

Database and Virtual Human Representation on the WEB

Nathalie Farenc, Christian Babski, Fabien Garat and Daniel Thalmann
Computer Graphics Laboratory
Swiss Federal Institute of Technology
Lausanne, Switzerland
nathalie, babski, garat, thalmann@lig.di.epfl.ch

Abstract

In the few last years, and as a result of the increasing growth of the Internet, more and more virtual places can be visited using applications available on the WEB (World Wide Web). The data provided (accessible) can be only text, or a 2D map with text associated or a 3D model or a mix of these technologies. The means of transport into data can belong to meta-data file management via hyper-text links, selection of fixed view of a model or via a HTML browser (netscape or InternetExplorer) for the WEB with the addition of free-plugins). We aim at visiting 3D model with a VRML plugin, querying an associated database in order to have more information about the environment. The paper introduces a virtual humanoid animation in a web-based application where HANIM humanoids are animated using VRML and JAVA. We propose an integration of a database containing geometrical and semantic data about the 3D model visited with a representation of virtual humanoids populating the place. We present an Informed Environment, which corresponds to a database dedicated to urban life simulation. This integration requires a client-server link to transmit data.

1. Introduction

In the few last years, and as a result of the increasing growth of the Internet, more and more virtual places can be visited using applications available on the WEB (World Wide Web). Restoration and maintenance of historic monument, evolution of places such as the Vatican [25], reconstruction of real urban model for testing infrastructure amelioration or virtual visit of museum and monuments are more and more possible and correspond to the fast technical improvement of computers. How to access and visualise the huge quantity of data stored since many years and via many different types of format? We have chosen in a first stage to use 3D model of the reconstructed scene and to have a human representation to visit it. The scene is associated to a database in order to furnish data less visible such as

associated text information or information about objects movable or not, name of places or how to reach a place from a location. To create realistic simulations the database can also provide information about basic actions to perform according to the place visited.

In a first time, created to provide virtual humans with that we shall call "urban knowledge", the database can also store more general information about the place such as historic or global knowledge information. Figure 1 shows example of questions that a user or an autonomous agent can give to himself in a virtual scene.

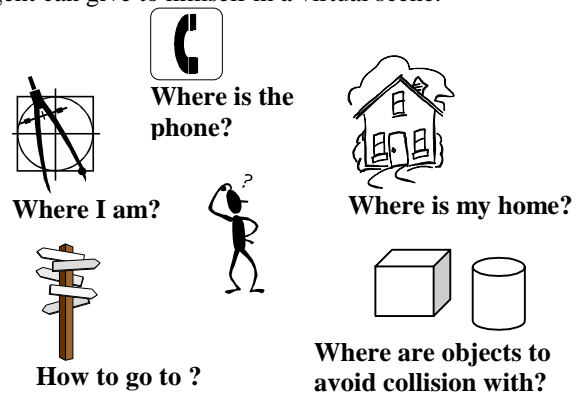


Figure 1. An autonomous agent in a virtual world

Web Human Director (WHD) is a JAVA based application. It is working in association with a normal web browser and a VRML plugin [4], [8], [28] to display the 3D world. A WHD application is able to construct an entire VRML [27] scene and to include HANIM virtual humanoids [17] inside the created scene (Figure 2). WHD can run from the WEB as well as locally on any platform without any further changes. WHD is made to be highly configurable by implementing its own system of plugin, which can take control of one or several virtual humans in the VRML scene. A more precise description will be made in section 3.

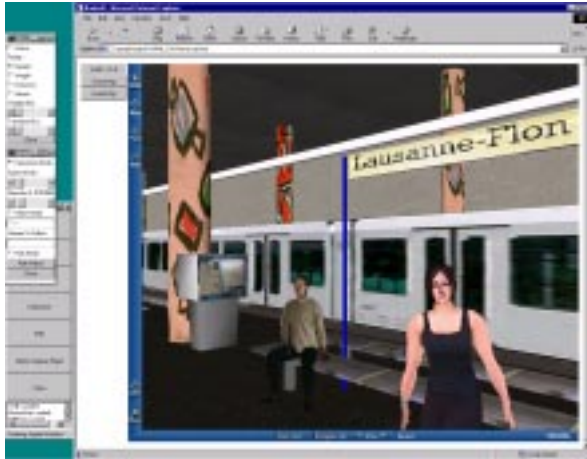


Figure 2. A WHD scene featuring two real-time deformed bodies in the Lausanne Metro.

