Proceedings of the 5th WSEAS Int. Conf. on MULTIMEDIA, INTERNET AND VIDEO TECHNOLOGIES, Corfu, Greece, August 17-19, 2005 (pp40-45)

Improving Virtual Environments Analysis Process

MARIA-ISABEL SÁNCHEZ-SEGURA
Computer Science Department
Carlos III Technical University of Madrid
Avda. Universidad, 30, Leganes 28911, Madrid
SPAIN

ANGELICA DE ANTONIO
Facultad de Informática. Universidad Politécnica de Madrid
Campus de Montegancedo s/n
28660 Boadilla del Monte (Madrid)
SPAIN

ANTONIO DE AMESCUA
Computer Science Department
Carlos III Technical University of Madrid
Avda. Universidad, 30, Leganes 28911, Madrid
SPAIN

Abstract: - The use of Virtual Environments (VEs) is increasing rapidly and people are demanding easier and more credible ways to interact with these new sites. We define a VE as a special kind of 3D virtual environment, inhabited by avatars which represent humans in the VE, or even autonomous agents. This kind of software was selected because of its increasing importance as the new future trend in interactive software applications. From a software engineering point of view, VEs can be seen as a special kind of information system, so they must be analyzed, designed, implemented, etc. Our aim is to improve software engineering's traditional software processes to achieve quality VEs. In this paper, we present the guidelines to perform properly the analysis of a VE.

Key-Words: - Virtual Environment, Software Engineering, Analysis Process, Use Concept

1 Introduction

Today, virtual environments are being used in many fields: social, finance, commerce, banking, information system sciences, communication, CSCW (Computer Supported Collaborative Work), education, entertainment and leisure, medicine, architecture, geography, etc., [1]. This kind of application also seems to be the future of interactive programming [2] 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 Avatars and autonomous agents. These types of applications are called VEs, the acronym for Virtual Environments. They are also referred to as Multi-user Virtual Worlds [3], but in essence, they are the same.

Nowadays, a large number of VEs' technical problems have been solved but the development of VEs is not following a mature process yet, so it is increasingly necessary to provide developers with Software Engineering Paradigms, Principles and Procedures.

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 constructions [4] derived from following the informal process.

Therefore, the need for a more formal process is evident. This paper presents a formal approach to VEs analysis under the SENDA framework, developed to improve the quality of VE development.

The section "Analysis process in the SENDA framework", describes the way VE analysis should be performed in order to gather the features of the VE under development. The remainder of the paper describes conclusions.

2 Background

The Software Engineering research community is not the only one interested in this area. The need to define new techniques inspired by the Software

Engineering discipline is widely-known to scientific bodies related to HCI (Human Computer Interaction).[5]. Outside the software engineering discipline, some researchers like Fencott [6] or Kaur [7] from the HCI field have already dealt with the problem of developing VEs from a usability of software point of view.

In order to correct the current VE development deficiencies, the ISO 12207 [8] and IEEE 1074 [9] process models, pillars of Software Engineering, were tailored to VEs development. It must be noted that the modifications of these processes are valid for both structured and OO paradigms.

SENDA improves traditional software engineering process models by providing new processes and techniques, by improving some existing processes and techniques, and by using techniques provided by different disciplines. SENDA is described in detail in [10] and comprises 10 processes and 36 tasks, as seen in Fig 1. Each task is described through its input products, corresponding outputs, techniques to be applied and the participants.

The notation used to represent tasks and the relationships among them can be found in [11].

The acronyms of the main processes in SENDA are:

- A: Analysis Process
- 3DD: 3D Design Process
- MD: Multimedia Design Process
- SD: System Design Process
- CIAD: Components Internal Architecture Design Process
- SCI: Support Components Implementation Process
- CI: Core Implementation Process

"Process Acronym plus Task Acronym" has been used to name tasks.

Although the SENDA framework specifies processes and tasks that cover the whole development lifecycle, in this paper we focus on the description of the analysis process.

3 Analysis process in the SENDA framework

The Analysis process is one of the traditional processes which have been improved, providing some new tasks and techniques. A summary of the Analysis Process appears in **Table 2**, showing the new proposed elements in italics.

Many researchers suggest that the analysis phase must take into account a requirements specification task, which must gather only the system features and not how the system performs [12]. Like Sommerville [13], we think that although this idea is very attractive, it is not very useful in practice.

In SENDA, the first task to carry out in the analysis process is Pre-conceptualization, which allows the identification of the set of tasks to be executed. To achieve this, Questionnaire 1 must be completed as the answers in this questionnaire allow the project manager to configure the development process.

