Tutorial Book of Virtual Concept 2005
Biarritz, France, November 8th – November 10th, 2005

Intelligent Virtual Reality Environments (IVRE): Principles, Implementation, Interaction, Examples and Practical Applications.

Osório, F. S.[(1)], Musse, S. R.[(1)], Santos, C. T.[(2)], Heinen, F. [(1)], Braun, A. [(1)], Silva, A. T. [(1)]

(1) : Unisinos University – Applied Computing
PIPCA - Bloco 6B – Av. Unisinos, 950
São Leopoldo – Brazil / 93022-000
Phone: +(55-51)591.1122 *Ext.* 1629 *Fax:*+590.8162
E-mail : {fosorio, soraiarm, heinen}@unisinos.br

(2) : UFRGS – Instituto de Informática - PPGC
Av. Bento Gonçalves, 9500 – Bloco IV
Porto Alegre – Brazil / 91501-970
Phone:+(55-51)3316.*6161 Fax:* +3316.7308
E-mail : {ctsantos}@inf.ufrgs.br

Abstract

This tutorial aims to present new trends, methods and applications of Virtual Environments endowed with information/knowledge in order to provide to the user more interesting immersive interaction. We present concepts related to environment modelling, visualization and user interaction, mainly focused on some important proprieties: dynamic entities, adaptive environment, intelligent agents and behavioural models. Intelligent Virtual Reality Environments (IVRE) integrate Virtual Reality, Artificial Intelligence and Simulation tools and techniques in order to provide more realism, more dynamic environments and to improve interaction with users. This tutorial aims to present an overview of some important techniques related to IVRE implementation, for example: intelligent objects; autonomous agents control architectures; knowledge acquisition, representation and manipulation techniques; behavioural models (physically based and/or human based); etc. The IVRE have been employed in many applications, described in this tutorial, such as: e-commerce, e-learning, games, simulation of real situations (from crowd simulation to robotics) and also visualization of animations and study of specific behaviours. We conclude this tutorial with some examples of IVRE practical applications that our group developed recently: CROMOS - Crowd Modelling and Simulation (e.g. simulation of emergency situations) and ADAPTIVE - Adaptive 3D Intelligent Virtual Environment (e.g. adaptive virtual bookstore and e-learning environment).

Table of Contents

CHAPTER 1
Intelligent Virtual Reality Environments (IVRE): Principles, Implementation, Interaction, Examples and Practical Applications.

From Static Virtual Reality Environments to IVRE – Intelligent Virtual Reality Environments: Interactive, Reactive, Adaptive and Populated with Autonomous Agents.

I. Introduction

Virtual Reality (VR) is an attractive alternative for the development of more interesting interfaces for the user. The environments that make use of VR techniques are referred as Virtual Reality Environments (VEs or VREs). In VEs, according to [Avradinis 2000], the user is part of the system, an autonomous presence in the environment. He is able to navigate, to interact with objects and to examine the environment from different points of view. As indicated in [Frery 2002], the three-dimensional paradigm is useful mainly because it offers the possibility of representing information in a realistic way, while it organizes content in a spatial manner. In this way, a larger intuition in the visualization of the information is obtained, allowing the user to explore it in an interactive way, more natural to humans.

Nowadays, the attention of research community has been focused to the integration of Artificial Intelligence (AI) and VEs. The objective is to create VEs that explore the use of intelligent entities, adopting an effective graphical representation and animation, allowing interactions of different forms between the VE components and improving dynamicity, realism and usability of these systems. According to [Aylett 2000, 2001], the environments that explore such integration originated a new research and development area, called Intelligent Virtual Environments (IVEs). When these Virtual Environments are implemented through the adoption of 3D Virtual Reality interfaces, they are called IVREs – Intelligent Virtual Reality Environments. As indicated by Anastassakis (2001) and Aylett (2000), the potential applications for these environments are substantial, since they can be applied in a variety of areas, mainly related to applications in simulation [Musse 2003, Heinen 2002b], entertainment and games [Milde 2000, Nijhold 2000], education [Rickel 1997, Santos 2004a], e-commerce [Santos 2004b, Chittaro 2002] and industrial applications [Rickel 1997].

IVRE research involves a multidisciplinary research approach, including concepts and techniques from different fields including: Computer Graphics (CG), Virtual Reality (VR), Artificial Intelligence (AI), Artificial Life (AL), Behavioural studies, Robotic concepts, Multi-Agents interaction, Interfaces and Human-Computer Interaction, and many other knowledge areas. This diversity makes IVRE a fascinating and challenging research field.

There are different ways and approaches used to develop IVRE systems, but most of them focus on the integration of Computer Graphics (CG) and Artificial Intelligence (AI) techniques. One of the most interesting possibilities of virtual environments concerns the possibility of populating them with virtual agents - intelligent and autonomous agents! Moreover, it is important that virtual agents could behave in a realistic way, regarding their visual aspect, their motion and also the realism of their actions. However, to develop really intelligent agents which could perceive, adapt and behave in a non-deterministic way in such type of environments is not a trivial task. This tutorial presents aspects of: i) how to model virtual environments in order to provide information for virtual agents reducing the complexity of virtual agents' behaviours; ii) how to control agents in order to accomplish a specific task; iii) how to implement interactions between agents and the environment and interactions among groups of agents; iv) how to automatically adapt the environment; and v) how to provide and improve the agent knowledge about the user and the environment;

These and other aspects will be discussed in this tutorial as well as we will describe some practical examples of IVRE application we have developed in our research group. This tutorial is organized as follows. In next section we describe some aspects of Virtual Reality Environments, including its modelling, visualization and interaction. Section III discusses how to add intelligence into VEs and some aspects of Intelligent Virtual Reality Environment, like its organization and the introduction of agents and group of agents with intelligent behaviours. In addition, we present some aspects of the connection between environment and virtual population, in this work related as individuals, groups and crowds. Section IV describes aspects of crowds and groups of virtual humans populating virtual environments, while Sections V and VI discusses some applications of IVREs.

II. Virtual Reality Environments (VRE)

Virtual reality (RV) is a set of advanced interface tools and techniques enabling: user immersion on virtual worlds, navigation in these environments and interaction with other real or virtual people and objects present inside this synthetic three-dimensional environment. The interaction should be in a natural and intuitive way, and if possible, using multi-sensorial channels. In this section are exposed some initial concepts about Virtual Reality (RV), starting from a global view of this area. An introduction to modelling techniques and visualization of three-dimensional environments will be presented, considering environments exploring non immersive navigation and interaction with the user and also the use of special immersion devices.

II-1. VR Environment Modelling

The 3D virtual environment modelling is the first stage needed in order to construct a VR application. It starts with the geometric, shape and spatial description of the objects and places represented in the environment. Some usual techniques adopted to model VR environments are: geometric representations, spatial occupation grids and topological representations (Figure 1).

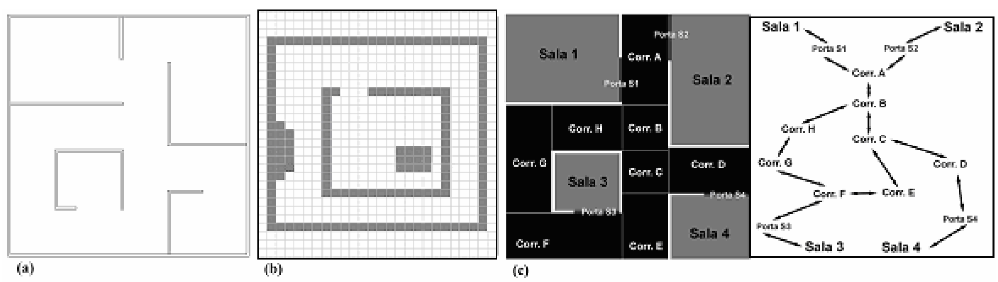


Figure 1: Maps: Geometric (a); Occupation Grid (b); Topologic-Semantic (c); [Heinen 2002a]

The geometric model of the environment can be easily built using different available software for 3D modelling. Among the main software used for environment modeling can be mentioned: 3D Studio MAX[1], Maya[2], Wings3D[3], Internet Space Builder and Internet Scene Assembler (ISB/ISA) [4] and Truespace[5]. These tools are used for 3D models creation and also for generation of animated characters, usually providing the possibility to create predefined animations through character's skeleton manipulation (e.g. Cal3D[6]).

---
[1] 3Dstudio Max - http://www.discreet.com/3dsmax/
[2] Maya - http://www.alias.com/
[3] Wings3D – http://www.wings3d.com/
[4] ISB / ISA – http://www.parallelgraphics.com/
[5] Truespace - http://www.caligari.com/
[6] Cal3D - http://cal3d.sourceforge.net/

All these tools support import/export of the modeled structure in several formats, including some widely known file formats like VRML (Virtual Reality Modelling Language[7] and 3DS. These 3D virtual environments are also very usual in action games, and some specific tools and world/character description formats are adopted in this domain (e.g. PK3/BSP, MD2/3/4, X Format).

So, in order to create a Virtual Environment (VE), we need to create: virtual places, virtual objects and virtual characters. All these VE components can be described using a static structure, but they can also include some predefined animations. Besides the object/character animations, other information can be added to them, as for example, environment topological information, object semantic information, objects associated actions and hot spots, etc. As the complexity of the 3D VE increases, moving towards 3D IVEs (Intelligent Virtual Environments), the complexity of objects and the associated information to these objects also increases, as discussed in the next sections.

Besides using the above cited software for “manual” modelling of 3D environments, some “intelligent” 3D VE semi-automatic generation tools has been proposed in the literature [Hu 2003]. Sometimes is possible to start from a 2D description (e.g. building blueprints, house floor plans, city streets and buildings map) and automatically obtain a 3D description, giving "volume" and "height" to the objects (e.g. walls).

Another semi-automatic method to generate virtual cities, based on maps of real cities, was presented by Marson et al. (2003). In this model, geometric descriptions (VRML) as well as semantic descriptions (XML) of the cities are considered. The input parameters used in this system concern the demographic data, such as population density, predominant social class and type of local constructions, among others. This model also allows real data input, like 2D maps of existing cities, supplying the geometry of the blocks. The main objective of this project was to provide a tool to make possible fast generation of complex virtual cities, including semantic information in order to allow the development of behavioural simulations using synthetic agents.

In the AdapTIVE environment (Santos 2004), the 3D structure of the environment, a flat building formed by a group of rooms and sub-rooms, is built automatically through an application implemented using the Java3D API. The user defines the number of rooms and their dimensions and the application generates automatically the corresponding 3D structure, in the VRML format. The rooms (and their content) are also automatically organized according to the user model of preferences.

Besides the structural information about the 3D virtual space (manually or automatically generated), another important component to be added into the description of the virtual environment are intelligent information about this environment and about its use. The structural/geometrical information should be

[7] VRML – X3D - www.web3d.org

enclosed by additional information like: topology (connectivity), predefined paths (predefined routes), semantic object information (possible actions), intelligent objects (associated to actions, scripts and hot spots), etc. We will return to discuss this topic in sections III-1 and III-3.

II-2. VR Environment Visualization

The visualization of 3D environments involves the use of specific computational tools and techniques. The most important graphical rendering support tools (graphical software packages) widely adopted in the market are OpenGL[8] (Multi-Plataform), DirectX[9] (Microsoft Windows) and Java3D API[10] (Multi-Plataform using OpenGL or DirectX support). These graphical libraries offer a complete set of Bitmap, 2D and 3D display routines, and even more (e.g. I/O devices interface), with support to different programming languages like C, C++, Delphi and Java. They are mainly used for 3D objects visualization, visualization of virtual environments, and for games development.

OpenGL (Open Graphics Library) (Woo 1998) is now an international standard for 3D graphical applications development, and is implemented in hardware in most of 3D graphic hardware accelerated cards. The OpenGL library provides access to all graphical resources available in the computer hardware, increasing the performance of different types of applications. Although it is a powerful and complete graphical library, programming in OpenGL remains relatively simple.

DirectX is an API provided by Microsoft to developers of multimedia, 3D visualization and games applications. This API provides a high-level interface to access the hardware components, such as sound and graphic cards. It controls low level functions, including providing support to different input devices as keyboard, mouse and joysticks, as well as access and control of sound devices.

Java3D API provides an object-oriented interface to a set of high-performance graphical functionalities, allowing a high-level programming of sophisticated 3D applications. This API provides a set of functions used for construction, rendering and control of 3D objects. Using this API, graphic objects can be easily incorporated into applications and Java based applets. Besides that, this API can also import files in VRML and 3DS formats. The low-level access to the computer hardware functions implemented in this API is accomplished through the use of OpenGL or DirectX functions, according to the initial user specified configuration.

Finally, all the above 3D graphical visualization toolkits, enable users to render complex virtual environments, and also provide resources to control different virtual environment components (e.g. illumination, navigation and virtual camera positioning, object texture mapping, shading, etc). These toolkits also provide resources to manage events, but usually the interaction control

[8] OpenGL – http://www.opengl.org
[9] DirectX - http:://www.microsoft.com/windows/directx/
[10] Java 3D – http://java.sun.com/products/java-media/3D/

functions (e.g. collision detection, automatic camera positioning) should be implemented in the user application level.

II-3. VR Navigation and Interaction

Navigation and interaction are important properties in a virtual environment. Navigation corresponds to the ability of moving through an environment, exploring its features and components. This movement across the VE should adopt an appropriate behaviour related to the type of represented element: avoid collisions against walls and obstacles, simulate objects' movement in an appropriate way (may include kinematics and dynamics of simulated models) and animate the characters in an appropriate way (controlling people's, animals and others beings movements). These items will be discussed later along this text, because they are directly related to the conception of an Intelligent Virtual Environment.

Interaction in VEs involve some aspects that include to avoid and treat object collision, select and move objects in the scene, and also can involve, information exchange, communication, voice synthesis and recognition and gestures exploration (multi-modal interactions).

On the other hand, interaction in VEs is usually associated to special hardware devices, like Data-gloves, HMDs, Virtual Caves and Haptic Devices, which can create special ways of interaction between users and VEs. In this work this aspect (interaction with special hardware devices) is not directly treated, once our main focus is on the environment and its objects and population. Special hardware devices, like those cited above, merit a special chapter that should be dedicated only to them.

III. From VREs to Intelligent VREs

Virtual Reality Environments (VREs) are very useful because they offer the possibility to represent different elements (e.g. real world things, documents, information) and its spatial organization in a realistic form. This allows us to obtain a more intuitive representation and manipulation of these elements present in the Virtual Environment, and even a child can easily interact with this kind of computer application. So, in this text, even if we consider the existence of textual or 2D virtual environments, we will make references to Virtual Environments (VEs) as Virtual Reality Environments represented in a 3D space.

At present, many works have made use of VE in order to present simulations results and to show realistic animations approximating as much as possible the users of real world experiments. The number of implementations increased considerably and several possible applications are available nowadays, including industrial plant simulation, games, entertainment environments (e.g. virtual museum galleries, virtual tourism), e-learning and e-commerce applications, among several other possible applications.

This section focus on the evolution of Virtual Environments to Intelligent Virtual Reality Environments (IVREs), presenting the main motivation for the integration of this “intelligence” into the conventional virtual environments. Virtual environments with intelligent agents and intelligent objects will be presented, and also the intelligent organization/adaptation of the environment.

III-1. Intelligent Environments

An Intelligent Virtual Environment (IVE) is a virtual environment, similar to our real world, filled with intelligent entities (e.g. virtual agents, autonomous agents, intelligent objects). Several works have been proposed aiming to create virtual humans as intelligent entities in these IVEs, which includes approximate the maximum as possible the virtual agent animation to the natural human behavior. In order to accomplish this task, the agent must be capable to interact with the environment, interacting with objects and other agents. The virtual agent needs to act as real people, so he should be capable to extract semantic information from the geometric model of the world where he is inserted, based on his own perception.

Some existing work related to IVREs establishes that the agent keeps all the information about the entities present in the environment. In this case, it can be complex and expensive to implement this approach. In some IVREs it is possible to remove part of the information load from the agents, so he can be simplified. The main idea behind this is the construction of intelligent environments, where the virtual agent can ask the intelligent environment for some information like: the position of a certain place, the best path to arrive to this place, the semantic regarding this place, and other information that can help the agent to act in this virtual environment.

Thomas and Donikian (2000) present a model of virtual urban environment using structures and specific information for behavioural animation. It was developed in this work a city modeller that allows the automatic construction of complex urban environments, where take place virtual humans behavioural animations. The main objective is to simulate streets and sidewalks urban life (closed places and interior of buildings are not considered). Each place has its functionalities defined (houses, stores, etc) and the input/output flow of pedestrians depending on the hour of the day.

Farenc (1999) also presents the creation of an environment with information, dedicated to urban life simulation. Tools and methods are proposed to create and to provide the necessary information for virtual humans' animation in this city, using an informed environment. In this environment, geometric information is supplied and also semantic notions about the simulated environment, allowing more realistic human behaviors simulation. Considering this, the virtual humans can interact more easily with the VE elements, since they can have certain knowledge about the environment.

ViCrowd is the name of the system and model we have introduced (Musse and Thalmann, 1997), (Musse et al, 1998) and (Musse et al, 1999). ViCrowd aims at dealing and managing with the crowd information allowing directability (crowds can be programmed as easily as one only actor), interactivity (users and participants can guide and interact with crowds during the simulation) and autonomy (behaviors can be generated automatically based on rules associated to the crowd). Then, in ViCrowd, at the same time, we can program behaviors, interact with groups of actors and describe rules to provide self-animating agents who can decide their actions and motions.