In order to merge the CITY database and WHD, a client/server architecture is used. Thus each application can run on different computers, distributing computational cost, and providing a way to exchange data between each of them. Chavez, in [9], uses such architecture to manage shared information between clients working on distant computers. A protocol based on an open architecture inspired from [10] is implemented in order to integrate dynamically new features. A similar architecture describes in [16] shows that associating a client to a given feature, provides the ability to update easily the set of available features. In addition of this flexibility, the fact that a client is in charge of only one feature brings stability to the system.

1.1.Related Work

We present here some works related to the following topics: description of databases application associated to a virtual scene (cities or monument), application on the WEB and example of client-server used to connected the previous type of application with databases.

We can find several fields where reconstruction of places or database management is used to give back knowledge related to a location. The first one concerns real actual places reconstructed in order to establish solution for urban problem [11], [14] such as traffic problem with data flow analyse for example [2], [18], [20], [19], [23]. Other ones concern what we shall call "virtual heritage", product proposed for tourism as leisure time increases, [1], [25] [22]. The educational aspect using the WEB is becoming increasingly important and access to different data archives starts to be a real challenge. It was also demonstrated that a WEB context (HTML browser plus plugins) is valuable for complex applications like motion capture [6]. How to make data accessible on the WEB in a convivial way? Hyperlink on html pages is, for now, a very common way to provide information. Virtual visits of 3D scene open a new field of investigation for data access [1]. An overview of the

different architecture for network systems can be found in [15]. The same architecture has already been used in order to merge CITY database and ViCrowd, an application managing crowds of virtual humans without WEB integration[24], an architecture study has been made with more clients[12].

1.2.Overview of the architecture

The presented architecture corresponds to an integration of a WEB based application handling animation of Virtual Humans (WHD) [4] with a database (CITY) via a client-server system (SERVER).

WHD is a JAVA application constructing an entire VRML scene handling HANIM [17] humanoids. During a simulation the user can animate a body giving action to perform such as walk with speed and direction information. The user controls humanoids in real time.

CITY is a database application providing information about a scene allowing high level orders such as "go to the station" or informing where are objects or where a pedestrian can walk. It can also provide additional information such as text information describing a place.

SERVER is a client-server application, passing on messages associated to WHD requests and CITY answers. The server evaluates its connections with its clients and provides to clients the requests that others client are able to deal with.

In this paper, the CITY furnishes information about the scene to WHD via the SERVER, using a usual network protocol.

1.3.Organisation of this paper

The paper is structured as follow: in the first section we present how the database associated to a scene is structured and how we can use it. The next section, presents the Web Human Director for visiting 3D scenes on the WEB. The section 4 corresponds to the client-server presentation and the last section shows the integration of these modules followed by a discussion and a conclusion.

2.The Database Associated to the scene

2.1. Description

A Virtual City Database (CITY) maintains the database of the virtual environment. In a first stage we have decomposed the scene in a hierarchy of Environmental Entities (ENV). Each ENV is either a group of other ENVs or a leaf in the hierarchical tree decomposition model. This description of decomposition of ENVs is described in a generic model and is used both by designers of the scene and database constructors. The nomination of objects, their location in the scene graph,