Once the Pre-conceptualization task is finished, the "Conceptualization" task must be initiated in order to obtain the "Conceptualization document" that contains "Use Cases" and "Use Concepts". Use Cases are taken from the Unified Modeling Language standard. We propose a new term "Use Concept", not yet defined, as a tool to describe the system functionalities not triggered by an external actor.

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's main goal.
- Working Mode: how the Use Concept is going to be used.
- Dynamic: the Use frequency.

Table 1 illustrates a use concept which represents the functionality to prevent the avatar from colliding with an obstacle in its path.

USE CONCEPT	
Use concept name: The avatar must not collide with the walls **Use concept code:** Concept(7)	**Purpose:** To prevent the avatar from going through the walls so the environment is more credible. **Working mode:** When an avatar arrives at a wall, it is not allowed to go through the wall and the avatar must stop. **Dynamic:** Each time the avatar is near a wall.

Table 1 Use concept example

As VEs are constantly evolving, the types of virtual reality devices, the development software, the hardware, etc. have to be chosen as soon as possible in order to test the compatibility of these elements, which influence the rest of the development processes. We propose the Specific Requirements task to list the VEs specific decisions. We propose a set of categories to classify use cases and use concepts according to their special characteristics. These categories are: connection to the VE, virtual reality devices interface, animation, perception, VE evolution, reasoning and decision, communication with other connected users, and scene visualization. Then, each of these categories will be dealt with in the Components Internal Architecture Design process. Using this classification, it is easy to trace the requirements into use concepts and use case. It can also be used as a checklist to verify the

completeness of the requirements. If a category is empty the analyst can wonder if the requirements were properly extracted, and take this opportunity to refine them.

Static and Dynamic Modeling tasks have been taken from object oriented methodologies and related to the rest of SENDA tasks.

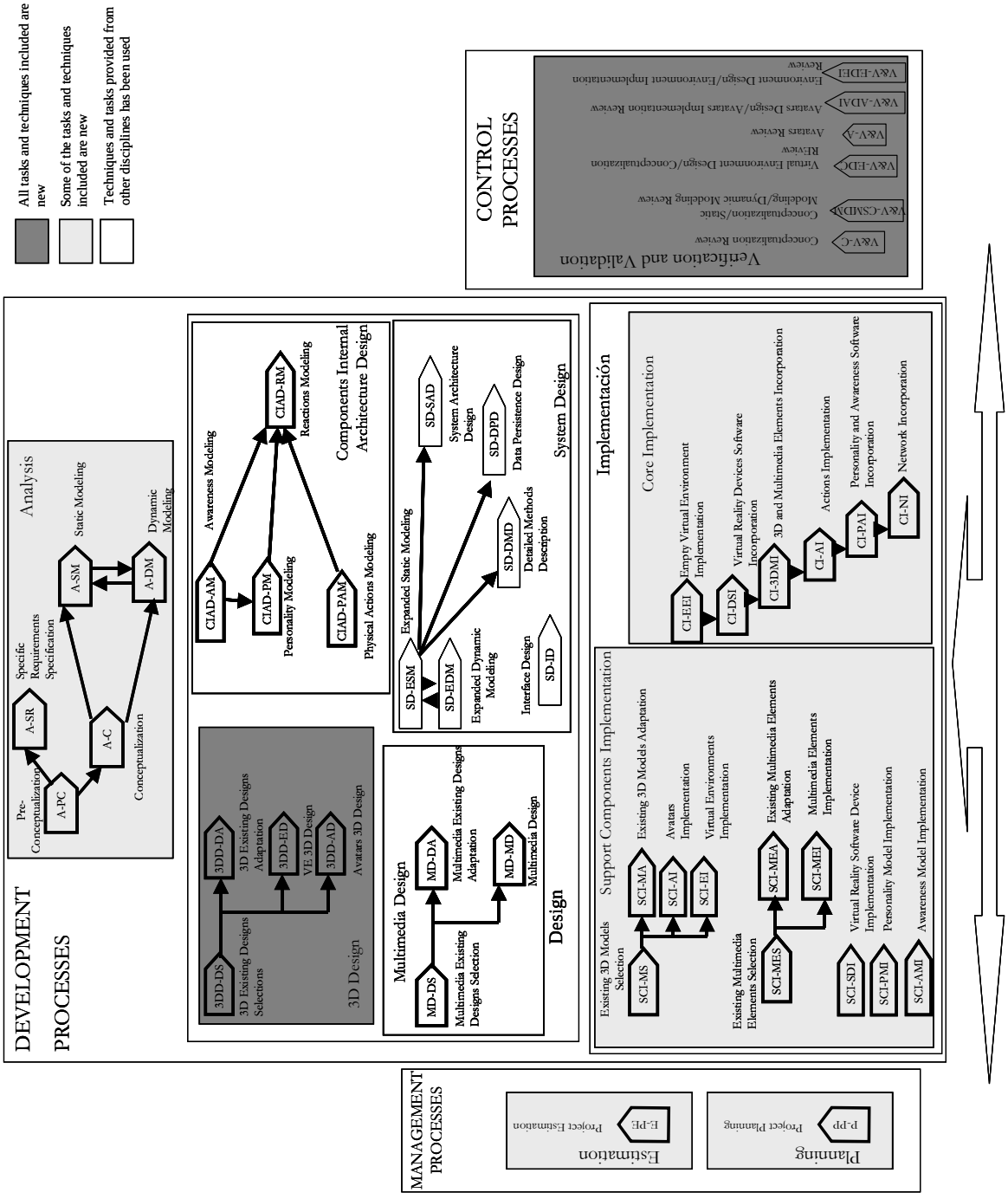
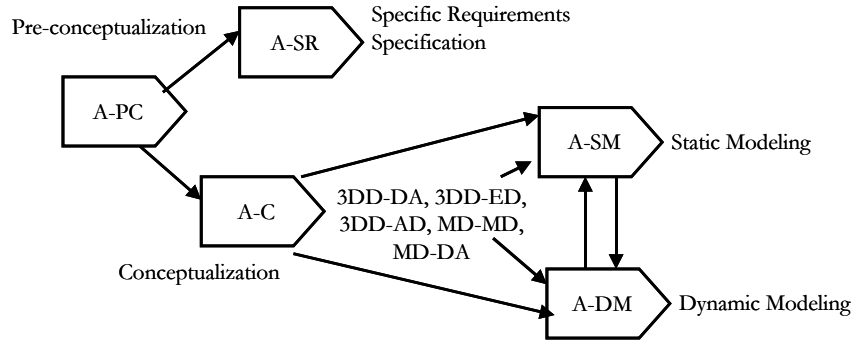


