

Software Engineering and HCI Techniques Joined to Develop Virtual Environments

María-Isabel Sánchez-Segura
Antonio de Amescua Seco
Juan-José Cuadrado
*Escuela Politécnica Superior
Universidad Carlos III de Madrid
Avda. Universidad, 30, Leganes 28911,
Madrid, Spain
(+34) 91 624 91 04
{misanche; amescua; cuadrado}@inf.uc3m.es*

Angélica de Antonio
*Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo s/n
28660 Boadilla del Monte, Madrid, Spain
(+34) 91 336 69 25
angelica@fi.upm.es*

1. Abstract

This paper focuses on a specific kind of software applications called Virtual Environments (VEs) or Virtual Worlds, Virtual Communities, etc., inhabited by avatars, which are the representation either of real humans users of the VE or autonomous agents. Virtual Environments (VEs), or Virtual World ,are growing very quickly and people demand easier and more believable ways to interact in these new sites [4],[10], [12].

From the point of view of software engineering, VEs can be seen as a special kind of software systems, in this sense, they must be analyzed, designed, implemented, etc., as any other software system. However, VEs development require special tasks and techniques, which are not provided by traditional software engineering methodologies. The question we would like to address is: define the products to be developed during the construction of an VE taking into

account other disciplines like Human Computer Interaction, Artificial Intelligence, etc., to enhance the quality of the development process.

1.1. Keywords

Virtual Environment, Software Engineering, Development Process, HCI techniques.

2. Introduction

Today, virtual environments are being used in many fields: Social Worlds, Finance, Commerce, Banking, Information System Sciences, Communication, CSCW (Computer Supported Collaborative Worlds), Education, Entertainment and Leisure, Medicine, Architecture, Geography, etc.[3]. This kind of application also seems to be the future of interactive programming [1] and can be used especially to demonstrate risky situations.

We are going to focus on the most recent VEs, based on 3D graphics and inhabited by Avatars¹ and autonomous agents.

Nowadays, the implementation process of VEs is well known but informal. In fact, good and useful results can sometimes be achieved with a modest outlay of hardware and resources. The problem comes from the very expensive construction [14], derived from following an informal process.

Therefore, the need for a more formal process is evident. From the Software Engineering perspective, a set of processes to formalize the development of this kind of application can be defined.

The Software Engineering Research Community is not the only one interested in the development of VEs. The need to define new techniques inspired by the Software Engineering discipline, is widely-known by scientists body related with HCI (Human Computer Interaction). [2].

Current software engineering process models [6] [7], must be enhanced allowing the use of different techniques coming from disciplines such as HCI or Artificial Intelligence to develop systems like VEs where the core of the development process will be provided by software engineering discipline. Specific techniques for the interface design will be provided by HCI discipline and Artificial Intelligence provides the appropriate techniques to design and implement the “knowledge” of the system because in this kind of applications the representation of the users into the VE must be as credible as possible.

3. Processes to be enhanced

In this section, we will present general and specific features of VEs-based applications with the aim of finding out why some processes must be used and what features cause the processes selected to be redefined.

Taking as software engineering pillars the process models [6] [7], Figure 1 shows the set of processes selected to be redefined or just enhanced. Those not selected could be used without any change because they do not depend greatly on VEs development

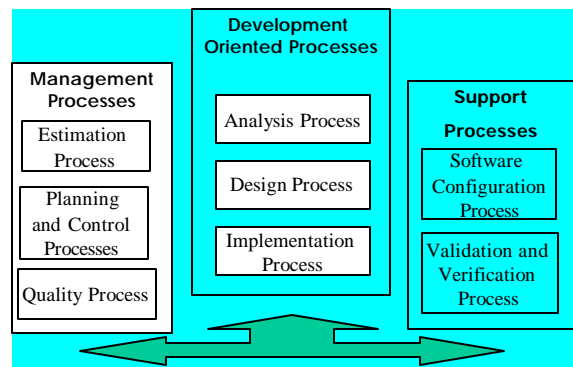


Figure 1 Processes Selected

In this paper we are just going to focus on Analysis and Design processes, the methodology which covers all the development processes was defined and was coined SENDA, the methodology details can be found in [13].

