Chapter 2.3
Conceptual Modeling of Virtual Environments Using Hypermedia Design Techniques

Paloma Díaz
Universidad Carlos III de Madrid, Spain

Susana Montero
Universidad Carlos III de Madrid, Spain

Ignacio Aedo
Universidad Carlos III de Madrid, Spain

Juan Manuel Dodero
Universidad Carlos III de Madrid, Spain

ABSTRACT

Traditionally, the development of virtual environments has been tightly dependent on the programmer's skills to manage the available toolkits and authoring systems. In such a scenario, the discussion of different design alternatives, future changes and maintenance, interoperability, and software reuse are all of them costly and quite difficult. In order to overcome this unsystematic and technology-driven process, conceptual modeling has to be included just before the implementation phase to provide a shared representation language that facilitates the communication among the different team members, including stakeholders, as well as the reuse and redesign for future requirements since conceptual models hide implementation details and constraints, and are cheaper and easier to produce than prototypes. As a first attempt to attain these aims, this chapter presents the basis of a constructional approach for the VE conceptual modeling through a set of complementary design views related to the VE components and functions. Moreover, we explore how these design issues might be addressed by hypermedia modeling techniques, given the simi-

larities between these two kinds of interactive systems and the maturity reached in hypermedia development.

INTRODUCTION

When facing the development of virtual environments (VEs), most developers turn to toolkits or authoring tools like those reported in Kessler, Bowman, and Hodges (2000) in which they pick different components out of a repository and build their environment from scratch in an unsystematic fashion driven by technology rather than by requirements. In such a scenario, abstraction is absolutely despised. Instead of describing the VE using concepts and relationships that describe the problem to be solved in terms of the universe of discourse (such as rooms, collections, or paintings in a virtual museum) as it is done in conceptual models (Hofstede & van der Weide, 1993), it is expressed using technical terms and implementation units, such as cylinders, spheres, or textures. But such a technology-driven development strategy brings a number of disadvantages. Firstly, development is boiled down to 3D modeling and programming, so that the stakeholders can only take part in the evaluation, whether formative or summative, of prototypes. Involving stakeholders in all the phases of the development process, including design, is a basic requirement for any kind of interactive systems, as they know which objects, facts, concepts, and relationships are relevant in the domain of the application (Preece, Rogers, & Sharp, 2002). Secondly, the lack of a conceptual design process leads to little flexibility for changes with a high cost in resources when the environment does not meet the user requirements or when technological evolution suggests the addition of new services. And finally, this implementation-driven approach makes maintenance, interoperability, and software reuse difficult or nearly unfeasible. A conceptual model, which is independent from the implementation units, provides a picture of the system that can be understood by different people and that can serve as an intermediate level among different technological options. Each concept and relationship in the real world has a correspondence with an element or more in the conceptual model, which in turn can be translated into different implementation platforms.

The scarce use of conceptual models in VE development can be due to the fact that there are no broadly accepted development methodologies and design techniques that apply a software engineering perspective to this domain. A VE software crisis, in the line of the phenomenon identified in the ’60s as the software crisis (Gibbs, 1994) and in the ’90s as the hypermedia/Web software crisis (Lowe & Hall, 1999), can be diagnosed just to try and raise concern for the way most VE software is being produced.

Compared to traditional software systems, VEs make use of a richer variety of complex types of objects, behaviors, interactions, and communications in order to provide users with more and more realistic and useful environments and, consequently, typical conceptual modeling tools, such as E-R diagrams (Chen, 1976) or UML models (Booch, Jacobson, & Rumbaugh, 1998), do not seem appropriate enough to encompass all their requirements. On the other hand, VEs share with hypermedia systems a number of features and problems. Consequently, hypermedia modeling techniques that address the aforementioned design issues and have reached a certain level of maturity may be considered in the development of VEs as well.

In this chapter, the use of hypermedia design techniques for VEs will be discussed in order to:

- Make evident the relationship between the ideas, concepts, and principles underlying different modeling techniques in both disciplines, and

A Virtual-Reality Approach for the Assessment and Rehabilitation of Multitasking Deficits
Otmar Bock, Uwe Drescher, Wim van Winsum, Thomas F. Kesnerusand Claudia Voelcker-Rehage (2018). *International Journal of Virtual and Augmented Reality (pp. 48-58).*
www.irma-international.org/article/a-virtual-reality-approach-for-the-assessment-and-rehabilitation-of-multitasking-deficits/203067