some specification about how to create an ENV and its location in the hierarchy are described in the generic model. A high level of decomposition example of an urban model can illustrate the decomposition model of the database. The city level is decomposed into "quarters" and the quarters are decomposed into block levels. We define a block as a part of a quarter, composed of streets, junctions and "parcels". Junctions are crossroads that connect streets. We consider as a parcel a portion of land that has no streets or junctions inside, like a park or a piece of land with buildings. In this way all the space covering a block can be marked out and all points on the surface of a block can be associated with an ENV. This global spatial cover can be completed or not, depending on the use of the database. To minimise the number of objects attached to an ENV, some conventions have been defined for an ENV conception. In a place such as a room, all the walls are outside of the ENV's surface and then are not recognised like objects for collision avoidance. If virtual autonomous humanoids use only surfaces defined by the ENVs the collision avoidance concerns only the objects belonging to the surface, it is not necessary to know all the objects surrounding an ENV.

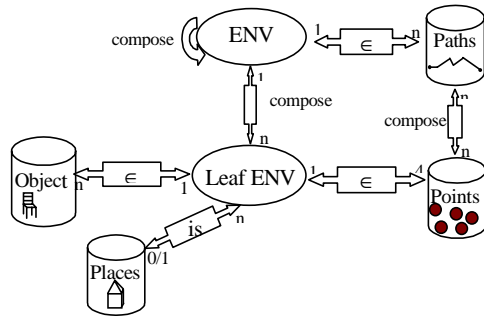


Figure 3. The database model

Figure 3 shows the database design and the association existing between the ENVs and the objects included in the ENVs surfaces, the computed and stored paths allowing the walkthrough in a scene, the entry/exit points and the notion of places.

2.2. Construction of the database

The ENVs correspond to some surfaces, with their names, carrying some semantic notions. The semantic corresponds to what we shall call an "urban knowledge". This knowledge allows more intelligent behaviour in an urban context defining for example "where to walk for pedestrians" or "how to cross a road", and "what is a road". Then, the database is able to provide different services to applications needing information about the environment such as the Web Human Director System: The scene and its internal hierarchy is analysed, the objects names and the labels found are used to create the database. The database is based on a set of ENVs structured in a hierarchy and containing different levels of information. The designers can have previously associated some semantic notions to the generic model of

ENV and the database can be automatically more complete at the stage of the construction. The information stored is about the geometry of the scene, the location of objects but also location of the places, and information inherent to the structure of the city such as "here is a location on a sidewalk, belonging to a street and after a quarter". The ENVs are also linked with some connectivity graphs using some Entry/Exit points allowing the entry and the exit of the ENV with the mention concerning the mobiles (objects with mobility such as pedestrians, cars, buses, or bicycles) able to use them. With such connectivity graphs it is possible to perform path calculation from one point to another one, using some criteria like the shortest distance. The user can add manual connections between different scenes to go through one scene to another one by switching to the associated database. An example of use can be a link between the interior of a building (scene and database associated) and the city scene composed of the streets and exteriors representation of the buildings. This notion allows the overlapping of scenes with no space constraint. A building door can give access to a new scene without relation with the outside building dimension. The database in such situation provides the name of the scene to load and the name of the new database to use. Thus, the user creates a link connecting a "world" to another one.

The utilisation of the complete database is the last stage and can answer questions like "Where I am?" "Where is the train station, or a chair", "How can I go to the supermarket", "Is there a description of the place where I am? ". All these requirements correspond to a query protocol defined in common with the server and others clients.

When the database is completed from a geometrical point of view, it is possible to add action information in order to have high level control for virtual agents or full autonomous agents. According to the utilisation required by users, the database can also be completed with some text descriptions of the places and allows tourist visit. The WEB offers possibilities such as links with html pages to provide additional information attached to an object, a place or an ENV. A more detailed description of CITY can be found in [13] and [26].

3. The Web Human Director System

The top-level application is a JAVA applet. After obtaining a pointer to the VRML plugin, the WHD application is able to create and update the 3D world. To get a pointer to the VRML plugin and to send/receive events from/to the VRML plugin, WHD is using the External Authoring Interface (EAI) [27], which defines a standard way to communicate with a VRML plugin. Then, WHD is able to modify the VRML world by adding or removing objects. It is possible to access internal fields of any VRML objects for modifying the attributes like translation or rotation, as well as the shape of the object.