Fig 1 SENDA processes and tasks

ANALYSIS PROCESS



TASK NAME	TASK ACRONYM	TASK DESCRIPTION		
Pre-conceptualization	A-PC	Artifacts	Input	Contract definition
			Output	Selected tasks to be developed according to the kind of VEs to be developed.
		Techniques		Interviews with the clients
		Participants		System analyst
Specific Requirements Specification	A-SR	Artifacts	Input	Problem definition
			Output	*Specific Requirements Document*
		Techniques		Alternatives study, interviews.
		Participants		System analyst, client, user.
Conceptualization	A-C	Artifacts	Input	Selected tasks to be developed, output of A-PC
			Output	Problem definition, Acronyms, Abbreviations, Initial list of Requirements, Use cases and use concepts classified.
		Techniques		Use cases, *Use Concepts*
		Participants		System analyst, client, user.
Static Modeling	A-SM	Artifacts	Input	Conceptualization Document, Outputs of 3D Design tasks, Outputs of multimedia design process.
			Output	Classes model. Amplified classification of Use cases and use concepts table.
		Techniques		Structural diagram
		Participants		System analyst
Dynamic Modeling	A-DM	Artifacts	Input	Conceptualization document, classes model
			Output	Dynamic model
		Techniques		UML sequence diagrams, Scenarios, Operational contracts
		Participants		System analyst

Table 2 Analysis Process

4 Conclusions

The whole SENDA framework has been used in the development of some VEs. From the results obtained it must be noted that the processes and techniques proposed 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.). A detailed explanation of the results obtained using SENDA in different projects can be seen in [14].

Moreover, even if these techniques guide the graphic designer, they do not interfere with their artistic approach to the task. The techniques are independent of the application implementation.

The proposed techniques have also proved useful to verify and validate the graphic designer's job after 3D models are implemented.

Proposed tools and mechanisms allow:

- communication between Graphic and System Designers.
- comparison between the designed and implemented 3D models.
- reuse of Sub-VEs design, and even implementation between different projects though the database of designs.

