JAVA designed to allow easy integration between JAVA and PROLOG mixing Logic/OO paradigms. Also, trying to take advantage of 3D visualizations we developed the View subsystem on the top of JAVA3D [14]. JAVA3D is an application programming interface for writing three-dimensional graphics applications and applets. Also, several examples were developed using this tool, as figure 4 shows: an automatic teller machine, a vending machine and a lift system.

References

- [1] A. Amandi, A. Zunino, and R. Iturregui. Multi-paradigm languages supporting multi-agent development. In *MAAMAW'99*, pages 128–139, 1999.
- [2] P. Breuer and J. Bowen. Towards Correct Executable Semantics for Z. In *Proc. 8th Z Users Workshop (ZUM)*, pages 185–212. Springer-Verlag, 1994.
- [3] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-Oriented Software Architecture: A System of Patterns*. John Wiley & Sons, 1996.
- [4] D. Hazel, P. Strooper, and O. Traynor. Possum: An animator for the SUM specification language. In *Proceedings: 4th Asia-Pacific Software Engineering*, pages 42–51. IEEE Press, 1997.

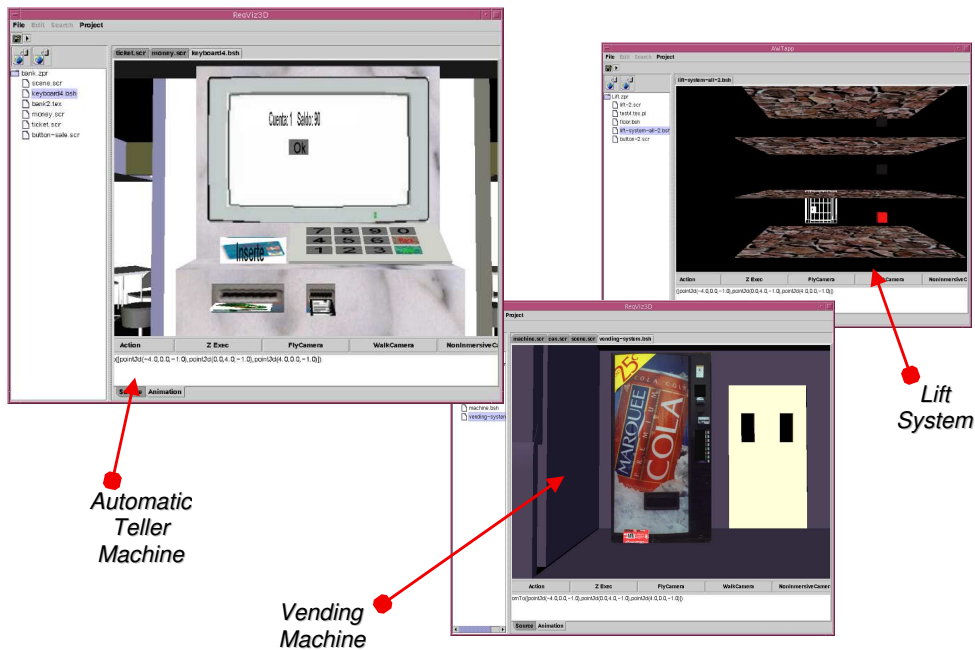


Figure 4: Examples

- [5] D. Jackson and J. Wing. Lightweight Formal Methods. *IEEE Computer*, 29(4):22–23, 1996.
- [6] John C. Kelly, Joseph S. Sherif, and Jonathan Hops. An analysis of defect densities found during software inspections. *The Journal of Systems and Software*, 17(2):111–117, February 1992.
- [7] Hideki Koike. The role of another spatial dimension in software visualization. *ACM Transactions on Information Systems*, 11(3):266–286, 1993.
- [8] J. D. Mackinlay, G. G. Robertson, and S.K. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of ACM CHI'91*, pages 173–179, 1991.
- [9] M. B. Ozcan, P. W. Parry, I. C. Morrey, and J. I. Siddiqi. Visualisation of executable formal specifications for user validation. *Lecture Notes in Computer Science*, 1385, 1998.
- [10] P. Parry, M. Ozcan, and J. Siddiqi. The application of visualization to requirements eng. Technical report, Computing Research Centre, Sheffield Hallam University, 1998.
- [11] Ben Potter, Jane Sinclair, and David Till. *An Introduction to Formal Specification and Z*. Prentice Hall, New York, 1991.
- [12] C. Potts. Expediency and appropriate technology: An agenda for requirements engineering research in the 1990s. *Lecture Notes in Computer Science*, 550, 1991.
- [13] George Robertson, Stuart K. Card, and Jock D. Mackinlay. Information visualization using 3D interactive animation. *Communications of the ACM*, 36(4):57–71, April 1993.
- [14] H. Sowizral, K. Rushforth, and M. Deering. *The Java 3D API Specification*. Addison-Wesley, 1998.
- [15] J. M. Spivey. *The Z Notation*. Prentice Hall International, UK, 2 edition, 1992.