Up to now we have been talking about Process Models. The next question could be: *What about the development methodology?* To date, there are many methodologies whose goals are the disciplined development of software systems supported by Software Engineering paradigms. Those best-suited to VEs development are Object Oriented methodologies. L.Casey [11] said that the use of Object Oriented Techniques could be the key to a real advance in VEs development based on Software Engineering development.

In our experience, however, traditional Object Oriented methodologies are not enough for VEs development so, traditional methodologies must be enriched with a set of new tasks mainly in the analysis and design processes.

In this paper, we are going to focus on Analysis and Design processes due to the extension of this paper, with a short description of each process and their interrelationship, as well as the tasks where HCI and Artificial intelligence techniques could be useful. The symbol notation used can be found in [9], and the next notation has been used to name tasks "*Process Acronym plus order-number*". Possible process acronyms are:

- ❖ A: Analysis Process
- ❖ 3DD: 3D Design Process
- ❖ SD: System Design Process
- ❖ AD: Actions Design Process

Order-Number, in tasks name, does not mean the order in which the tasks must be completed. It is only used to name the tasks, without any significant notation, except for the process acronym.

¹ Avatar comes from Sanskrit and means incarnation.

4. VE Analysis Process

IEEE Requirements Standard [5] will be used for some tasks in this process. Object Oriented techniques [8] will be used for the Static and Dynamic Modeling. Conceptualization task need special treatment because there are a set of requirements such as auto-triggered behaviors, collision detection, etc., which cannot be defined with either Structured or Object Oriented Techniques. We have defined specific techniques to deal with this task.

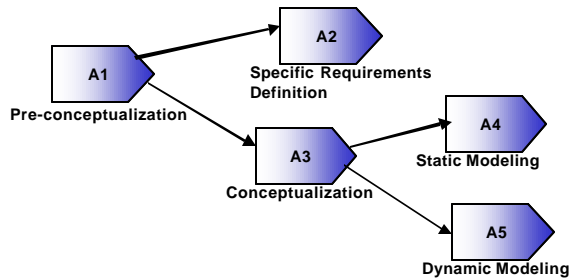


Figure 2 Analysis Process Tasks

The relationship between identified tasks, Figure 2, is defined through the output and input artifacts flow.

During the "Pre-conceptualization" task, the analyst, with the client's assistance, must elaborate a "List of Initial Requirements" and the "Problem Definition". The "Conceptualization" task takes these products in order to obtain the "Conceptualization document" that contains "Use Cases" and "Use Concepts". We propose a new term "Use Concept", not yet defined. Each Use Concept is defined by a brief description of the functionality, which will not be demanded directly by the user, and the following three fields:

- ❖ **Purpose:** Use Concept main goal.
- ❖ **Working Mode:** the way the Use Concept is going to be used.
- ❖ **Dynamic:** the Use frequency.

Conceptualization can be easily enhanced using HCI techniques, methods such as GOMS (Goals, Operators, Methods and Selection Rules) which is useful to model an activity as it is understood by the user.

5. VE Design processes

The traditionally known "Design Process", has been subdivided into three processes, due to the features of VEs developments.

- ❖ **3D Design Process:** includes the graphic design of scenarios, decorative objects, avatars, etc. The products obtained from this process can be seen as

requirements to explain "how" the environment and its elements are. This is why they are defined under *3D Design Process*.

- ❖ **Actions Design Process:** in VEs development special attention must be dedicated to give credibility to avatars and the rest of the elements. So a special set of tasks must be defined in this sense.
- ❖ **System Design Process:** this process corresponds with the traditional and well known Design process in software engineering methodologies.

5.1. Design Processes Roles Involved

In design processes, two kinds of roles are involved.

System Designer: typically assigned to define "how" is the application. By this, we mean the person who defines the control of the system following the System Analyst's definition of "What". In VEs, the System Designer is also the person who guides the Graphic Designer because of his/her knowledge of the application to be developed. The System Designer must also have basic knowledge in graphic design.

Graphic Designer: his job in the Design Process is feedback, View Maps, Environment Modeling Forms and Avatar Modeling ²Forms for the System Designer. After the feedback stage, Graphic Designers can begin the task of implementing 3D objects in the Graphic Design Tool selected, although this task is included in *Implementation Process*. It is important to remember that Graphic Designers are artists and do not need to have any computer knowledge.