Some objectives of ViCrowd are:

i) Modeling of crowd information and structure.

ii) Dealing with basic behaviors related to crowd problems, e.g. sharing virtual space, keeping the individual space in order to avoid collisions, be near to agents of same group, keep group goals, etc.

iii) Studying and modeling of some sociological effects which can occur in crowds or groups of people, e.g. membership, leadership, goals changing, etc.

iv) Providing an interface to easily program crowds as well as behavioral rules in order to describe events and reactions to be treated by agents.

v) Including the required structure of crowd control in order to provide interactivity, directability and autonomy in virtual crowds.

In our current researches, information used by agents in crowd simulation are obtained in a semi-automatic way. Indeed, specialized systems are responsible for extract geometric information from virtual environments in order to provide enough data to agents to populate them. Firstly, geometric data is loaded and then some components can be automatically extracted, such as buildings. Then, the tool is user controlled in order to generate information to be stored in VE database. In addition, paths, obstacles and interest points are defined by the user.

III-1-1. Intelligent Virtual Environment Organization

An important aspect in a virtual environment concerns the spatial organization of its components. Such organization demands, in many cases, a specific disposition of the components, according to some semantic criterions. For example, in a virtual e-commerce store one can be interested in automatically group and place products in the VE according to the department they belong (electronic, bazaar, or other). This approach also can be applied in Educational Virtual Environments for Distance Learning, where the educational contents can be organized in groups and places according to the knowledge area they represent. So, it is very interesting to provide in IVREs a set of tools for help users to organize contents in the environment, providing an automation of this task as much as possible.

In this context, metadata about VE components and objects (also referred as contents models) can be adopted in order to allow the automatic components organization in the environment. This metadata contains information describing the corresponding component, such as category, keywords, among other semantic information. Components metadata can be introduced manually by a domain specialist, or an automatic metadata information acquisition approach can be adopted, usually consisting of the application of machine learning techniques. For example, the metadata acquisition can be performed by an automatic text categorization algorithm, using as input a textual description of a specific object. Even non-textual objects (such images, videos and 3D objects), can have initially a short description in natural language, and then an automatic metadata extraction algorithm can be applied.

Nowadays, machine learning and information retrieval techniques are broadly applied in the organization and recovery of textual information based on contents models (e.g. the amazon bookstore site[11]). However, an analogous process of automatic spatial organization of the contents in VEs is still few explored. Santos and Osório (2004b), proposed the use of a method for automatic extraction of models contents and using this information (associated to the user preferences model) automatically organize the virtual environment contents.

III-1-2. Intelligent Behaviour

Various researches on behavioural modeling have mainly focused on several aspects of virtual human control and autonomy. In this section we present a model

[11] Amazon Books – http://www.amazon.com/

describing Intelligent Information, required in order to perform behavioural simulations, outside the virtual agents. In our case, we are interested in modeling Intelligent Virtual Environments (IVE) in order to provide information to be used in behavioural simulations. Our main goal is to be able to produce behavioural simulations in Virtual Environments using virtual agents endowed with different Levels of Autonomy (LOA): from low to high. Table.1 shows a possible classification of agents LOA [refxx].

BEHAVIOUR CONTROL	GUIDED AGENTS	PROGRAMMED AGENTS	AUTONOMOUS AGENTS
LOA	Low	Medium	High
Level of Intelligence	Low	Medium	High
Execution Frame-rate	High	Medium	Low
Complexity of Behaviors	Low	Variable	High
Level of Interaction	High	Variable	Variable

Table 1: Characteristics of different agents' control: Guided agents deal with Low *Level of Autonomy* (LOA) and autonomous agents with High LOA.

IVEs can be modeled including geometric and semantic information. Geometric information is related to locations of places in the IVE, paths to go to specific locations, regions to walk, objects to avoid, etc. Semantic information deals with behavioral aspects of the spaces: specific actions or movements to be applied, emotional states to be assumed, information related to the characteristics of the space (like comfortable, beautiful, etc). Afterwards, the autonomous agents are able to evolve in the IVE using only the information found in the space which can be used in the decision process.

The main goal of some researches on IVE domain is to demonstrate that not sophisticated virtual humans can behave in a convincing way only by interacting with IVEs. Figure 2 shows three available configurations of simulation concerning different levels of autonomy (LOA) for virtual humans (VH) as well as the level of informed information (LOII) contained in the VEs. According to Figure 2 (c), the agents should have a high level of autonomy in order to be able to evolve in non-Intelligent VEs (Low LOII). On the other hand, IVEs can be easily integrated with agents endowed with different levels of autonomy (Figure 2 (a) and (b)).

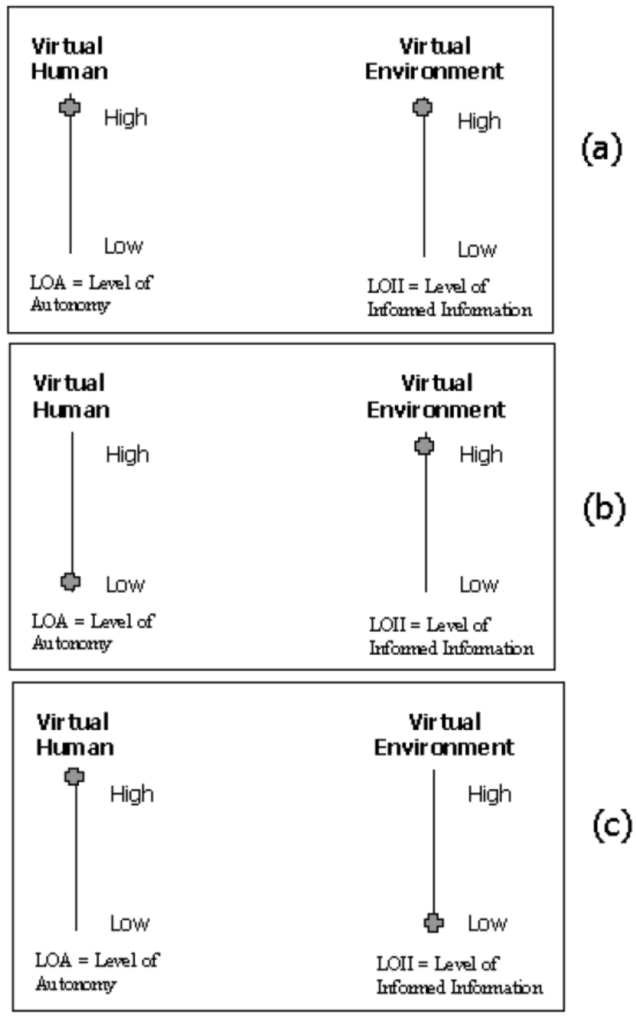


Figure 2: High LOA for VH and High LOII for VE (a); Low LOA for High LOII (b); High LOA for Low LOII (c)

III-1-3. Intelligent Agents

Virtual environment components are implemented and described in several different ways. When these components are animated virtual characters with a specific behaviour, they usually are called “intelligent virtual agents”, or in games terminology, they are called NPCs (Non-Player Characters). The definition of what is an “intelligent agent” is complex. Since they are very important elements in IVREs, in the next sections we will discuss more in details about the intelligent virtual agents: their animation, behaviour and control, and also about the simulation of groups and crowds of agents.

III-2. Intelligent Virtual Agents

From a more general point of view, Intelligent Agents approach has been broadly investigated and applied into several areas of the Computer Science, such as Artificial Intelligence (AI), Computer Networks (CN), Information Systems (IT) and Computer Graphics and Virtual Reality (CG/VR). As one of the central approaches studied in the modern AI, quickly it was adopted in other areas. This interdisciplinary characteristic reflects the current state of the research in this field, where there is no single and "perfect" definition or even a common sense about what is an agent and which are its main properties and desired abilities. Each research group has its own definition of an Intelligent Agent according with their objectives and applications, even if some points of these different definitions are common: Goodwin (1994); Maes (1994); Hayes-Roth (1995); Russell and Norvig (1995); Wooldridge and Jennings (1995); Franklin and Graesser (1996); Tecuci (1998).

In VREs (Virtual Reality Environments), agent based approaches are largely explored, and they are becoming one of the central pieces of such applications. In the next sections an introduction to agents' definitions and properties are presented (focused on VRE agents). Some of the main definitions for the "agent" term are presented, its properties and classifications, its modelling and implementation methodologies, as well as the main agent control architectures. Besides that, we also discuss some topics directly related to intelligent virtual agents, like perception and action in a virtual environment.

III-2-1. Agents: Definitions and Properties

Russell and Norvig (1995) define an agent as a system capable to perceive information from the environment through sensors, and to react through actuators. According to Tecuci (1998), an agent is a knowledge based system that perceives the environment where it is immerse (e.g. virtual world objects, other agents and also the user actions); reason in order to interpret perceptions; solve problems and determine actions; and finally acts over the environment elements in order to accomplish a group of tasks for which it was designated. As defined by Garcia and Sichman (2003), Intelligent Agents are computational characters that act following a script, directly or indirectly defined by a user.

Intelligent Agents can be characterized from a group of properties, which differentiate them from the traditional software systems. According to Wooldridge and Jennings (1995), they possess certain fundamental properties, like: autonomy (they can operate without human direct intervention or from the other agents, and they have some control about their own actions); social ability (they interact with other agents – humans/avatars or computational agents - or interact with the environment elements, depending of their necessity to solve problems, or even to help other agents); reactivity (they perceive the environment and they react to the modifications on it); proactive (they can exhibit a behavior driven by goals, making decisions and actions when they judge appropriate).

Franklin and Graesser (1996) list other properties: adaptability (ability that the agents possesses to acquire knowledge from their experiences and to adapt their behaviour according to the acquired knowledge); temporally continuous (the agent is a process that is continually in execution); mobility (the agent capacity to move from one computer to another in a computer network); flexibility (the agent ability to act even when your actions are not well described and detailed in predefined scripts); and personality (regarding to the agent that possess a personality and an emotional state).

In the virtual agents context, especially virtual humans, Thalmann and Thalmann (1994) indicate certain properties that should be considered in the agents' modelling: behaviour (the way the agent acts in the environment); perception (the agent observations, obtained from "physical" sensors); memory (represents the recovery process of what it was learned, especially based on associative mechanisms); emotion (defined as an effective aspect of conscience, a feeling state); conscience (state that characterizes sensations, emotions and desires); and freedom (characterized by the absence of restrictions in the agent choices or actions).

III-2-2. Classifications of Agents

In order to create a classification of Intelligent Agents, usually is suggested to adopt a scheme based on a subset of properties that they possess. A binary classification can be associated to agents, just considering a specific property (property present or not), or a multiple classification, adopting several of these properties. For example, following a binary classification, it can be stated that an agent is mobile or stationary, or, following a multiple classification, it can be indicated that an agent is mobile and with no adaptive capabilities, given the mobility and adaptability properties.

Besides that, other classification possibilities exist, according with: ***(a)*** the task executed by the agent (e.g. e-mail or information filtering, security agent); ***(b)*** the control architecture (e.g. reactive, cognitive, hybrid, based on mental states); ***(c)*** the environment where the agent is inserted (e.g. web, database, virtual environment); and ***(d)*** the type of knowledge that the agent possess (e.g. user's preferences and interests, business information, e-commerce transactions, etc).

Some authors propose specific agents classes or denominations. Franklin and Graesser (1996) present a wide classification, based on a biological taxonomy; firstly they divide agents into biological, robotic or computational. Then the computational agents are divided into artificial life agents or software agents, and the last ones are classified into tasks based agents, entertainment agents or virus agents.

Jennings and Wooldridge (1996) distinguish three classes of agents, according with the level of the agents' sophistication: "gopher" (they execute tasks based on predefined rules); "service performing" (they accomplish tasks from user requisitions); and "predictive" (they offer information or execute actions for the user, with certain autonomy degree). Besides that, they denominate users' agents, the ones that possess knowledge about the users' preferences and interests, and businesses' agents, the ones that hold information about the business (e.g. services, products, etc). Brenner et al. (1998) distinguish three categories, depending on the task the agent execute: informative (offer support to the user in information search from distributed sources); cooperative (act in the resolution of complex problems, through cooperation and communication with other objects, agents or external sources); and transactional (whose main tasks are to process and to monitor processes).

Reilly and Bates (1992) introduce the emotional agents (they possess an emotion and personality model); Maes (1994) describes autonomous agents (they accomplish tasks essentially in an autonomous operation mode); Sycara et al. (1996) introduce the interface agents (they interact directly with the user) and the task agents (they help user in the accomplishment of tasks, formulating plans to problem resolution).

Nwana (1996) relates the collaborative agents (they emphasize the cooperation with other agents); mobile agents (they are able to move from one environment to another); reactive agents (they act directly in response to environment stimulus, without reasoning); and hybrid agents (they possess characteristics of reactive agents and also reasoning capabilities); Rickel and Johnson (1997) mention the pedagogic agents (they act in educational environments); and Aylett and Cavazza (2000) define the virtual agents (they act in virtual environments and they possess a graphical representation associated to the agent), that can be classified as physical agents (emphasis is given to the agent's physical behavior in the environment) or cognitive agents (emphasis is attributed to the agent's cognitive abilities and to the interaction with the user of the system).

Garcia and Sichman (2003) present a taxonomy in which an agent is a special type of computational system and it can be classified according to some axes: cognitive, focus, activity and environmental. In the cognitive axis, the agents can act based on reasoning models of decision (cognitive) or based on reaction models that react to stimulus proceeding from the environment (reactive). In the focus axis, they emphasize physical similarities (structural) or behavioural similarities (behavioural) with humans. Related to the activity axis, the agents can act in an individual way (isolated) or with other agents (social). Finally, the environmental axis divides them into local agents (desktop agent) or in a network agent (internet agent).

Finally, Thalmann (1996,1999) proposes a classification focused on virtual human agents, based on the degree of autonomy of them in the environment: avatars, the user's representations in the environment; guided actors, when they are driven by a user through orders; autonomous actors, capable to have their own behaviour (autonomy of control); interactive and perceptive actors, that interact with the environment and communicate with other actors.

Table 2 presents a synthesis of the main classifications attributed to Intelligent Agents. This table lists some important features that can be used to classify them, together with the corresponding class.

Feature	**Classification**
Type of Agent Entity	Real (human, biological, physical robot) or Computational (artificial life, software)
Type of Human Similarity	Structural or Behavioural
Control Architecture	Reactive, Cognitive, Hybrid, Based on mental states, Based on model of emotions
Task	Transactional, Informative, Business, User's Agent, Interface's Agent
Autonomy Degree	Avatars, Guided, Autonomous, Interactive and Perceptive
Localization	Mobile, Stationary, Distributed
Type of Environment	Desktop (closed/local environment), Network (open/distributed environment), Pedagogical (educational environment), Virtual (virtual reality environment)
Type of Activity	Isolated or Social (Group, Cooperative or non Cooperative).
Type of Interaction	User interaction, Other Agents interaction, Environment interaction, Multiples Interactions

Table 2: Intelligent Agents Classification.

III-2-3. Virtual Environment Agents

In virtual environments, agents can act as users' assistants in the exploration of the environments and can help to locate information. They could establish with the user a verbal communication (in natural language, for instance) or non verbal communication (through movement, gestures and facial expressions). They can also execute actions in the environment, according to the user's solicitations. In this way, these agents need to have abilities that range from physical interaction with the environment and its objects to capabilities to make plans and decisions (cognitive behaviours), in order that they can act in an effective way in these environments and also interact with the users.

According to Aylett and Cavazza (2001), if we consider the different aspects of a virtual agent, a series of possibilities exists. First, an agent should possess a body, which should move in a physically convincing way through the environment. It should also possess some associated behaviours. Second, in order to make the agent seems to be responsible for the environment where he acts, a joining should exist between his behaviour and the state of the environment. Finally, the virtual agent behavioural repertoire can vary depending on the application: in some cases the focus is in the physical interaction with the environment and, in other cases, more focused in the cognitive behaviour, usually expressed through speech or natural language (synthesis and speech recognition, for instance).

So, it can be considered that a virtual agent needs both cognitive and physical behaviours. In this way, integration should exist among environment perception and reasoning, planning and action in this environment. This involves the use of Artificial Intelligence (AI) techniques, more specifically directed to manage cognitive behaviours, and the use of Computer Graphics (CG), more specifically directed to the physical behaviours (animation of the graphical structure of the agent and physical behaviour simulation). In the following sections, these aspects will be discussed. In the next section, Perception (of the environment), some approaches used to make agents perceive the environment they are immerse will be presented. In the other section, Action (in the environment), the aspects related to agent physical interactions are discussed. The methodologies used to define how the agents will behave during their interaction with the environment and to define how to make them accomplish their tasks are commented in the following section “Control Architectures”. After at the end of this section, the integration between all these aspects (perception, action and control) will be discussed.

III-2-3-1. Perception

So as to act in the environment, the virtual agents should perceive it. In accordance with Aylett and Luck (2000), this perception is an interaction between an agent and his environment. The perceived objects by the agent can be static objects (e.g. walls, furniture, obstacles, etc), other agents (static or moving agents), and dynamic obstacles (moving objects, like cars and doors), where the moving obstacles can be predictable (they move using predefined routes) or not. The mechanisms used to implement the virtual perception may vary from the projection of lines starting from the agent's front side searching for an intersection with an other object, up to biologically inspired models of vision systems.