Is VE only for guided tours without any type of interaction?	**Yes** **No** ☐ ☐ If Yes, ignore the Internal Characteristics of the VE components in the Design process; the ICS-IMCI and ICS-IMP tasks of the Implementation of Components and Support process; and the IMP-IMCL task of the Principal Module Implementation process. .
Will VE be networked?	**Yes** **No** ☐ ☐ If No, delete the IMP-ISRE task of the Implementation of the Principal Module process.
Will VE use virtual reality mechanisms?	**Yes** **No** ☐ ☐ If No, delete the ICS-SDRV task in the Implementation of Support Components process and the IMP-ISRV task in the Implementation of the Principal Module process.
Will VE be used for teaching?	**Yes** **No** ☐ ☐ If Yes, a tutor module should be considered in the general architecture of VE.
Will VE be used to develop social relations?	**Yes** **No** ☐ ☐ If No, the DAI-SMCI task should be deleted from the Internal Characteristics of Components Design process of the VE. If Yes, the need to include a personality module or a social module in the VE should be considered.
Will the VE have 3D elements?	**Yes** **No** ☐ ☐ If No, the 3D Design and the ICS-S3D, ICS- AR3D, ICS-IA3D, ICS-IVE tasks of the Implementation of Support Components, and the IMP-IO3D tasks can be deleted. Remember that the part corresponding to loading the 3D elements of the VE should not be executed.
Will the VE have multimedia elements?	**Yes** **No** ☐ ☐ If No, the Multimedia Elements Design, and the ICS-SEM, ICS-AREM, ICS-IEM tasks of the Implementation of Support Components can be eliminated. Remember that the part corresponding to the insertion of multimedia elements of the IMP-IO3D task should not be executed.
Will the VE have avatars guided by agents?	**Yes** **No** ☐ ☐ If Yes, the avatars should be modeled to be controlled by agents, that is, they should be automatically controlled by an interface within the system. Therefore, the formalism of Use Concepts to define some of the requirements of the Conceptualization task should be used.
Will the VE control the personality model of the avatar partially or totally?	**Yes** **No** ☐ ☐ If No, the DAI-SMCI task of the architecture of the Internal Components Design process, and the ICS-IMCI task of the Implementation of Support Components process can be deleted.
Will the VE partially or totally control the reasoning model of the avatar?	**Yes** **No** ☐ ☐ If No, the DAI-DMR task of the architecture of the Internal Components Design process can be deleted.
Will the VE totally or partially control the model perception of the avatar?	**Yes** **No** ☐ ☐ If No, the DAI-IMP task of the architecture of the Internal Components Design process architecture of the Internal Components Design process and the ICS-IMP task of the Implementation of Support Components should be deleted.

Questionnaire 1 Pre-conceptualization Questionnaire

References:

[1] 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/ComputerSciences/VR/vr.htm

[2] Berenguer, X., Writing Interactive Programs. Magazine Formats. 1997

[3] Damer, B. Interacting and Designing in Virtual Worlds on the Internet. *Tutorial for CHI97*. 1997

[4] Venus: Virtual Environments For You. 1999 Available at: http://leonardo.ucs.ed.ac.uk/venus/foryou/foryou.html

[5] Brown, J., Encarnaçao, J., Schneiderman, B. Human-Centered Computing, Online communities, and Virtual Environments. *IEEE Computer Graphics and Applications.* Vol 19, Nº6, Pp 70-74. 1999

[6] Fencott, C. Towards a Design Methodology for Virtual Environments. *Workshop on User Centered Design and Implementation of Virtual Environments.* University of York. UK. 1999

[7] Kaur, K. Designing Virtual Environments for Usability, Ph. D. Thesis, City University, London. 1998

[8] ISO/IEC Standard 12207:1995. Software Life Cycle Processes. Ginebra (Suiza), International Organization for Standarization. 1995

[9] IEEE Std. 1074-1991. IEEE, Standard for Developing Software Life Cycle Processes. New York (EE.UU.), IEEE Computer Society. 1991

[10] Sánchez-Segura, M.I. Aproximación Metodológica al Desarrollo de Entornos Virtuales Ph. D. Thesis. Technical University of Madrid. Spain. 2001

[11] Kruchten, P. *The Rational Unified Process. An Introduction.* Addison-Wesley Object Technology Series. 1999

[12] Davis, A., Hsia, P. Status report: requirements engineering. IEEE Software. Vol 10 Nº 6 pp 75-79. 1991

[13] Sommerville, I., Sawyer, P. *Requirements Engineering: a good practice guide.* Willey and Sons. 1997

[14] Sánchez-Segura M. I., Cuadrado J. J., de Antonio A., de Amescua A., and García L. "Adapting traditional software processes to virtual environments development". *Software Practice and Experience.* Volume 33, Issue 11, pp 1050-1080. 2003. Online http://www3.interscience.wiley.com/cgi-bin/jhome/1752.