5.2. 3D Design Process

We have fully defined this process. Many times, the problem Graphic Designer has during the implementation process, is the absence of guidelines on how to develop his task.

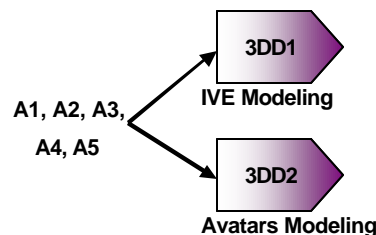


Figure 3. 3D Design Process Tasks

² See 5.2.

The *System Designer*, which is the person in charge of these tasks, must have a good knowledge of the physical IVE features and must also have a basic knowledge of graphic design. The result of these tasks will be a set of *forms and View Maps*, for the Graphic Designer to be used during the Implementation Process, to build three-dimensional models with the selected tools.

Both, "*IVE Modeling*" and "*Avatars Modeling*" tasks, take output product from Analysis Process Tasks. The former includes the detailed definition of a set of virtual spaces and the objects to be included, and the latter, the detailed definition of the virtual inhabitants, their appearance and physical structure. The products of these tasks are:

- ❖ *View Maps*
- ❖ *Environment Modeling Forms.*
- ❖ *Avatar Modeling Forms.*
- ❖ *Avatars Hierarchy Model.*

In a larger version of this paper, more detailed information is given on these tasks and techniques.

5.3. Actions Design Process

In VEs development, as in any kind of interactive development, "behaviour definition" is very important. VEs must be used to make interaction among users, through avatars, easier and believable.

The aim of this process is the definition of the actions that can take place within the VE. Many people will be involved in this process: psychologists, sociologists, etc., because a multi-disciplinary work is necessary to provide this kind of application with sufficiently interesting interactive features to give credibility to avatars and the rest of VE elements.

It is very important to emphasize the relation between "Actions Design" and "3D Design" processes. They must be coherent.

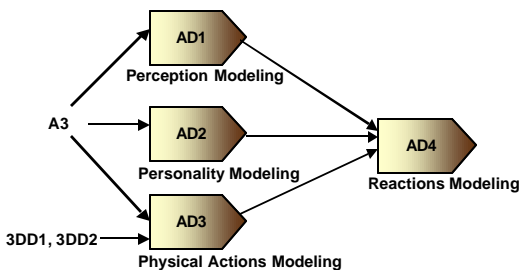


Figure 4. Actions Design Process Tasks

"Conceptualization" task A3, returns every action to be done by avatars and the rest of the elements into the VE. These actions include the detection of the events, which occurs in the environments, how avatars feel these events, and how these feelings are shown through the physical representation of elements.

These are all defined by the set of tasks included in this process. Techniques, like rules, etc, taken from Artificial Intelligent Field, must be used in this process. Also Genetic Algorithms are very useful for the avatars to be able to learn into the VE along its execution.

5.4. System Design Process

As we have mentioned before, the tasks included in this process are well defined in most Object Oriented methodologies.

"*Expanded Static Modeling*" (SD1) and "*Expanded Dynamic Modeling*" (SD2) tasks take:

- ❖ Class Diagram from "*Static Modeling*" (A4) and Transition Diagrams and Event Traces from "*Dynamic Modeling*" (A5) task (both from Analysis Process).
- ❖ "*Physical Actions Modeling*" task (AD3) to create new classes and methods derived from the exact definition of movements, etc., which is the output of AD3.

"*System Architecture Design*" (SD4) task and "*Data Persistence Design*" (SD5) tasks take the "Specific Requirements Document", returned from "A2", where restrictions or details to be used in SD4 and SD5 can be found. "*Interface Design*" task can be widely enhanced using HCI techniques because the interface use to be a very important part of the VE.