The simplest mechanism of virtual perception is based on the projection of lines starting from the agent's front side. The perception consists of obtaining information about the objects that are intersected by this line. This approach tries to simulate infrared/laser based systems, commonly used in Robotics [Dudek 2000]. A similar approach is proposed by Villamil (2003a), where the autonomous agents' perception is based on the use of a perception area (Figure 3), considering a distance (di) and a perception angle (θi). This area composes the field of vision of the agents in the environment.

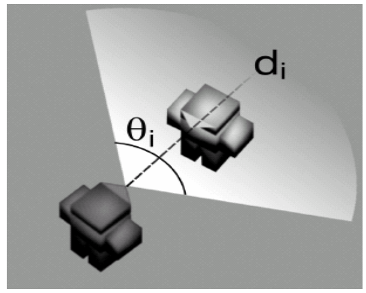


Figure 3: Agent perceptive region (Villamil, 2003a).

Another way of virtual perception simulation is based on an artificial vision model inspired in a biological model. This approach was used by Terzopoulos et al. (1994) in a virtual fish perception system. In this system, the agent's field of vision is projected to an artificial retina and algorithms process the pixels that are inside of the projected sphere, detecting the presence of elements of the visible environment.

An intermediate approach of perception consists of a mechanism based on messages about environment changes, allowing to consider all objects as static objects with their position well defined. This mechanism was used by the virtual agent STEVE (Rickel and Johnson, 1997), a pedagogical agent inserted in an environment based on Virtual Reality for training the use of naval equipments. In the STEVE case, the perception of the environment was based on the use of a communication data bus, which enables to update the agent's knowledge about the virtual world element positions. This data bus carry messages that describe all the modifications occurred in the environment, including changes in the position of the objects and also in terms of the object specific attributes. So, the agent maintains updated his symbolic model of the environment, from which he can reason and plan actions.

III-2-3-2. Action

From a more general point of view, the physical performance of a virtual agent is directly related to the movements of his graphical structure, what can involve gestures and even facial expressions. The complexity of this performance varies according to the sophistication of the agent's graphical structure, his activities in the environment, and his interactions with objects, other agents or users of the system.

In this process, we need to consider the geometric model of the agent's corporal structure and the animation of this structure, which is usually supported by toolkits (e.g. 3DStudioMax, Maya, TrueSpace or Cal3D), and involves approaches that emphasize the physical realism and/or that seek to provide to agents a more autonomous control. A simple form of animation involves the interpolation of the agent's structure through a set of control points, without considering specific animations, such as those of the arms and legs. This method is usually applied when the agent's physical performance in the environment doesn't involve sophisticated interactions, gestures or facial expressions.

In the case of more sophisticated interactions, specific Computer Graphics animation techniques are used. Among these techniques can be mentioned those based on key-frames, motion capture, procedural animation, animation libraries based on scripts, based on physics, based on rules and based on events. The first two techniques consider lower level aspects, related to the physical movements and to the correspondent visual realism; while the last two offer a high level of abstraction, based on the invocation of behaviors, such as walking or run. It is important to highlight that in the animation context, the term behavior differs from the adopted in it Artificial Intelligence (AI) domain. In Computer Graphics, this term refers to a predetermined animation, while in AI it involves mainly the occurrence of events in the environment that lead to a specific reaction (reactive behaviour).

In general, computer animation techniques involve a predefined repertoire of animations, which is applicable when the agent actions are repetitive, or in other words, when the agent doesn’t needs to have an autonomous control. According to Aylett and Luck (2000), considering that the virtual agents have a realistic corporal structure, an important aspect is to provide an autonomous control on it, with the agent's movements determined by internal activations and not only as a result of an external animation of their corporal surface. In this context, the virtual agent's physical actions involve not only geometric structure calculations for the animation, but also a high level autonomous control, associated to the semantics of the actions in the environment.

Efforts in Computer Graphics and Artificial Life research have been focused in order to improve virtual agent behaviours, evolving from predefined behaviours to more autonomous physical behaviours and actions in their environments. This approach is adopted by several behavioural animation researchers (Reynolds, 1987; Terzopoulos et al. 1994). According to Parent (2002), the behavioural animation consists of modelling elements that obey certain behaviour rules.

One of the first works in this area of behavioural animation was from Reynolds (1987), which simulated the behavior of virtual birds (Figure 4). According to this author, complex movements can be modeled starting from a group of simple rules, associated to each agent, such as maintaining a minimum distance from obstacles and a certain movement speed. Following this approach, Terzopoulos et al. (1994, 1999) presented the simulation of virtual fish groups (Figure 5), where each fish is endowed with perception (artificial vision), locomotion control (based on a mass-spring system used to propel the fish in the water) and with behaviors (based on a group of parameters, such as hunger degree and predators fear).

Figure 4: Behaviour simulation of birds (Reynolds, 1987).

Figure 5: Behaviour simulation of fishes (Terzopoulos, 1999).

Besides these works, related with virtual animals simulation, an intense research is focused in behavioural animation involving virtual humans groups (Tomaz and Donikian, 2000; Musse, 2000; Evers, 2002; Villamil, 2003a). In these populated environments, the control of virtual humans is made starting from a rule based system, which guides the behaviour of these virtual humans. This approach can be used to generate hierarchies of objectives that influence agents' movements. Schweiss et al. (1999) present a rules system that manages virtual humans' behaviour. In this system, the agents possess objectives as "to buy a train ticket" and "to take the train", following a hierarchy. According to Aylett and Cavazza (2001), the virtual agent's movement combines different voluntary movements, driven by goals, with involuntary movements, driven by the psychology and biology, involving also physical modelling and simulation. If the movement is generated from goals - sometimes described as task based animation - an global animation representation method combining the symbolic objectives level and the executed movement level usually is necessary.

We can also cite other specific approach based on behavioural animation, which is also based on physical models, used in the behavioural simulation of virtual humans' groups in emergency situations, proposed by Braun et al. (2003). This approach is based on a model that considers individuals' characteristics in emergent groups, consisting of a generalization of the physical model of crowds simulation proposed by Helbing (2000).

On the other hand, besides making the animation of the agent's corporal structure, the use of facial expressions is another important type of agents' animation, which is especially related to verbal or even non-verbal communication (expression of feelings and emotional state). It has been subject of intense research in conversational agents (Cassel, 2000). The verbal communication involves also synchronized speech and voice intonation, besides the agent's face animation techniques.

Concluding, as discussed above, virtual agents usually will need a complex control of their actions (including characters animation), that should consider the environment perception/interaction and also their objectives. This imposes the necessity to define a strategy and/or control architecture, that will be responsible for the agent's actions control. This control architecture will also be responsible for monitoring external events that happen in the environment and can affect the agent decisions and actions. Some classical autonomous and semi-autonomous agent control architectures are discussed in the next section.

III-2-3-3. Control Architectures

An agent’s architecture is a methodology adopted for the implementation of the cognition and decision process of this agent. Through this architecture, it is specified how an agent behaves during the interaction with the environment and how he acts in the accomplishment of his tasks. According to Corrêa (1994), in order to define an agent's architecture it is necessary to know the type of task this agent will accomplish, and also his role in the environment. The control architectures can be classified according with the mechanisms used by the agents to select an action [Viccari and Giraffa, 1996, 2001]. From an AI perspective, and considering the above classification principle, the agent control architectures can be classified in: reactivate, cognitive, hybrid and based on mental states. In the following sections, these architectures are presented.

It is important to highlight that we found different mechanisms used to control virtual agents' actions in their environments, depending if they are adopted by AI or CG domain researchers. The AI researchers tend to adopt approaches that vary from a "simple loop" of type stimulus-reaction, where it doesn't involve reasoning, until approaches based on mental states, where the agent's internal processing is described using a specific group of mental states (e.g. beliefs, desires, intentions and expectations). On the other side, the CG researchers tend to adopt approaches based on libraries of scripts which describe predefined behaviours. So, the adopted approach for control agents may vary depending on the degree of agent's autonomy in the environment, which is directly related with the application and the type of activity that this agent executes.

Reactive Architecture. The control architecture is denominated reactive or non-deliberative when the choice of the action to be executed is related strictly and directly with the occurrence of events in the environment. In this architecture, the agent's actions control is accomplished based on behaviours of type perception-action (or stimulus-response). The agent acts almost immediately to the moment he perceives some special “sign” coming from the environment, and his acts should be based only in this information provided by his perceptive sensors.

In this type of control architecture there is no explicit knowledge representation about the environment (Brenner et al. 1998). The agents' knowledge is “implicit” and it is manifested through their reactive behaviours. This can restrict the agent's autonomy and his capacity to learn and to improve the way he acts and behaves. Another outstanding characteristic of this architecture is the absence of past actions memory, resulting on the fact that the last actions doesn't influence directly to the next actions.

These agents based on a reactive architecture are denominated reactive agents or non-deliberative agents, and they don't possess high level reasoning or planning capacity. Because that, they are considered simpler entities than the cognitive agents. They are simple agents based on simple behaviours, with their actions defined according to the current state of the environment, or more specifically, according to their perception (sensorial input) of the environment state. In order words, events originated in the environment can fire agent actions.

As examples of this kind of reactive agents, some simple (autonomous) robots adopt controllers based on a reactive architecture. These robots possess a certain number of sensors that allow them to perceive the environment, and using this information they can activate their motors executing an action in the environment. Whenever a sensor detects an obstacle in the robot trajectory, the control architecture will generate an order to change the robot's movement direction or to stop it. This is the typical example of a "perceive- action" situation. In some situations, virtual autonomous agents can act like reactive robots, exploring the environment and avoiding obstacles.

Cognitive Architecture. The control architecture is denominated cognitive or deliberative when the choice of one action to be executed by the agent is obtained from a symbolic model of the environment in conjunction with a plan of actions used to achieve a goal. This architecture is based on the generation of a sequence of actions (plan) that are executed in order to reach a specific objective. These actions are based on the knowledge and some hypotheses the agent possesses about the environment, the other agents and the way he needs to act to achieve his main goal. In order to do that, an explicit knowledge representation of the environment is maintained, as well as the agent can keep in his memory a list of the last executed actions.

However, the cognitive architecture is typically unable to act fast and appropriately when faced to unexpected situations. It adopts the hypothesis that the world state remain static while the agent is executing his actions or processing some information to deliberate about the next actions. On the other hand, those architectures can explicitly represent goals and environment descriptions (e.g. environment map), being able to reason, plan, act and even to learn (adapt the internal knowledge base of the agent).

The agents based on a cognitive architecture are denominated, cognitive agents or deliberative agents, and they reason and decide about which objectives they should reach, what plans they should proceed and which actions should be executed in a certain moment. In this way, a cognitive agent executes an "intelligent action" when, possessing a certain objective and the necessary knowledge indicating that a certain action can conduct him to the accomplishment of this goal, he selects and executes this action.

The cognitive agents act based on their knowledge, because they have a knowledge base and high level reasoning skills. The agent knowledge base is directly related to the application domain and it is also related to the specific kind of interactions exist between the agents and the environment. Cognitive agents can also plan future actions based on the memory of past actions and accomplished tasks. So, cognitive virtual agents can represent in their memory information like: maps of the environment, specific predefined paths in the environment, predefined plans of actions used to accomplish specific tasks; but they can also reason in order to: establish a new route from one position to another one in the environment, answer a specific user question, find a solution for a specific posed problem.

Hybrid Architecture. The control architecture is denominated hybrid when the choice of one action is accomplished using a combination amongst the techniques used in cognitive and reactive architectures [Dudek, 2000; Heinen, 2001; Heinen, 2002a]. This architecture was proposed as an alternative to solve the main deficiencies of the two previously described architectures. The agent based on a pure cognitive architecture is typically unable to react fast and appropriately when faced to new and unexpected situations. The agent based on a pure reactive architecture is unable to discover new alternatives and execute complex tasks that are different from those accomplished by its initial objectives and behaviour, as he doesn't possess reasoning and planning abilities.

The main objective of a hybrid architecture is to build an agent including at least two subsystems: a cognitive system, that contains a symbolic model of the world, used in planning and decision making, and a reactive system, that is able to react to unexpected events that can happen in the environment. The hybrid agents are usually designed using a hierarchical architecture. The lowest level in the hierarchy is represented by the reactive system and it is used to acquire information (and react immediately if needed) from the environment, from the other agents or from other sources. The cognitive components, responsible for planning, goals determination and high level reasoning, are used in the highest levels of the hierarchy (Brenner et al. 1998). So, hybrid virtual agents can plan, and execute tasks and also react to different kind of situations.

Architecture based on Mental States. The main idea of an architecture based on mental states is to describe the agent's internal processing using a basic group of these states, like: beliefs, desires, intentions, expectation, among others. The BDI architecture (Belief, Desire and Intention) is an example of architecture based on mental states. Some authors consider BDI as a deliberative architecture, once this control architecture maintains a symbolic representation of the environment, expressed and tasks in terms of beliefs, desires and intentions.

According to Rao and Georgeff (1995), Belief, Desire and Intention represent, respectively, the information, the motivation and the agent's deliberative state. From an empirical point of view, the beliefs correspond to the information that the agent possesses about the environment (e.g. maps), desires represent possible future states of the agent (motivation – available plans/actions) and intentions are the next states that the agent probably will go during the execution of the selected plan (deliberation – steps on the plan execution).

The beliefs, according to Brenner et al. (1998), are the agent's vision of the world. Beliefs compose the agent knowledge about the environment, represented in an explicit form. For Rao and Georgeff (1995) it is the informative component of the system state, which is necessary to supply information about the probable state of the environment.

The desires represent the judgment of the future states (Brenner et al. 1998). The agent can plan to attain some specific states contained in his beliefs of possible future states. They represent the motivational state of the system (Rao and Georgeff, 1995). A desire is an abstract notion that indicates preferences on future states of the environment. It represents a situation or group of situations in that the agent would like to be.

The intentions are the chosen goals according to the agent's priority. They represent the deliberative components of the system (Rao and Georgeff, 1995), and they are used to decide the course of action the system should follow. It is the result of the choices, which takes the agent to an action.

In this context, the agent's practical reasoning involves repeatedly: update the beliefs considering the new environment perceptions, decide which plan options are available, filter these options in order to determine new intentions, and act based on these intentions. The main idea is to implement a mechanism to decide which action when executed will help the agent to approach his objectives, or in other words, decide which goals should be reached and how this will be done.

One example of use of this approach in virtual agents' cognitive modeling, with the purpose of implementing animated characters is proposed by Torres et al. (2003). The agents' reasoning system uses mental states, defined based on the BDI methodology, and drives the agents' autonomous behavior in an animation system.

Other Architectures. In the context of multi-agents systems, the AEIO methodology (Demazeau, 1995) considers the problem to be modeled as composed by four elements: *Agents*, *Environment*, *Interactions* and *Organization*. In order to specify these elements, it should be considered, respectively: the number of agents and their types (cognitive or reactive, for instance); the characteristics of the environment (usually dependent of the problem domain); the interactions among the agents; and the coordination and cooperation mechanisms among the agents.

Another methodology that considers the interactions among agents is KSI (Musse, 2000). This methodology was applied by Evers (2002), in a model used for virtual humans groups' simulation. This model creates crowds from complex individuals, that possess intentions, emotions and memories (individuals learn based on their past experiences). The virtual humans' grouping is made based on their common knowledge/experiences.

A higher level methodology was presented by Aylett and Cavazza (2001), where it is proposed to describe the agent's behavior structured in three levels: movements, actions and intentions. The movements correspond to the physical simulation (animation) in the virtual environment; the actions correspond to patterns of movements (scripts) necessary to execute specific actions; and intentions refer the agent's high level objectives.

Finally, in the Artificial Life approaches (AL), where, in general, the emphasis is to animals groups' behaviour simulation, the control architectures reflect behaviours based on stimulus and motor control, responsible for animate the body. However, according to Aylett and Cavazza (2001), identical sensor inputs can produce different behaviours, based on internal rules, which activate different parts of the behavioural repertoire.

III-2-3-4. Perception, Action and Control Integration

Considering an intelligent virtual agent, the "intelligence" of this agent is composed by several components, such as: perception, reasoning, action, learning, natural language communication, all integrated together (Aylett and Cavazza, 2001). So, an intelligent behavior involves to chose the best actions sequence, considering the agent's objectives and the present environment situation. Thus, \virtual agents involve cognitive levels (reasoning and planning), reactive levels (react to unexpected events) and physical levels (action - animation). One important aspect within respect to these three levels is how they can be integrated. In the previous sections, were presented approaches related to agents' perception, action and control architectures. In this section some works that explore the integration among the cognitive, reactive and physical levels will be presented.

According to Aylett and Cavazza (2001), a simple way of integration among these levels, involves the use of AI algorithms and animation primitives. It can be illustrated by the example of path planning and following (environment navigation). In this case, the planning techniques are coupled directly with the animation level. This technique consists of finding the best path between a starting point and a destination point, considering the existence of some obstacles in the environment. This technique was originally explored in autonomous robots' applications (Latombe, 1991) and now is being largely applied in virtual environments (Kuffner, 1998; Heinen 2002a; Heinen and Osório 2002b; Bandi and Thalmann, 1998; 2000). The A* (Nilsson, 1980) is one of the most important algorithms of path planning adopted in several IVRE implementations. Once the path from initial to destination position is defined, the animation of character (e.g. walk, run) is done in conjunction with his displacement along the predefined path.

An example of the use of path planning together with animation primitives is presented by Chittaro et al. (2003). This approach is used for a virtual agent's autonomous navigation, according with the user's objectives. In this application, the virtual agent is designated to help the user in the navigation into a virtual museum environment. From a list of places or objects of interest to be visited, the agent creates autonomously the appropriate path.

Another example of use of AI planning techniques was implemented in the agent STEVE [Rickel and Johnson 1997]. This agent acts as a tutor or instructor in a domain where the student is trying to learn procedures to operate a machine. In a cognitive level, is maintained the representation of a plan (sequence of actions), that guides his behavior in the environment. The planning formalism adopted allows the representation of the actions as a group of nodes linked by causal links (causal links denote when the effect of an action is condition for the execution of another action). The actions sequence is defined in a plan, which is assigned to a component responsible by executing it in a physical level. In this system, the task development and monitoring of actions execution are perception sources. Besides that, STEVE is an example of agent with an objective oriented cognitive component, which allows him to select his own action, based in his plan of actions.

Moreover, distinctions among the cognitive and physical levels are presented by Perlin and Goldberg (1996) and Thalmann and Thalmann (1995). Perlin and Goldberg introduce a distinction between low level animation control and high level behaviour, presenting the architecture IMPROV. This architecture is based on a scripts library and it supports high-level animation control of virtual actors. Thalmann and Thalmann (1995) in their work present a distinction between movement control and more complex patterns of behaviours.

Other examples of approaches that combine control architectures and animation primitives are present in some works developed at the CROMOS Lab. and they will be mentioned with more details in the next sections.

III-3. Groups and Crowds of Agents

Some constraints arise when we deal with crowds of virtual actors different from the modeling of virtual individuals. Indeed, there are different types of constraints depending on the different applications. First of all, the requirements for each crowd application are dependent on the nature of application. We use a classification of crowd application based on different purposes.

1. ***Entertainment*** applications include film productions and games. In this case, the groups have to be easily controlled and directed as well as individuals. Some examples are the recent films AntZ (1999) and Bugs Life (1999). The challenges include the simulation of lots of people as well as the automatic generation of individualities that are important in the visual aspects. In

addition, the easy control of groups represents a requirement of this kind of application.

2. ***Crowd motion simulation*** aims at evaluating the motion of lots of people in a constrained environment. It normally involves a simple visualization (embodiment of crowd) and a strong compromise with the numerical results in order to evaluate the environment. Some commercial software, e.g. SIMULEX (Thompson, 1995), are examples of this kind of crowd application. The main challenges in this case include a realistic method of collision avoidance, a strong connection with the environment and the compromise with the numerical and statistical results.
3. In order to ***populate collaborative virtual environments***, e.g. in a virtual reality system, the crowd requirements implies in providing a real-time simulation, as well as many aspects of interactivity in order to guide and direct crowds during the simulation. Yet, the facility to program crowds behavior represents an important challenge in collaborative virtual environments, where participants can create and interact with crowds. Some experiments have been made in the context of V.R systems, as presented in (Musse et al, 1998).
4. The last kind of crowd application concerns the ***behavioral modeling of crowds***. In this case, the goal is to provide autonomous or semi-autonomous behaviors that can be applied by self-animating agents which form the crowd. In order to deal with the constraints of having lots of actors (hundreds or thousands), this kind of application presents the following challenges: description of sociological and social aspects which arise in crowds, dealing with directed and emergent behaviors and connection with the virtual environment. Yet, needed optimizations and simplifications are required in order to be able to model and simulate crowd intelligence and decision ability.

III-3-1. Connection with the Virtual Environment

The connection between the virtual crowd and the virtual environment can happen at different levels. Indeed, as normally agents in crowd models do not have vision, they have to be informed about obstacles to avoid, objects to interact, path to get to a specific goal, an interesting location where the people can pass through, among others. This is the specific case of ViCrowd, which is presented in order to illustrate the connection between the virtual environment and virtual population. This environment data can be:

i) included in the script before the simulation;
ii) informed using a database associated to the virtual environment *(Farenc et al, 1999b)* or;
iii) defined during the simulation by the external controller.

A virtual environment can be represented by a complex scene with several objects, obstacles, and constraints. Depending on the level of detail of information transferred to the crowd, the groups can behave with more or less realism. Indeed, the simulation of crowds in ViCrowd is completely independent of the virtual environment. The connections between the groups and the environment is made as a function of the script language, where the IPs (interaction points) and APs (action points) can be defined, as well as the obstacles to avoid and the objects to interact.

The minimum information required by the crowd is the location of goals. Having this information, the crowd is able to walk from one goal to the next. In order to generate different trajectories between two locations, we use a spatial division of regions around the goals. Fig. 6 shows the graphical interface used to define the goals of the crowd and two goals specified to determine the crowd motion in a virtual city.

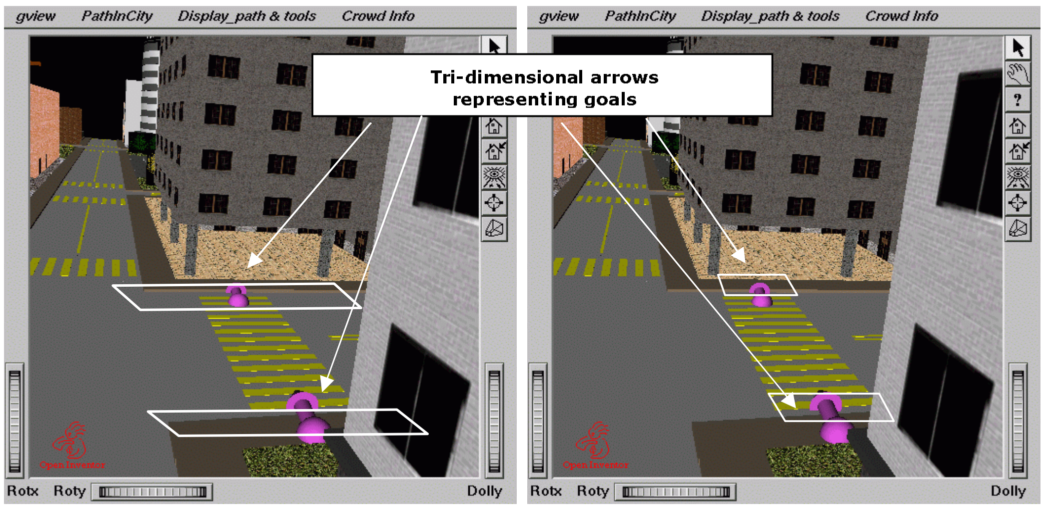


Figure 6: The arrows represent two goals of crowd. On the left: regions represent surfaces computed by the system. On the right: regions represent surfaces described according to the virtual environment (using a graphical interface).

Figure 7: On the left, a group walks through the goals defined as in Fig. 6 considering the default region sizes. On the right, the region size is determined using the graphical interface.

The region around the goals where the crowd is placed can be specified according to the virtual environment (Fig. 6 - right). Otherwise, the regions are defined using a pre-defined size (Fig. 6 - left). However, in order to improve the connection between the virtual environment and the actors, the sizes of regions have to be associated to the virtual model, then the realism of simulations can be improved. Fig. 7 shows two simulations. The first one uses the default size of regions, and only the goal locations have been defined (as shown in Fig. 6 - left). The second image presents a simulation where the regions have been defined according to the virtual model.

In the case of Fig. 6, on the left, regions are defined using the default sizes determined in ViCrowd. As a result, the agents do not respect the surface to walk (zebra crossings). If the user wants to fit more connection between the environment and the actors, more geometric information is required in order to define the size of region for each goal, then agents movements are restricted to the specified surface as shown in Fig. 6 on the right.

The obstacles specified in the script are taken into account by the collision avoidance methods. However, if objects are not specified, the crowd can not avoid collisions, as agents do not have vision.

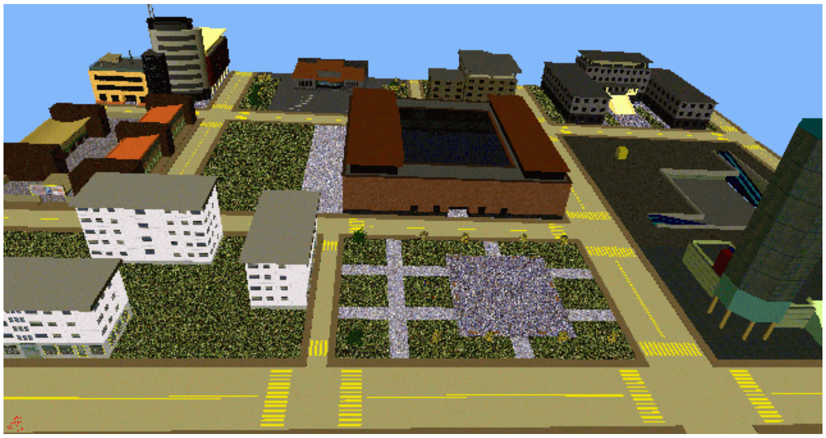


Figure 8: Image of the virtual city (Farenc et al, 1999a).

If there is a database of the associated virtual environment where the obstacles could be captured automatically as well as the walking regions, the realism of the simulation can be improved. In this case, the environment can be decomposed into regions where the crowd can walk. Then, the avoiding collision method developed in ViCrowd can take into account that the agents have only to walk through the specified goals and regions, avoiding locations outside those regions. Figure 8 shows the virtual city and Fig. 9 shows the geometrical information captured from the environment in order to provide information of regions to walk and objects to avoid. The database used in this example is the work presented in Farenc *(1999a)*.

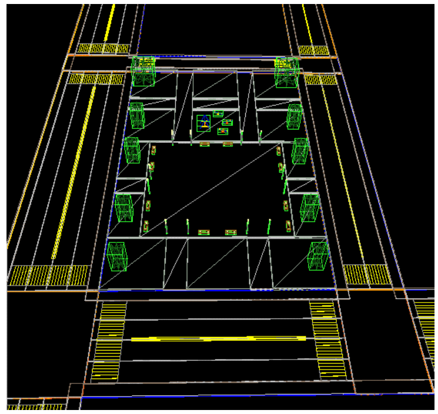


Figure 9: Image of system to capture information from the virtual environment (Farenc et al, 1999a).

When considering a large environment with several objects, IPs, APs, obstacles and regions to walk, the computer memory required to load the virtual environment, the geometrical information and the agents of the crowd can limit the frame rate of the simulation. A solution is the specification of geometrical data depending on the location of the crowd. For instance, if a crowd is in the park and nobody is in the supermarket, the objects to be avoided in the latter location do not need to be specified. In order to provide functions to manage this kind of dynamic information, the crowd in ViCrowd is able to deal with information sent by an external module during the simulation. The external information can be associated to the different aspects of the crowd information.

III-3-2. Connection with Intelligent Objects in the Environment

The main objective of intelligent objects, named here as smart objects, is to build a framework where the designer could model an object not only including the geometry but also information about specific parts to interact with and its functionality *(Kallmann and Thalmann, 1998)*. Then, instead of providing only geometrical objects which require more complex structures like vision and intelligence, smart objects are developed in order to provide interaction with non-complicated and non-complex agents (meaning agents which have not vision and can not perceive objects and their features).

In order to integrate ViCrowd and smart objects, we have defined a simple algorithm, which consist of changing the controller module that manages the agents when a smart object is perceived. For instance, if one agent perceives a smart object (e.g. an automatic door) and verifies that the region occupied by the door has to be used by the agent, before getting into the smart object region, the control of the agent is transferred to the smart object module.

The smart object module manages the interaction between the agent and the object, and then, returns the control of the agent to the crowd module. During the interaction with the object, the agent is disconnected from the crowd, and then the behaviours managed by ViCrowd are not considered. Fig. 10 shows the modelling of smart object features as well as one agent interacting with a desktop and being controlled by the smart object module.

Figure 10: On the left the smart object features modelling, and on the right the smart object module controls one agent (Kallmann and Thalmann, 1999).

Also, the smart object can be included in an action point (AP) in order to be performed by the crowd. The listing below shows an example of an AP specification including a smart object interaction with a lift and Fig. 11 shows an image of this simulation.

Example of an action point which specifies smart object interaction to be applied

```
AP_0        POS ( 11423.867188 40.000488 -5947.833984 )
            ROT ( -0.015588 -0.000002 0.999879 )
            REGION 0
            TYPE SHARED
            ACTION SOBJ so-Lift
```

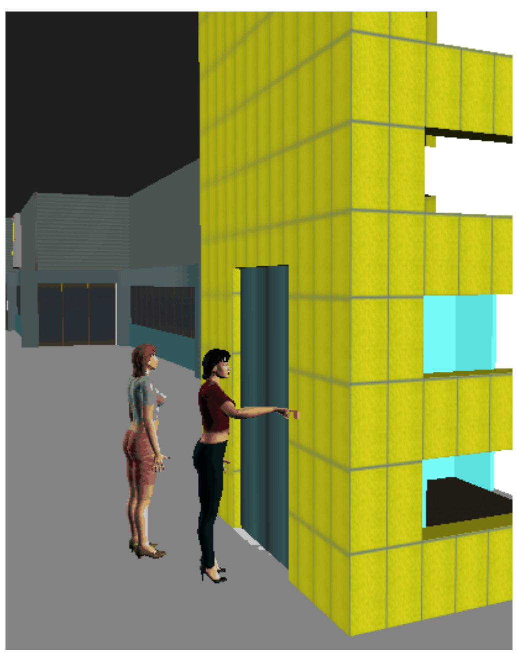


Figure 11: Simulation of agents interacting with a lift (Kallmann and Thalmann, 1999).

IV. Agents Group Behaviour Simulation in IVREs

Some authors have discussed how to simulate virtual crowds. Reynolds (1987) described a distributed behavior model for simulating *flocks of birds* formed by actors endowed with perception skills. In fact, the birds (or 'boids') maintain proper position and orientation within the flock by balancing their desire to avoid collisions with neighbors, to match the velocity of neighbors and to move towards the center of the flock. Reynolds's work shows realistic animation of groups by applying simple local rules within the flock structure.

Le Goff (1994) described an approach to create a behavioral model of groups formed by heterogeneous entities. His concept of groups is related to an association of individual behaviors and the management of internal resources as well as a decisional process inherent in the group entity.

Tu and Terzopoulos (1994) have worked on *behavioral animation* for creating artificial life, where virtual agents are endowed with synthetic vision and perception of the environment. The repertory of fishes' behaviors relies on their perception of the dynamic environment, and the fishes' reactions are not entirely predictable because they are not scripted.

Bouvier (1997) used particle systems adapted for studying crowd movements where human beings are modeled as an interactive set of particles. The motion of people is based on Newtonian forces as well as on human goals and decisions. They introduced the concept of "decision charges" and "decision fields" modeled by using notions of the so called decision charges of a person, interacting with a surrounding decision field in the same way an electric charge is influenced by an electric field.

Brogan and Hodgins (1997,1998) have used dynamics for modeling the motion of groups with significant physics. They reproduced movements of legged robots, bicycle riders and point-mass systems based on dynamics, considering an algorithm to avoid collisions, which determines the desired position for each individual, given the locations and velocities of the visible creatures and obstacles. Indeed, a perception model to determine creatures and obstacles visible to each individual in the group precedes the displacement algorithm.

Specifically concerning Intelligent Virtual Environments (IVE), some authors have discussed methods for building informed virtual scenes focused on urban context [Musse 2001, Donikian 1999, Farenc 1999, Thomas 2000 and Thomas 2001]. Donikian (1997) have studied the animation and simulation of autonomous vehicles in urban environments. Farenc (2000) described a database model in a framework to simulate virtual humans. Ulicny (refxx) presented a model to provide virtual reality training in emergency situations.

Musse & Thalmann (2001) proposed ViCrowd: a model to describe crowd behaviours allowing the virtual agents to interact among them as well as with the VE. ViCrowd's goal is to simulate groups of autonomous agents endowed with different levels of autonomy. This model is based on a hierarchical model to describe crowds with different levels of control: from guided to autonomous ones (Musse & Thalmann 2001). The behavior of crowds is based on rules dealing with the information contained in the groups of individuals.

The groups of virtual humans can be endowed with more complex abilities in order to apply other behaviours called complex behaviours. They can include *sociological behaviours*, responsible for the individual and group effects arisen as a function of the inter-relationship between agents.

In our approach [Musse, 2001], the crowd information is specified by defining the knowledge, status and intentions associated with the groups. This is followed by the creation of virtual humans based on the groups' behaviour information. The individual status and intention deal with information used in the sociological behaviour to provide rules to simulate sociological effects by changing individual and group parameters as a function of their inter-relationship. This section is concerned with an example of sociological model. Therefore, we have defined:

i) A a virtual human from the group GA and
ii) B a virtual human from the group GB.

When A and B are created in the initial population, they have the status and intentions defined as follows: (exemplified for A)

A.Status = Relationship(A, GA),A.Emotion,A.Leadership

A.Emotion = GA.Emotion
Relationship(A,GA)=Normalised_value $\sum_{i=0}^{N} Agents_N (GA).Emotion > A.Emotion$

The relationship between A and GA is described through a normalised value of the sum of the emotional status of all agents from GA which are higher than the emotional status of A.