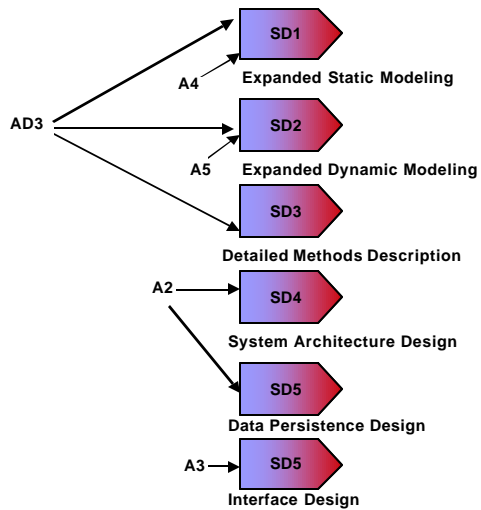


Figure 5 System Design Process Tasks

6. Experimental Results

The process model presented has been used to develop the Run&Freeze environment, funded by the Amusement ESPRIT IV project 25197.

Run&Freeze is a children's game. We have selected it because the rules are very simple but rich in features such as social interaction, mobility of the participants, immersion, spatial sense, etc.

The VE shown in Figure 6 and Figure 7 was developed from the process model SENDA described above.



Figure 6 Run&Freeze environment with two avatars



Figure 7 Run&Freeze environment lateral view

7. Conclusions and future trends

From the results obtained, it must be noted that the processes and techniques proposed as well as the combination among HCI and Artificial Intelligence techniques are powerful and flexible enough to allow for the creation of different VEs respecting the constraints of the application (to run in real time, etc.)

HCI techniques has been very useful in the design process to a better understanding of the problems the users could have with the use of the VE.

The proposed techniques have also useful to improve the communication among people with different backgrounds.

As we have already mentioned, there are many different processes to be enriched; the improvement of management processes, refining the estimation and the planning processes also using HCI knowledge to enhance estimation methods, because this discipline can provide information about the cost and effort in the interface design which can be used to refine the drivers in real estimation methods.

8. Acknowledgements

This research is partially supported by the Amusement Project (ESPRIT IV LTR project 25197)

9. References

1. Berenguer, X., (1997). Writing Interactive Programs. Magazine Formats.
2. Brown, J. Exploring human-computer interaction, and software engineering methodologies for the creation of interactive software. SIGCHI Bulletin (ACM), Vol. 29, Num 1, 32-35. 1997

3. CALT: The Center For Advanced Learning Technologies. INSEAD -Bd de Constance - F-77305 Fontainebleau Cedex, France. (2000). Available at: [http://www.insead.fr/CALT/Encyclopedia/Computer Sciences/VR/vr.htm](http://www.insead.fr/CALT/Encyclopedia/Computer_Sciences/VR/vr.htm)
4. Damer, B. Interacting and Designing in Virtual Worlds on the Internet. Tutorial for CHI97.
5. IEEE Std 830-1998. (1998) IEEE Recommended Practice for Software Requirements Specification Institute of Electrical and Electronics Engineers
6. IEEE Std. 1074-1991. (1991). IEEE, Standard for Developing Software Life Cycle Processes. New York (EE.UU.), IEEE Computer Society.
7. ISO/IEC Standard 12207:1995. (1995). Software Life Cycle Processes. Ginebra (Suiza), International Organization for Standarization.
8. Jacobson, I., et al. (1992) Object Oriented Software Engineering: A Use Case Driven Approach. Reading, MA: Addison-Wesley.
9. Kruchten, P. (1999). The Rational Unified Process. An Introduction. Addison-Wesley Object Technology Series.
10. Landauer C., Bellman K. Integration and Modelling in MUVES. Proceedings of the Virtual Worlds and Simulation Conference. VMSIM' 98. Society for Computer Simulation International. San Diego, USA, pp 187-192. 1998
11. Larijani, L.C. Realidad Virtual. McGrawHill, 1994.
12. McKay, D.P. Wath' s in a virtual world?. Proceedings of the virtual worlds and simulation conference. VSIM' 98. Society for computer simulation, San Diego, USA. P-9. 1998
13. Sánchez-Segura, M.I. Aproximación Metodológica al Desarrollo de Entornos Virtuales Ph. D. Thesis. Technical University of Madrid. Spain. (2001)
14. Venus: Virtual Environments For You. (1999) Available at: <http://leonardo.ucs.ed.ac.uk/venus/foryou/foryou.html>