Where, *N = number of agents of GA and*
Agents $_N$(GA) represents all the agents of GA

A.Leadership = random([GA.Leadership-0.2, GA.Leadership+0.2)

The level of leadership of A represents a random value but following the GA tendency

A.Intention = A.Communicative, A.ChangeGroups, A.BeLeader
A.BeLeader = A.Leadership > 0.7? TRUE
A.Leadership <= 0.7? FALSE
A has the intention to be a leader when the A.Leadership is higher than 0.7
A.ChangeGroups = random(TRUE, FALSE)
A has the intention to change groups depending on a randomly parameter

A.Communicability = GA.Communicability *A communicability depends on the GA ability to communicate (group status)*

Using these parameters, some sociological effects have been modelled.

i) A can ***change from group*** GA to group GB if,

i.1) Relationship(A,GA) < Relationship(A,GB): *if the relationship between A and GA is smaller than between A and GB.*

i.2) A.ChangeGroups = TRUE: *if A has the intention to change groups.*

i.3) A.BeLeader = FALSE: *if A is not the leader of GA.*

As a result, ***A changes from GA to GB****.*

If A.Leadership > All_Agents(GB).Leadership, then: :

(if the leadership value of A *is greater than the leadership value of all agents from* GB, then ***A will be the new leader of* GB.)**

GB.Leader = A Relationship(A, AllGroups) is recomputed A.BeLeader = TRUE A.Emotion = increased

The other agents from GB will have a new tendency to follow the new leader. This represents the changes of group parameter consequent to individual changes.

GB.Emotion = A.Emotion GB.Communicative = A.Communicative A.Memory = GB.Memory

Else **(*A is not the new leader of* GB)**

Then, some parameters of virtual human A change according to the parameters of its new group GB:

A.Emotion = GB.Emotion Relationship(A, AllGroups) is recomputed Agent.BeLeader = Agent.Leadership > 0.7? TRUE Agent.Leadership <= 0.7? FALSE A.Communicability = GB.Communicability A.ChangeGroups = random(TRUE, FALSE)

ii) The emotional status can also be changed when one agent meets another. Only when both have a high value for the domination parameter, one virtual human is chosen randomly to reduce its emotional status. In this case a ***polarisation*** between two leaders will follow. In any other event, both agents should assume the highest emotional status between them.

The sociological effects modelled in these rules are:

i) *grouping* of individuals depending on their inter-relationships and emotional status (changing of groups) and the *domination* effect (arising of new leaders);

ii) *polarisation* (between two leaders) and the *sharing* effects as the influence of the emotional status and domination parameters (agent parameters change because of new leaders); and,

iii) *adding* (group parameters change because of a new leader)

Case Study: Sociological Behaviours

We have applied our model to simulate a sociogram, which means a sociological graphic representing one population, its relationships, and the different levels of domination. Fig. 12 shows an example of a sociogram.

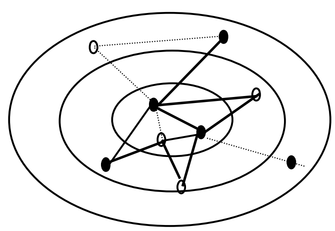


Fig. 12: A sociogram

The filled circles represent the women and the empty circles the men. The dashed lines represent a relationship of hostility and the full lines represent a friendly relation. In addition, this graphic represents a certain hierarchy in the group, as the individuals in the centre of the sociogram have more relationships as compared to the others in the external circles.

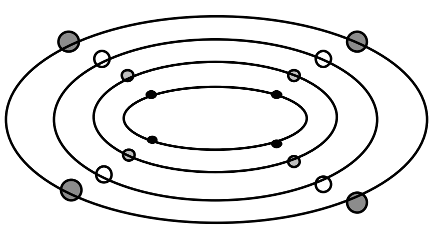
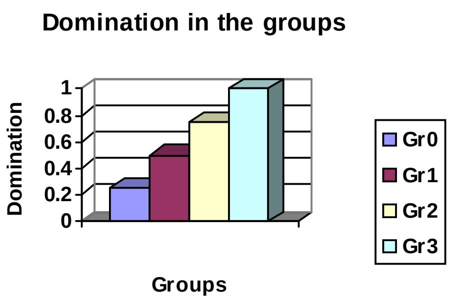


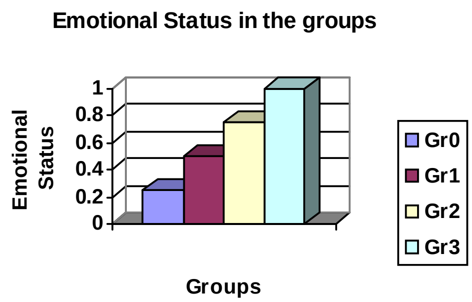
Fig. 13: The sociogram in our case

In our example, we used four groups with different behaviours and parameters (does not include the gender parameter). The agents walk on the limit of the circles, like in the sociogram and these circles are delimited by points, which represent the goals for each group, as we can see in Fig. 13.

Two simulations have been made where the values for crowd domination and emotional status have been described according to the next Figures.



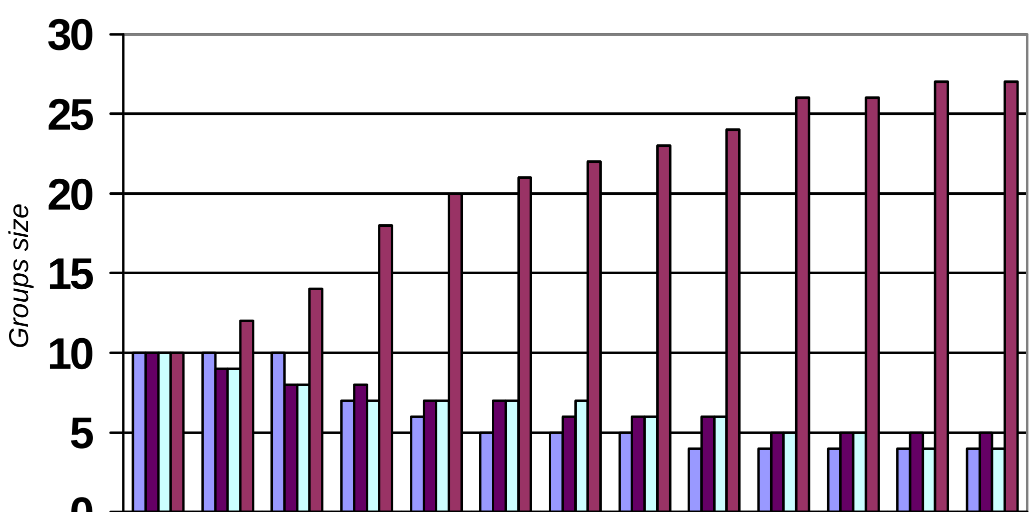
Values for domination distributed among 4 groups



Values for Emotional Status distributed among groups

Figure 14: Crowd domination and emotional status distribution.

Then, depending on the affinity and repulsion effects generated during the simulation using the data described in Graphics 8-1 and 8-2, agents can change groups in order to be part of a group with high emotional status and domination values. Graphic 8-3 shows the variation of groups' size as a function of simulation time. We can observe that the agents try to go to Gr3 in order to have higher levels for emotional and domination status.



	0	50	100	150	200	250	300	350	400	450	500	550	600
Gr0	10	10	10	7	6	5	5	5	4	4	4	4	4
Gr1	10	9	8	8	7	7	6	6	6	5	5	5	5
Gr2	10	9	8	7	7	7	7	6	6	5	5	4	4
Gr3	10	12	14	18	20	21	22	23	24	26	26	27	27

Time

Figure 15: Variation of groups' size as a function of simulation time.

Figures 16 and 17 present some images of simulation.

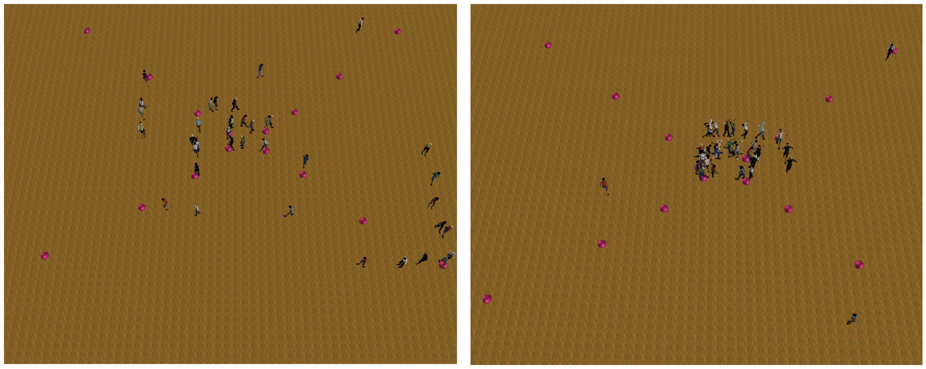


Fig. 16: On left, 40 agents are initially distributed in 4 groups in a sociogram. On right, after 600 frames of simulation, the major part of agents goes to the centred level in the sociogram.

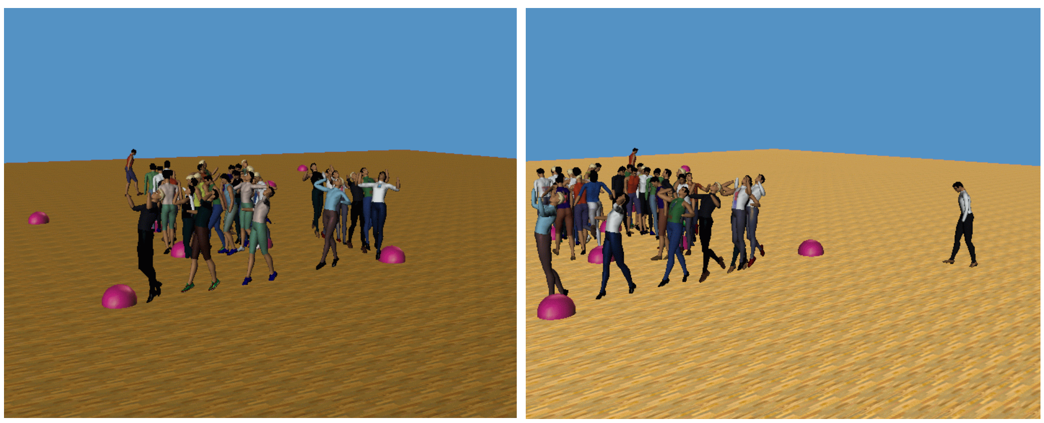


Fig. 17: Agents in the centre of diagram have higher level of emotional status than the agents in the other groups.

V. Applied IVREs

In this section, some of main examples of IVREs are presented. The objective is to present a general view of the area through the description of related research works.

A virtual intelligent agent, STEVE (*Soar Training Expert for Virtual Environment*), which acts as guide in training sessions is proposed by Rickel and Johnson (1997). The training is made in a three-dimensional interactive environment. The agent monitors trainees' progress and provides guidance and assistance. Moreover it interacts with simulations of objects in the environment. Figures 18(a) and 18(b) present the STEVE agent in the 3D environment.

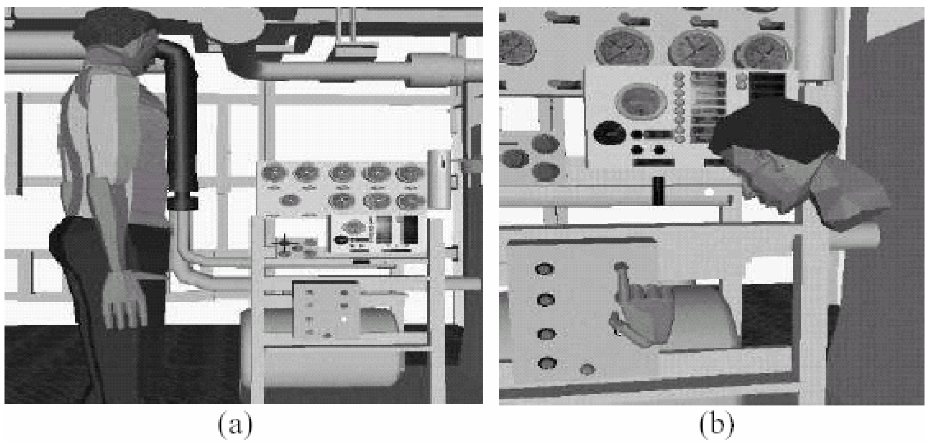


Figure 18. (a) Steve agent watching a machine; (b) Steve pressing a button [Rickel 1997].

Bersot et al. (1998) present Ulysses, a conversational agent in a virtual environment. The agent helps the users during navigation in the environment and accepts natural language commands to manipulate objects. A speech synthesizer is used to response to the user's solicitations. Figures 19(a) and 19(b) show the agent moving around the environment.

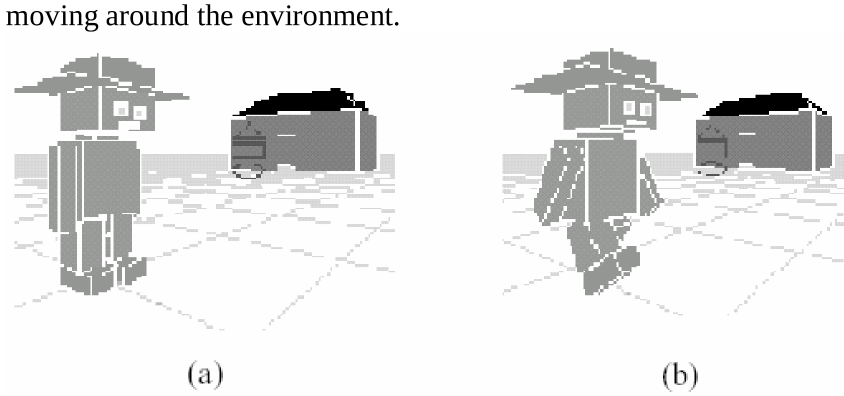


Figure 19. Ulysses moving in the environment [Bersot 1998].

A virtual agent, George, who guides the users to the relevant places in a virtual university, according to the user's information interest, and presents appropriated multimedia documents, is proposed by Panayiotopoulos et al (1999). The users can communicate with the agent through an interface based on line command. Figures 20 (a) and 20 (b) present the communication interface among the users and the agent, and the agent moving around the environment.

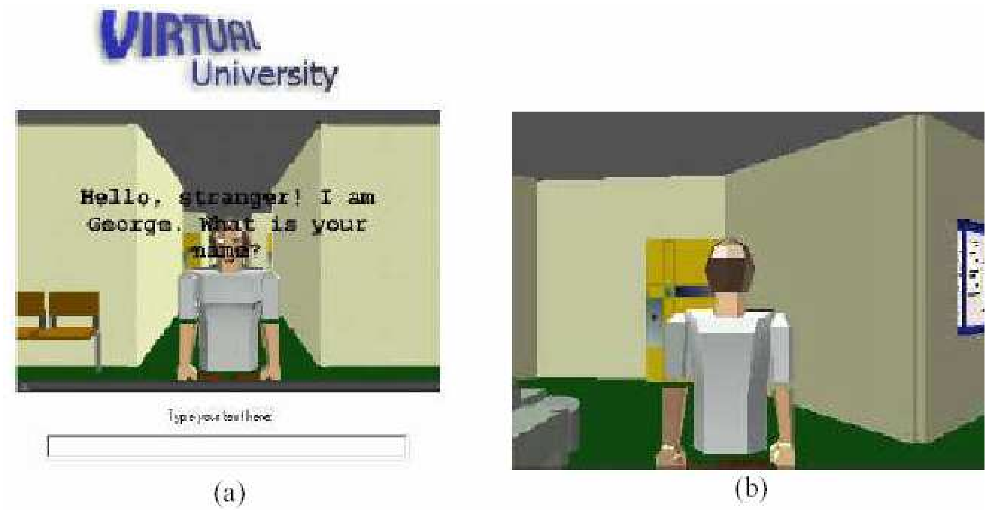


Figure 20. (a) Communication interface (b) Agent moving in the environment. [Panayiotopoulos 1999].

Noll et al. (1999) present the GuideAgent, a virtual agent which possesses the following main tasks: navigate in a three-dimensional environment; provide information about the environment, its users and objects; and help the users in the environment exploration. The agent and users are graphically represented in the environment and the agent can model the user's preferences and behaviors.

A virtual theater is presented by Nijholt and Hulstijn (2000). The environment has been built using VRML (Virtual Reality Modeling Language) and it can be accessed through World Wide Web (WWW). In the environment, the users can navigate, visiting rooms and observing pictures, and they can interact with a virtual agent called Karen. The agent possesses information about theater sessions, actors, and tickets. The communication between the agent and the user is made through the natural language. During the dialog with the user, it can present different facial expressions. Figures 21 (a) and 21 (b) show the external theater interface and the agent virtual, respectively.

Figure 21. (a) Theater entry (b) Virtual agent [Nijholt 2000].

The "Lokutor" system, proposed by Milde (2000), is a three-dimensional environment where a presentational agent acts as a sale assistant. The main task of the agent is to present cars to the user. The agent is able to convey information about the functionality of the car (e.g. how to open a door, how to open the tank lit, what type of fuel the car takes, the size of the space in the trunk etc). The user is able to interact with the agent by natural language commands. Information given by the agent is presented to the user in synthetic spoken natural language. Figures 22 presents the sequence of actions of the agent during the presentation of the car to the user.

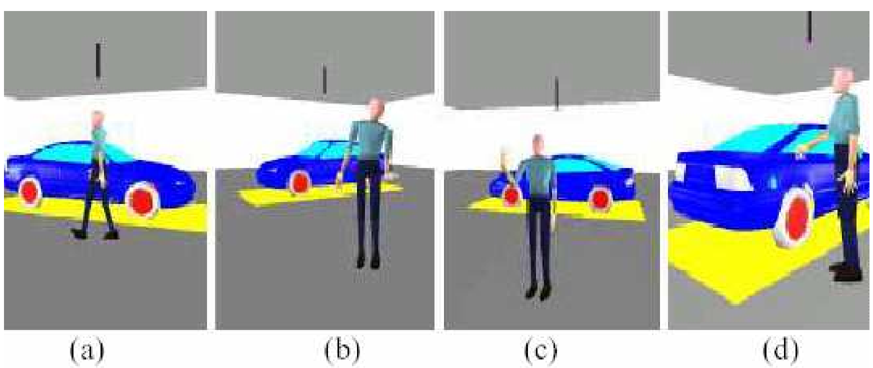


Figure 22. Agent presenting the car to the user [Milde 2000].

Anastassakis et al (2001) present an environment inhabited by agents which explore it, locate specific items, interact with others agents, and response to users' instructions (e.g. go to the specific place). In the presented scenario, a virtual library made using VRML, two agents, a librarian and a library client, interact. The client is looking for a book. In order to get it, it approaches the librarian and asks for it. Then, the librarian sets off and starts looking for the requested book among the library's shelves. If the book is available, the librarian brings it to the client. Figure 23 shows the virtual library interface, with the interaction between the librarian and the library client.

Figure 23. Interaction between the librarian and library client [Anastassakis 2001].

A methodology to navigate and explore three-dimensional virtual environments is described by Frery et al. (2002). Avatars are used as interactive guides, which have navigation strategies based on information about the environment. The avatar's intelligence is represented by its behaviors and knowledge about the environment and user. Based in this knowledge, collected using forms, the environment structure and the avatars graphical representation are modified.

The application of an AVI in the treatment of hyperactive children is presented by Rizzo et al. (2002). The environment consists of a virtual three-dimensional classroom where an avatar represents a teacher and there are some static and moving objects in the environment**.** An immersion system (HMD - *Head Mounted Display*) allows to monitor the movements of the children, detecting their view field. The collected data are used to analyze the children behavior. Figure 24 present the virtual classroom interface.

Figure 24. Virtual room interface [Rizzo 2002].

Moreover, Rizzo et. al. (2002) use a virtual office with several dynamic objects (e.g., phones playing, computers and avatars), to verify cognitive aspects of the users (e.g., memory, perception, and attention). One of the avatars represents the office boss and he can request to the user to do a specific activity in the environment. The HMD equipment is used to monitor the users' movements during the activities execution. Moreover, the environment is modified and the perception aspects of the user in relation to these modifications are collected. Figure 25 shows the several configurations of the virtual office.

Figure 25. Configurations of the virtual office [Rizzo 2002].

A three-dimensional and adaptive virtual environment is presented by Chittaro and Ranon (2002, 2002a). In the environment, which consists in a virtual store, the users can navigate and obtain information about the products. The information about the user's characteristics and interests is used to customize the environment. This data is collected using forms and monitoring the user's actions in the environment (e.g., visualized products, buys products). Walking objects help the user during the navigation and localization of specific products. Figures 26 (a) and 26 (b) represent the environment adaptations and the walking objects.

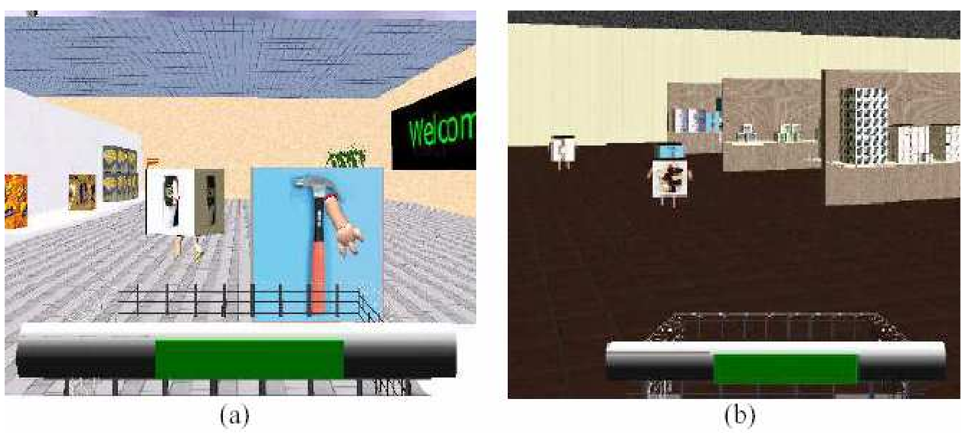


Figure 26. Environment adaptations and walking objects [Chittaro 2002, 2002a].

In a more recent work, Chittaro el. al. (2003) present a virtual agent which helps the user during the navigation in a virtual museum. From places or objects descriptions of user's interest, the agent creates the appropriated trajectory. Moreover, the agents can stop during the trajectory and present each place or object. Figures 27 (a) and 27 (b) show the agent in an explanation about an object and a created trajectory, respectively.

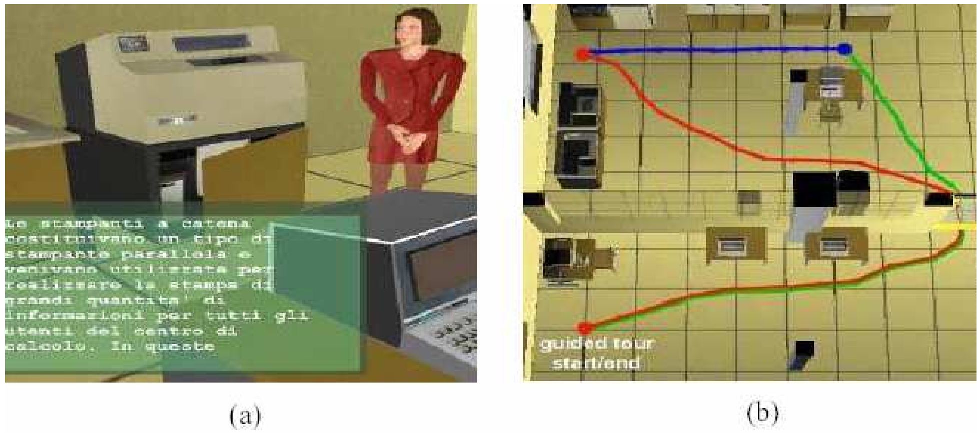


Figure 27: Virtual museum navigation: (a) Agent in an explanation about an object; (b) created trajectory [Chittaro 2003].

The IVEs also has been used as tool to experiment robotic techniques. Heinen and Osório (2002b) use a virtual environment (Figure 28) to simulate a hybrid control system for moving autonomous robots. In order to validate the proposed system, a simulator was implemented. This simulator allows using different models of three-dimensional environments and applying several sensorial and cinematic models to the robots.

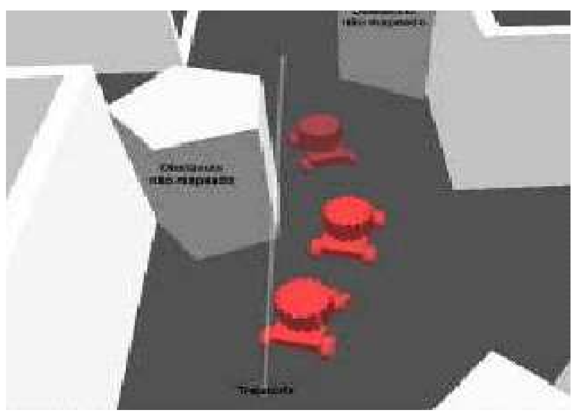


Figure 28. SimRob simulator [Heinen 2002b].

VI. Applied IVREs: Practical examples

In this section are presented some examples practical applications of IVREs developed in our research group – Graphit Group[12] (Computer Graphic and Vision Research Group). The practical applications are divided in two groups: i) AdapTIVE applications – This is an adaptive virtual environment served by an autonomous agent; ii) ViCrowd and CROMOS Research Project[13] – These are research and applications developed in order to study and implement Simulations of Groups of Virtual Humans (Virtual Crowds / Crowds Modelling and Simulation).

VI-1. AdapTIVE

The AdapTIVE is an intelligent virtual environment, which has its structure and presentation customized according to users' interests and preferences (represented in a user model) and in accordance with insertion and removal of contents in this environment [Santos 2004, 2004a, 2004b]. An intelligent agent assists users during navigation in the environment and retrieval of relevant information. This system can be applied in different application domains, as for example, distance learning, e-commerce and entertainment (games).

In this environment, there is support for two types of users: information *consumer* (e.g., student) and information *provider* (e.g., teacher). The users are represented by avatars, they can explore the environment searching relevant content and can be aided by the *intelligent agent*, in order to navigate and to locate information. The contents also posses a model and are grouped according to its knowledge areas. The consumer, provider, and content models are used in the environment adaptation. This adaptation involves the customization of the environment presentation and structure, where the contents that correspond to the areas of major user's interest are placed, in a visualization order, before the contents which are less interesting (easier access).

The user models (consumers and providers) contain information about the user's interests, preferences and behaviors. In order to collect the data used in the composition of the model, the explicit and implicit approaches are used. In the explicit approach, a form is used to collect fact data (e.g., name, gender, areas of interest and preferences for colors). In the implicit approach, the monitoring of user navigation in the environment and his interactions with the agent are made. Through this approach, the environment places visited by the user and the requested (through the search mechanism) and accessed (clicked) contents are monitored. These data are used to update the initial user model. The explicit approach is adopted to acquire the user's preferences compounding an initial user

---

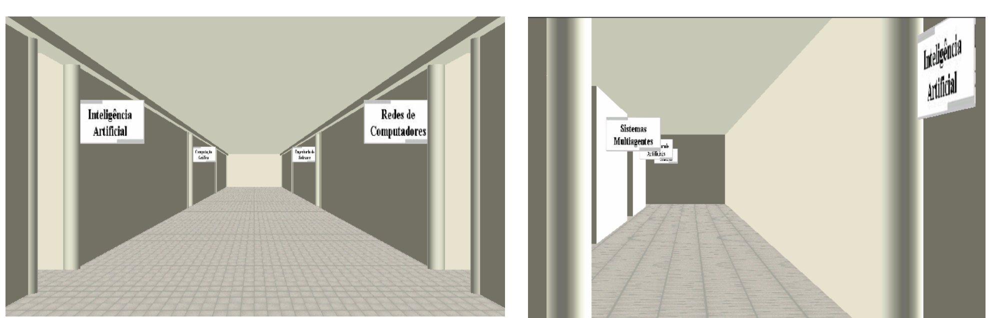
[12] Graphit Group – http://inf.unisinos.br/~graphit/
[13] CROMOS Lab. – http://inf.unisinos.br/~grapjit/cromoslab/

model and the implicit one is applied to update this model. A user model manager module is responsible for the initialization and updating of user models.

The contents are added or removed by the provider through a content manager module and stored in a content database. Each content contains a content model, containing the following data: category (among a pre-defined set), title, description, keywords, type of media and corresponding file. The *provider*, aided by an automatic content categorization process, acts in the definition of this model.

The representation of the contents in the environment is made by three-dimensional objects and links to the data (detailed description, e.g., text document, web page). An environment generator module is the responsible for the generation of different three-dimensional structures that form the environment and to arrange the information in the environment, according to the user and content models. Moreover, this module sends to the agent the information about the user models which are interacting in the environment and information about the contents and its positions. This way, the agent possesses sufficient information to help the users.

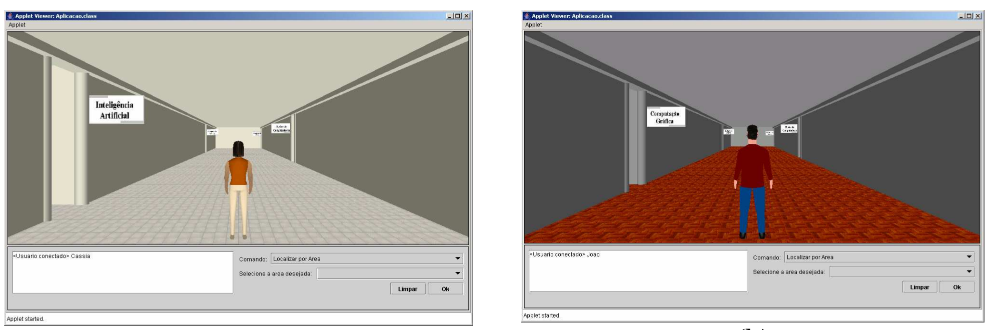
In order to validate the AdapTIVE, a prototype of a distance learning environment, used to make educational content available, was developed. In the prototype, a division of the virtual environment is adopted according to the areas of the contents. In each area a set of sub-areas can be associated. The sub-areas are represented as subdivisions of the environment. In the prototype the following areas and sub-areas were selected: Artificial Intelligence (AI) – Artificial Neural Networks, Genetic Algorithms and Multi Agents Systems; Computer Graphics (CG) – Modeling, Animation and Visualization; Computer Networks (CN) – Security, Management and Protocols; Software Engineering (SE) – Analysis, Patterns and Software Quality. A room is associated to each area in the environment and the sub-areas are represented as subdivisions of rooms. Figures 29 (a) and 29 (b) show screen-shots of the prototype that illustrate the division of the environment in rooms and sub-rooms. In screen-shots, a system version in Portuguese is presented, where the description “Inteligência Artificial” corresponds to “Artificial Intelligence”.



(a) (b)

Figure 29. (a) Rooms of the environment; (b) Sub-rooms of the environment.

According to the user model, the reorganization of this environment is made: the rooms that correspond to the areas of major user's interest are placed, in a visualization order, before the rooms which contents are less interesting. The initial user model, based on explicit approach, is used to structure the initial organization of the environment. This involves also the use of avatars according to gender of user and the consideration of users' preferences by colors. Figures 30 (a) and 30 (b) represent examples of initial adaptations of the environment structure. According to the Figure 30 (a), the user (female gender) has interest by Artificial Intelligence area and has preference by clean colors. In Figure 30 (b) the user (male gender) has interest by Computer Graphics area and has preference by dark colors. As the user interacts with the environment, his model is updated and changes in the environment are made. After n sessions (time window), for each area, the evidences of interest (navigation, request and access) are collected, the rules are applied and CFs are calculated. By sorting the resulting CFs, a new area ranking is made and, this way, the environment is restructured. It must be addressed that each modification in the environment is always suggested to the user and accomplished only under user's acceptance.



(a) (b)

Figure 30. (a) User with interest in Artificial Intelligence; (b) User with interest in Computer Graphics area.

On the other side, in relation to contents in the environment, the provider model is used to indicate the area (e.g., Artificial Intelligence) that the content being inserted belongs, and the automatic categorization process indicates the corresponding sub-area (e.g., Artificial Neural Networks), or either, the sub-room where the content should be inserted. In this way, the spatial disposal of the content is made automatically by the environment generator, on the basis of its category. In the prototype, thirty examples of book descriptions, for each sub-area, had been collected from the Web, and used for learning and validation of the categorization algorithm. In the stage of learning, experiments with binary and multiple categorizations had been carried through. In the binary categorization, a tree is used to indicate if a document belongs or not to the definitive category. In the multiple categorization, a tree is used to indicate the most likely category of one document, amongst a possible set. In the experiments, the binary categorization presented better results (less error and, consequently, greater recall and precision), being adopted in the prototype. In this way, one book can be categorized (and placed) in more than one category at the same time. For each sub-area, the rules obtained from decision tree (C4.5) were converted to rules of type IF – THEN, and associated to content manager module.

In relation to communication process between the agent and the users, they interact by a dialog in pseudo-natural language. The user can select one request to the agent in a list of options, simplifying the communication. The agent's answers are showed in the corresponding text interface window and synthesized to speech.

Figures 31 (a), (b), (c) and (d) illustrate, respectively: a request of the user for the localization of determined area and the movement of the agent, together with a 2D environment map, used as an additional navigation resource; the localization of an area by the agent; the user visualization of a content; and (d) the visualization of details of the content, after selection and click in a specific content description.

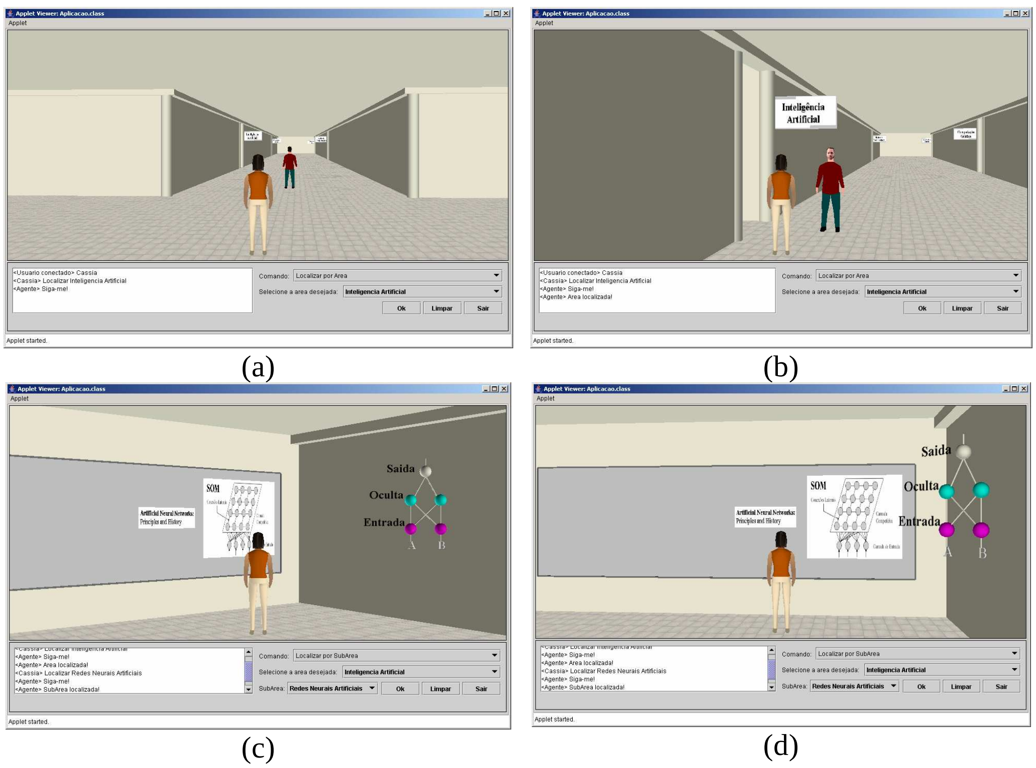


Figure 31. (a) Request of the user; (b) Localization of a area by the agent; (c) Visualization of contents; and (d) Visualization of content details.

AdapTIVE was also tested in other type of applications. It was also implemented a virtual bookstore using AdapTIVE [Santos 2004b], where books are automatically categorized, organized and placed in the environment, and the user preferences are defined and updated in order to present to him a customized vision of the virtual bookstore.

VI-2. Virtual Humans Simulation

The aggregated motion is both *beautiful* and *complex* to contemplate. *Beautiful* due to the synchronization, homogeneity and unity described in this type of motion, and *complex* because there are many parameters to be handled in order to provide those characteristics. History reveals a great amount of interest in understanding and controlling the motion and behavior of crowds of people. Psychologists and sociologists have studied the behavior of groups of people during several years. They have been mainly interested in the effects occurring when people with the same goal become one entity, named crowd or mass. In this case, persons can lose their individualities and adopt the behavior of the crowd entity, behaving in a different way than they were alone [Benesch 1995].

The mass behavior and motion of people have also been studied and modeled in computers for different purposes. An application aims to simulate the ***motion of crowds*** providing the evacuation of people in complex environments, for example, in a football stadium or in buildings.

The goal of this case-study is to present a model for studying the impact of the agents´ individual characteristics in emergent groups obtained on the evacuation efficiency and its dependence on local interactions. The starting point for the following discussion is the physically based model of crowd simulation proposed by Helbing (2000), which is generalized in order to deal with different individuals and group behaviors.

In this work, the groups´ behavior is attained as an emergent function of local interactions between individuals. We generalized the Helbing model in order to include different individualities in the particle systems as well as group behaviors.

VI-2-1. The Helbing Model

Helbing (2000) proposed a model based on physics and socio-psychological forces in order to describe the human crowd behavior in panic situations. It uses a particle system where each particle *i* of mass m_i has got a predefined speed v_i^o**,** i.e. the desired velocity, in a certain direction $\vec{e}_i^{\,o}$ and to which it tends to adapt its instantaneous velocity $\vec{v}_i$ within a certain time interval τ_i (1st term of Equation 1). Simultaneously, the particles try to keep a velocity-dependent distance from other entities *j* and walls *w* using interaction forces $\vec{f}_{ij}$ and $\vec{f}_{iw}$ (2nd and 3rd term of Equation 1), respectively. The change of velocity in time *t* is given by the dunamical equation:

$$m_i \frac{d\vec{v}_i}{dt} = m_i \underbrace{\frac{\vec{v}_i^o(t)\vec{e}_i^o(t) - \vec{v}_i(t)}{\tau_i}}_{1^{st}} + \underbrace{\sum_{j \neq i} \vec{f}_{ij}}_{2^{nd}} + \underbrace{\sum \vec{f}_{iw}}_{3^{rd}}$$

(Eq. 1)

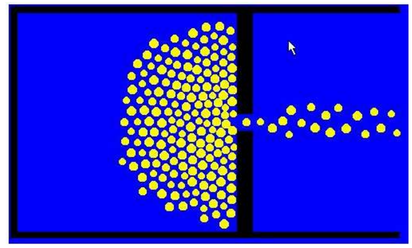


Figure 32. An image of Helbing´s Model Simulation

This model generates realistic phenomena, as arcs formation in the exit (Fig. 32) and the increasing evacuation time with increasing desired velocity as described by Helbing (2000).

VI-2-1. The Generalized Model

The main motivation for our model is the fact that different people can react in different ways depending on their *individual characteristics* and on *group structure*. For instance, an adrenaline maniac is less affected in panic situation than an always concerned being, who probably stops walking all of a sudden and thus interferes in the crowd dynamics as a whole. Furthermore, depending on the *group structure*, the individual action can change because the agent is part of a group, e.g. returning to the dangerous place in order to rescue a member of that group. Situations like those motivated our work, in order to enrich the simulation beyond a scenario where agents react only as individual and homogeneous particles.

The first contribution of our work is to attribute individuality to each agent, and thus allows the model to deal with different agent behaviors generated as a function of individual parameters. We describe below the parameters treated in order to define our virtual agents. A disscussion about how group behaviors are generated depending on the interaction between individuals is also presented. This second fundamental aspect in this model is related to the possibility of grouping people. In this case, agents are able to form groups which cause them to change their individual behavior as a function of emerged group structure.

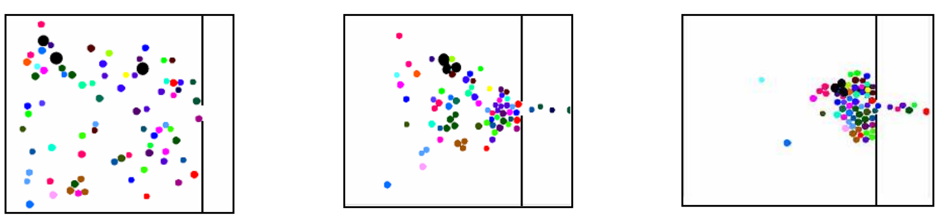


Figure 33. Sequence of images representing agents grouping process. Each agent is represented by its specific family color (groups).

The agents population can be composed heterogeneously by individuals with different characteristics. Each agent$_i$ is defined according to the following parameters:

- Id$_i$ – Identifier of the agent.
- IdFamily$_i$ – Identifier of the family. A family is a predefined group formed by some agents who know each other. All of them are indicated by the same color (in this paper represented by the same gray scale), to facilitate group identification during the simulation as shown in Figure 33.
- DE_i – Dependence level of the agent represented by a value in the interval [0,1], which mimics the need for help of agent$_i$.
- AL_i – Altruism level of the individual represented by a value in the interval [0,1]. It represents the tendency of helping an other agent. For simplicity, we consider altruism existent between members of the same family only, i.e. agents with high altruism try to rescue dependent agents of the same family.
- v_i^o – Desired speed of the agent.

In order to model the effect of the dependence parameter in the individual velocity, we computed v_i^m as a function of DE_i and maximum velocity v_i^o , as follows:

$$v_i^o = (1 - DE_i)v_i^m \qquad \text{(Eq. 2)}$$

If the agent is totally dependent (DE_i=1), v_i^o will be equal to zero, which is typical for disabled people, small children, etc. In the case of DE_i = 0 for all agents one recovers Helbing´s original model.

The impact of parameter AL_i is presented in next section where the altruism force is introduced according to the interaction between agents, forming groups of altruist and dependent individuals.

Group formation is related to $\overrightarrow{Fa_i}$ (altruism force) which is implemented as an interaction between two or more agents who are part of the same family. The resultant $\overrightarrow{Fa_i}$ is mathematically described as follows:

$$F\vec{a}_i = K.\sum_j AL_i DE_j \left|\vec{d}_{ij} - \vec{d}_{ip}\right| \vec{e}_{ij} \qquad \text{(Eq. 3)}$$

The vector $\vec{d}_{ij}$ represents the distance between the two agents with the origin at position of agent$_i$ and $\vec{d}_{ip}$ is the distance vector pointing from the agent$_i$ to the door´s position p of the simulation environment (Figure 34). K is a constant and $\vec{e}_{ij}$ is the unitary vector with origin at position i.

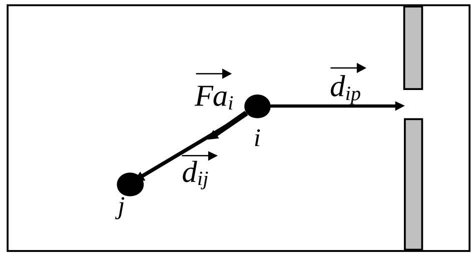


Figure 34. : Representation of vectors for a pair of agents.

Consequently, the greater the parameter AL_i of agent$_i$, the bigger will be $\overrightarrow{Fa_i}$ which points to the agent$_j$ and has the high level of DE_j. When both agents are close enough to each other, the one with high DE (agent$_j$ in this example) adopts the value of agent$_i$ ($DE_j = DE_i$). This means that the evacuation ability of *agent$_i$* is shared with *agent$_j$* and both start moving together (shown in Figures 33 and 34).

The results analyzed in this work are subdivided in two parts. First, we present a discussion about the impact of average *DE* and *AL* parameters on the resulting flow of people passing the door per second, during the simulation. We investigated these parameters varying *DE* with *AL* fixed and vice-versa. In the second set of simulations, we compared the average flow of people per second obtained with different distributions of initial population. Moreover, these flow values are compared with the one obtained with Helbing´s model.

We performed 25 simulations using 5 different seeds for random number generator with evaluated values of AL (0.1, 0.3, 0.5, 0.7 and 0.9), as shown in the following list of parameters:

Value of *DE* = 0.5
Standard Deviation of DE_i = 0.5
Value of *AL* = (0.1, 0.3, 0.5, 0.7 and 0.9)
Standard Deviation of AL_i = 0.1

Figure 35 presents the average flow of people per second obtained with the 25 simulations with fixed DE and variable AL.

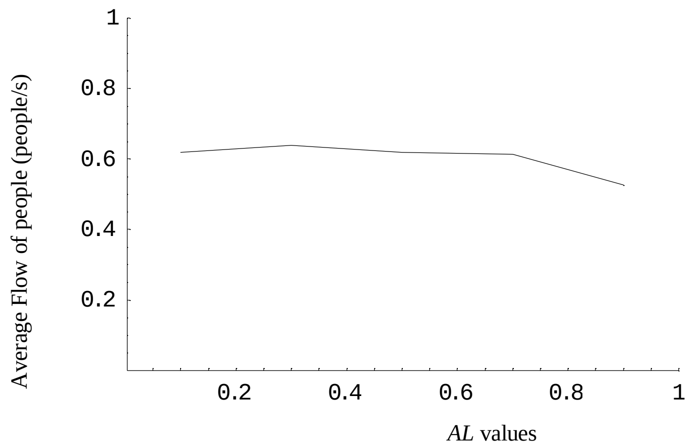


Figure 35: Flow of people x AL values.

Twenty five simulations (using also 5 different seeds) were performed considering the following data:

Value of *DE* = (0.1, 0.3, 0.5, 0.7 and 0.9)
Standard Deviation of DE_i = 0.1
Value of *AL* = 0.5
Standard Deviation of AL_i = 0.5

Figure 36 presents the average flow of people per second obtained with the 25 simulations with fixed *AL* and variable *DE*.

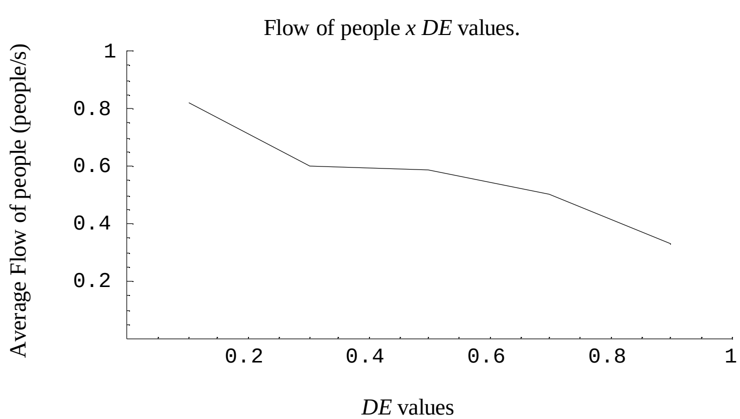


Figure 36: Flow of people x DE values.

From those plots one can perceive that the average flow of people per second decreases according to the increased values of *DE*. Figure 36 indicates that the flow of people does not increase or decrease significantly, for a more accurate study considering a combined AL-DE dependency we refer to future work. Below, we provide more detailed results defining different possibilities of initial populations.

In order to measure the We Impact of Population Distribution in the Resulting Average Flow of People performed further 25 simulations using 5 different seeds, considering the following populations:

A) Value of *DE* = 0.0 Standard Deviation of DE_i = 0.0 Value of *AL* = 0.0 Standard Deviation of AL_i = 0.0	B) Value of *DE* = 0.9 Standard Deviation of DE_i = 0.2 Value of *AL* = 0.9 Standard Deviation of AL_i = 0.2
C) Value of *DE* = 0.8 Standard Deviation of DE_i = 0.2 Value of *AL* = 0.1 Standard Deviation of AL_i = 0.2	D) Value of *DE* = 0.5 Standard Deviation of DE_i = 0.5 Value of *AL* = 0.5 Standard Deviation of AL_i = 0.5
E) Value of *DE* = 0.1 Standard Deviation of DE_i = 0.2 Value of *AL* = 0.9 Standard Deviation of AL_i = 0.2	

An interpretation for each type of population is described as follows:

Population A reproduces Helbing´s implementation, i.e. individuals are considered homogeneous and without altruism or dependency. Population B describes a socially complicated configuration since the major part of agents are altruist and are also dependent. We have classified these agents as problematic since they want to help others but their ability to escape from dangerous locations is limited. Population C describes a more egoist population; only the minor part desires to help others but the major part of agents needs help. Population D is described with a normal distribution and serves as a mean reference sample. Population E presented the higher values of flow of people since major part of agents desires to help others and there is a small portion of them who needs help.

Figure 37 presents average flow of people per second obtained in these 5 simulations.

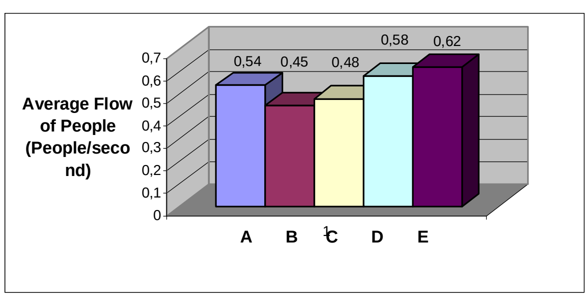


Figure 37. : Flow of people x Initial Population.

Comparing the results A to E, one observes that including individuality into agents, permits to simulate different populations which is manifest in a change in the average flow of people passing the door per second. For instance, we can simulate a population of children in a school (like Population B) and observe that the average flow of people decreases compared to other populations, as described by Helbing´s case (Population A).

On the other hand, we are able to simulate a population of trained people who knows how to evacuate a dangerous location. In this case, people can help others if necessary, but in fact, there are not enough people with necessity for assistance. This situation can be an example of Population E) and in this case, we obtained the highest flow compared to all other populations simulated.

VII. Conclusion

Intelligent Virtual Reality Environments (IVRE) enclose an attractive and challenging research field, which combines different technologies used to construct 3D environments integrated with Artificial Intelligence and Behavioural Simulation methods. IVRE research involves a multidisciplinary team group, which should concentrate efforts in the development of both graphical environment aspects and embodied intelligence.

In this tutorial we presented an overview of several aspects related to the development of IVREs:

- Virtual environment modelling, including the addition of "intelligent information" into the environments components;
- Agents simulation, including environment interaction, scripts, behaviours and agent control architectures used in order to accomplish a specific task;
- Groups and Crowds simulation, including studies about behavioural groups simulation and practical examples of application of this methods;
- Virtual Environment automatic creation and organization, including some practical examples of adaptive IVREs;

The combination off all those aspects allows us to create more dynamic, interactive and realistic IVREs. Several applications developed based on these techniques, have already been used with great success in different domain applications, ranging from educational tools, e-commerce, behavioural simulation, analysis of dangerous situations, games and entertainment and industrial applications.

However, there are still many other "open questions" and problems to be solved, and also techniques that can be greatly improved, in order to achieve better IVRE applications. We believe the joint effort of different research groups and from different knowledge areas is the way to achieve better results. Virtual Reality and Artificial Intelligence are definitively coupled, and topics as for example, Agents Simulation, Multi-Agents Systems, Artificial Life, Behavioural Simulation, Robotics, Advanced Human-Computer Interfaces and Interaction, and Machine Learning are key-points to the development of more powerful IVREs.

With the improvement and addition of new capabilities to IVREs, certainly new and very attractive home and industrial applications will come!

REFERENCES

[Anastassakis 2001] Anastassakis, G.; Ritching, T.; Panayiotopoulos, T. "Multi-agent Systems as Intelligent Virtual Environments". LNAI 2174, 381-395, 2001.

[Aylett 2000] Aylett, R. and Luck, M. (2000) "Applying Artificial Intelligence to Virtual Reality: Intelligent Virtual Environments". Applied Artificial Intelligence, v. 14, n. 1, p. 3-32.

[Aylett 2001] Aylett, R. and Cavazza, M. (2001) "Intelligent Virtual Environments – A state of the art report". Eurographics Conference, Manchester, UK.

[Bandi 1998] Bandi, S. and Thalmann, D. (1998) "Space Discretization for Efficient Human Navigation". Computer. Graphics Forum, v. 17, n.3, p.195-206.

[Bandi 2000] Bandi, S. and Thalmann, D. (2000) "Path finding for human motion in virtual environments". Computational Geometry: Theory and Applications, v. 15, n. 1-3, p. 103-127.

[Barros 2002] Barros, L. M.; Evers, T. F.; Musse., S. R. "A Framework to Investigate Behavioural Models". Journal of WSCG. Plzen, Vol. 10, n.1, pp-40-47. February 2002

[Benesh 1995] Benesh, H. (1995) "Atlas de la Psychologie. Encyclopédies d'Aujourd'hui". Librairie Générale Française.

[Bersot 1998] Bersot, O.; Guedj, P.; Godéreaux; C.; Nugues, P. (1998) "A Conversational Agent to Help Navigation and Collaboration in Virtual Worlds". Virtual Reality, v. 3, n. 1, p. 71-82.

[Bouvier 1997] Bouvier, E.; Cohen E.; and Najman. L. "From crowd simulation to airbag deployment: particle systems, a new paradigm of simulation". Journal of Electronic Imaging 6(1), 94-107 (January 1997).

[Braun 2003] Braun, A.; Musse, S.; Oliveira, L.; Bodmann, B. (2003) "Modeling Individual Behaviors in Crowd Simulation". Proceedings of the Computer Animation and Social Agents, p. 143-148, May, New Jersey, USA.

[Brenner 1998] Brenner, W.; Zarnekow, R.; Witting, H. (1998) "Intelligent Software Agents: Foundations and Applications". Springer-Verlag.

[Brogan 1997] Brogan, D. and Hodgins, J. "Group Behaviours for Systems with Significant Dynamics". Autonomous Robots, 4, 137-153. 1997.

[Brogan 1998] Brogan, D.C., Metoyer, R.A. and Hodgins, J.K. "Dynamically simulated characters in virtual environments". In IEEE Computer Graphics and Applications. Vol.18, No5, pp 58-69. Sept. 1998.

[Cassel 2000] Cassel, J. (2000) "Embodied Conversational Agents". MIT Press, Boston.

[Chittaro 2002] Chittaro, L. and Ranon, R. (2002) “New Directions for the Design of Virtual Reality Interfaces to E-Commerce Sites”. Proceedings of the 5th International Conference on Advanced Visual Interfaces, ACM Press, New York, May.

[Chittaro 2002a] Chittaro, L. and Ranon, R. (2002a) “Dynamic Generation of Personalized VRML Content: a General Approach and its Application to 3D E-Commerce”. Proceedings of the 7th International Conference on 3D Web Technology, ACM Press, New York, February.

[Chittaro 2003] Chittaro, R.; Ranon, R.; Ieronutti, L. (2003) “Guiding Visitors of Web3D Worlds through Automatically Generated Tours”. Proceedings of the 8th International Conference on 3D Web Technology, ACM Press, New York, March.

[Correa 1994] Correa, Filho. (1994) “A arquitetura de diálogos entre agente cognitivos distribuídos”. Rio de Janeiro: UFRJ (PhD Thesis).

[Correa 1998] Correa, M. ; Coelho, H. . From Mental States and Architectures to Agent's Programming. In: Helder Coelho. (Org.). Progress in Artificial Intelligence - IBERAMIA 98. Heidelberg, 1998, v. , p. 64-75.

[Demazeau 1995] Demazeau, Y. (1995) “From Interactions to Collective Behaviour in Agent-Based Systems”. Proceedings of the 1st. European Conference on Cognitive Science, Saint-Malo, France.

[Donikian 1995] S. Donikian, E. Rutten. “Reactivity, Concurrency, Data-flow and Hierarchical Pre-emption for Behavioural Animation.” Paradigms in Graphics’95, Eurographics collection. (Springer-Verlag, 1995).

[Donikian 1997] S. Donikian. ‘‘VUEMS: A virtual urban environment modeling system.” Computer Animation ’97. 127-133, (1997).

[Donikian 1999] S. Donikian. “Towards Scenario Authoring for Semi-Autonomous entities.” International Conference on Visual Computing (ICVC99). (Goa, India, Feb 1999).

[Dudek 2000] Dudek, Gregory; Jenkin, Michael. Computational principles of mobile robotics. Cambridge: Cambridge University, 2000. 280 p.

[Evers 2002] Evers, T., and Musse, S. (2002) "Building Artificial Memory of Autonomous Agents Using Dynamical and Hierarchical Finite State Machine". In Computer Animation, Geneva, Switzerland. June 2002.

[Farenc 1999] N. Farenc, R. Boulic, D. Thalmann. “An informed environment dedicated to the simulation of virtual humans in urban context.” Proc. Eurographics ’99, Computer Graphics Forum, 18,(3),C–309–C–317, (1999).

[Farenc 2000] N. Farenc, et al. “A paradigm for controlling virtual humans in urban environment simulation.” Journal Applied Artificial Intelligence, 14,(1),69–91, (2000).

[Franklin 1996] Franklin, S. and Graesser, A. (1996) “Is it an agent, or just a program? A taxonomy for autonomous agents”. Proceedings of the 3rd International Workshop on Agent Theories, Architectures and Languages, Springer-Verlag, Berlim.

[Frery 2002] Frery, A.; Kelner, J.; Moreira, J., Teichrieb, V. “Satisfaction through Empathy and Orientation in 3D Worlds”. CyberPsychology and Behavior, vol. 5, no. 5, pp. 451 – 459, 2003.

[Garcia 2003] Garcia, A. and Sichman, J. (2003) “Agentes e sistemas multiagentes”. In: Rezende, S. Sistemas Inteligentes: fundamentos e aplicações. Barueri, SP: Manole. cap. 11, p. 269-306.

[Goodwin 1994] Goodwin, R. (1994) “Formalizing Properties of Agents”. Technical Report, School of Computer Science, Carnegie-Mellon University, Pittsburg, PA, September.

[Hayes-Roth 1995] Hayes-Roth, B. (1995) “An architecture for adaptive intelligent systems”. Artificial Intelligence (Special Issue on Agents and Interactivity), v. 72, pp. 329-365.

[Heinen 2002a] Heinen, F. (2002a) “Sistema de Controle Híbrido para Robôs Móveis Autônomos”. São Leopoldo: UNISINOS (Master’s thesis).

[Heinen 2002b] Heinen, F. and Osório, F. (2002b) “HyCAR - A Robust Hybrid Control Architecture for Autonomous Robots”. HIS 2002 - Hybrid Intelligent Systems, 2002, Santiago, Chile.

[Helbing 2000] Helbing, D.; Farkas, I.; Vicsek, T. (2000) “SimulatingDynamical Features of Escape Panic”. Nature, v. 407, pp. 487-490.

[Hu 2003] Hu, jinhui; You, Suya; Neumann, Ulrich. “Approches to Large-Scale Urban Modelling”. IEEE Computer Graphics and Applications, V.23, No. 6. Nov./Dec. 2003.

[Kallmann 1998] Kallmann, M. and Thalmann, D. (1998) “Modeling Objects for Interaction Tasks”. Proceedings of the Eurographics Workshop on Animation and Simulation.

[Kallmann 1999] Kallmann M. and Thalmann, D. (1999) “A Behavior Interface to Simulate Agent-Object Interactions in Real-Time”. Proceeding of Computer Animation, IEEE Computer Society Press, p. 138-146.

[Kuffner 1998] Kuffner, J. (1998) “Goal-Directed Navigation for Animated Characters Using Real-Time Path Planning and Control”. Workshop on Modeling and Motion Capture Techniques for Virtual Environments, Lecture Notes in Artificial Intelligence 1537, Springer-Verlag, Berlin, p. 171-187.

[Latombe 1991] Latombe, J. (1991) “Robot Motion Planning”. Kluwer Academic Publisher, Boston, MA.

[Le Goff 1994] Le Goff, Hervé. “Modélisation Comportamentale de Groupes d’Entités Hétérogènes”. Rapport d’ètude bibliographique. IRISA, France, 1994.

[Maes 1994] Maes, P. (1994) “Modeling Adaptive Autonomous Agents”. Artificial Life Journal.

[Marson 2003] Marson, F., Jung, C. and Musse, S. (2003) "Modelagem Procedural de Cidades Virtuais". In SVR - Simpósio Brasileiro de Realidade Virtual, Ribeirão Preto, Brazil, October 2003.

[Milde 2000] Milde, J. (2000) “The instructable Agent Lokutor”. Working Notes – Autonomous Agents 2000, Workshop on Communicative Agents in Intelligent Virtual Environments, Barcelona, Spain.

[Musse 1997] Musse, S., Thalmann, D. (1997) "A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis". Proceedings of Eurographics - CAS'97 - Workshop of Computer Animation and Simulation. Budapeste, Hungary, Sept, 1997.

[Musse 2000] Musse, S. (2000) “Human Crowd Modelling with Various Levels of Behaviour Control”. Lausanne: EPFL (Tese de doutorado).

[Musse 2001] Musse, S., and Thalmann, D. (2001) "A Hierarchical Model for Real Time Simulation of Virtual Human Crowds". IEEE Transactions on Visualization and Computer Graphics, V. 7, N.2, pp. 152-164, April-June, 2001.

[Musse 2003] Musse, S., Silva, A., Roth, B., Hardt, K., Barros, L., Tonietto, L. And Borba, M. (2003) “PetroSim: A Framework to Simulate Crowd Behaviors in Panic Situations”. In MAS 2003 - Modeling & Applied Simulation, Bergeggi, Italy, 2003.

[Nijholt 2000] Nijholt, A. and Hulstijn, J. (2000) “Multimodal Interactions with Agents in Virtual Worlds”. In: Kasabov, N. (ed.): Future Directions for Intelligent Information Systems and Information Science, Physica-Verlag: Studies in Fuzziness and Soft Computing.

[Nilsson 1980] Nilsson N. (1980) “Principles of Artificial Intelligence”. Tioga Publishing Company.

[Noll 1999] Noll, S.; Paul, C.; Peters, R.; Schiffner, N. “Autonomous Agents in Collaborative Virtual Environments”. IEEE 8th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, June, Palo Alto, California, 1999.

[Nwana 1996] Nwana, H. (1996) “Software Agents: An Overview”. In: Knowledge Engineering Review, v. 11, n. 3, p. 205-144.

[Panayiotopoulos 1999] Panayiotopoulos, T.; Zacharis, N.; Vosinakis, S. “Intelligent Guidance in a Virtual University”. Advances in Intelligent Systems – Concepts, Tools and Applications, pp. 33-42, Kluwer Academic Press, 1999.

[Parent 2002] Parent, R. (2002) “Computer Animation: Algorithms and Techniques”. Morgan Kaufmann. San Francisco. 527p.

[Perlin 1996] Perlin, K. and Goldberg, A. (1996) “Improv: A system for scripting interactive actors in virtual worlds”. ACM Computer Graphics Annual Conference, p. 205-216.

[Rao 1995] Rao, A. and Georgeff, M. (1995) “BDI Agents: From Theory to Practice”. Proceedings of the 1st International Conference on Multi-Agents Systems, San Francisco, CA, June.

[Rapaport 1996] Rapaport, O. C. “The Art of Molecular Dynamics Simulation”. Cambridge University Press. New York, 1996.

[Reilly 1992] Reilly, S. and Bates, J. (1992) “Building Emotional Agents”. Technical Report, School of Computer Science, Carnegie-Mellon University, Pittsburg, PA, May.

[Reynolds 1987] Reynolds, C. “Flocks, Herds and Schools: A Distributed Behavioral Model”. Proc. SIGGRAPH ’87, Computer Graphics, v.21, n.4, July, 1987.

[Rickel 1997] Rickel, J. and Johnson, W. “Integrating Pedagogical Capabilities in a Virtual Environment Agent”. Proceedings of the 1st International Conference on Autonomous Agents, February, ACM Press, 1997.

[Rickel 2000] Rickel, J and Johnson, W. “Task-Oriented Collaboration with Embodied Agents in Virtual Worlds”. Embodied Conversational Agents, MIT Press, Boston. 2000.

[Rizzo 2002] Rizzo, A.; Bowely, T.; Buckwalter, G.; Schultheis, M.; Matheis, R.; Shahabi, C.; Neumann, U.; Kim, L.; Sharifzadeh, M. “Virtual Environments for the Assessment of Attention and Memory Processes: The Virtual Classroom and Office”. Proceedings of the International Conference on Disability, Virtual Reality and Associated Technology, Vesaprem, Hungary, September, 2002.

[Russell 1995] Russell, S. and Norvig, P. (1995) “Artificial Intelligence: A Modern Approach”. Prentice Hall Series in Artificial Intelligence, New Jersey.

[Santos 2004] Santos, C. “Um Ambiente Virtual Inteligente e Adaptativo baseado em Modelos de Usuário e Conteúdo”. São Leopoldo: UNISINOS (Master’s thesis). Available in: http://www.inf.unisinos.br/~cassiats/dissertação.

[Santos 2004a] Santos, C. and Osório, F. (2004) “An Intelligent and Adaptive Virtual Environment and its Application in Distance Learning”. Advanced Visual Interfaces, Gallipoli, Italy, May, ACM Press.

[Santos 2004b] Santos, C. and Osorio, F.: AdapTIVE: An Intelligent Virtual Environment and its Application in E-Commerce In: 28th Annual International Computer Software and Applications Conference, 2004, HongKong. Proceedings of the 28th Annual International Computer Software and Applications Conference. Los Alamitos: IEEE Computer Society Press, 2004, p.468 – 473.

[Schweiss 1999] Schweiss, E.; Musse, S.; Garat, F.; Thalmann, D. (1999) “An architecture to guide crowds based on rule-based systems”. Proceedings of the 1st International Conference Autonomous Agents, ACM Press.

[Sycara 1996] Sycara, K.; Decker, K.; Pannu, A.; Williamson, M.; Zeng, D. (1996) “Distributed Intelligent Agents”. IEEE Expert.

[Tecuci 1998] Tecuci, G. (1998) “Building Intelligent Agents: A Multistrategy Learning Theory, Methodology, Tools and Case Studies”. Academic Press, 320 p.

[Terzopoulos 1994] Terzopoulos, D; Tu, X.; Grzeszczuk, R. (1994) “Artificial fishes with autonomous locomotion, perception, behavior and learning, in a physical world”. In Maes, P. and Brooks, R., editors, Proceedings of the Artificial Life IV Workshop, MIT Press, p. 17-27.

[Terzopoulos 1999] Terzopoulos, D. (1999) “Artificial Life for Computer Graphics”. Communications of the ACM, v.42, n. 8, p.33-42.

[Thalmann 1994] Thalmann N. and Thalmann D. (1994) “Creating Artificial Life in Virtual Reality”. Artificial Life and Virtual Reality, John Wiley, Chichester, pp.1-10.

[Thalmann 1995] Thalmann, N. and Thalmann, D. (1995) “Actors for Interactive Television”. IEEE Special Issue on Digital Television.

[Thalmann 1996] Thalmann, D. (1996) “A New Generation of Synthetic Actors: The Interactive Perceptive Actors”. Pacific Graphics’96, p.200-219.

[Thalmann 1999] Thalmann, D.; Musse, S.; Kallmann, M. (1999) “Virtual Humans’ Behaviour: Individual, Groups, and Crowds”. Digital Media Features.

[Thomas 2000] G. Thomas, S. Donikian. “Modelling virtual cities dedicated to behavioural animation. Eurographics 2000, (Interlaken, Switzerland, 2000).

[Thomas 2000a] G. Thomas, S. Donikian. “Virtual humans animation in informed urban environments Computer Animation 2000, IEEE, (Philadelphie, USA, May 2000).

[Tomaz 2000] Tomaz, G. and Donikian, S. (2000) “Modeling virtual cities dedicated to behavioural animation”. The International Journal of The Eurographics Association (Computer Graphics Forum), v.19, n.3, p.71-80.

[Torres 2003] Torres, J.; Nedel, L.; Bordini, R. (2003) “Using the BDI Architecture to Produce Autonomous Characters in Virtual Worlds”. Proceedings of the Interactive Virtual Agents, Irsee, Germany, p. 197-201.

[Tu 1994] Tu, X. and Terzopoulos, D. “Artificial Fishes: Physics, Locomotion, Perception, Behavior”. Proc. SIGGRAPH ’94, Computer Graphics, July 1994.

[Viccari 1996] Viccari, R.; Giraffa, L. (1996) “Sistemas Tutores Inteligentes: abordagem tradicional x abordagem de agentes”. Proceedings of the 13th Brazilian Symposium on Artificial Intelligence, Curitiba, PR, Brazil, Springer Verlag.

[Viccari 2001] Viccari, R. M. ; Giraffa, L. M. . “The Use of Multi Agent System to Build Intelligent Tutoring Systems”. In: International Conference on Computing Anticipatory Systems (CASYS), 2001, Liege. Computing anticipatory systems. Melville : American Institute, 2001. v. 1. p. 340-348.

[Villamil 2003a] Villamil, M., Musse, S. and Luna, L. (2003a) A Model for Generating and Animating Groups of Virtual Agents. In IVA 2003, Kloster-Irsee, Germany.

[Woo 1998] Woo, Mason. (1998) “OpenGL Programming Guide: The Official Guide to Learning OpenGL”. Addison-Wesley, 650 p.

[Wooldridge 1995] Wooldridge, M. and Jennings, N. (1995) “Intelligent Agents: Theory and Practice”. In: Knowledge Engineering Review, v. 10, n. 2.