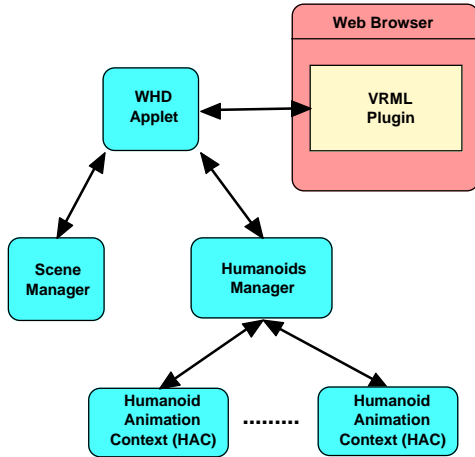


Figure 4. WHD overview

WHD is composed of two managers: a scene manager which is in charge of the environment and a humanoids manager which is in charge of all virtual humans included in the 3D environment (Figure 4).

A Human Animation Context (HAC) is associated to a virtual human. It produces the final animation that is applied to this humanoid.

Virtual humans controlled by WHD have to be described by using the HANIM specification [17]. It defines a standard way to obtain virtual humans in VRML. It describes a hierarchy (Figure 5) and defines a common way to name body articulations. A HANIM compliant library is able to animate any kind of HANIM compliant bodies without any problem, whatever who produced the virtual body.



Figure 5: An example of a HANIM body.

3.2. The WHD plugin system

WHD is designed to manage virtual humans in a virtual 3D world. After inserting virtual humans in the 3D

scene, WHD is in charge of the animation. As there are many ways to animate a virtual human (inverse kinematics, keyframe, motion capture...), a system of plugin was implemented. This system permits to integrate any kind of animation methods and to mix them together. The final mix is produced by the Human Animation Context (HAC) which groups all registered WHD animation plugins.

To implement a WHD plugin, one has to follow a given interface. The plugin interface is composed of a set of functions that permits to communicate with the HAC (Figure 6). It mainly allows sending new joint values to the HAC in order to update the position of the virtual human. In addition, the HAC is able to start and stop the WHD plugin at anytime (stop generating animation) in relation with the information coming from the graphic user interface (the user does not want anymore to use this plugin for example).

A plugin can be added at any time and will appear the next time WHD is launched. A default list of animation plugins includes a walking motor, a MPEG4 animation player [3], a real-time motion capture plugin [6] and a manual editor (joint by joint) [4].

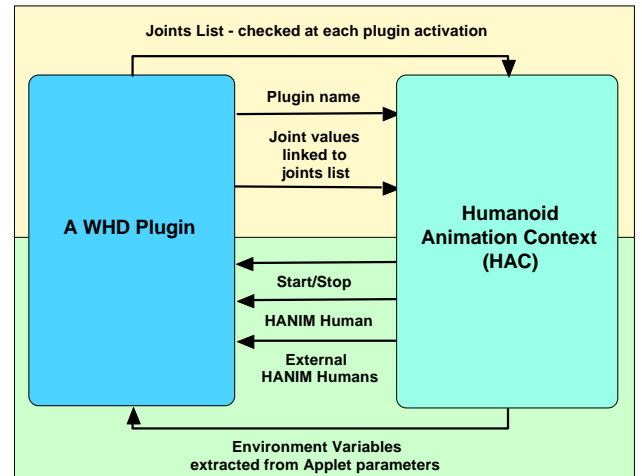


Figure 6. The WHD plugin interface.

3.3. The Human Animation context

A HAC is associated to each HANIM virtual body added to the 3D VRML scene. The HAC is receiving position updates from all active WHD plugins. HAC compute the final animation to apply to the humanoid by mixing all updates together. The mix is controlled through a graphic interface. It permits to define the way each update is taking in account. A plugin can need to be the only one modifying a joint or it is possible to apply partially the effect of a plugin.

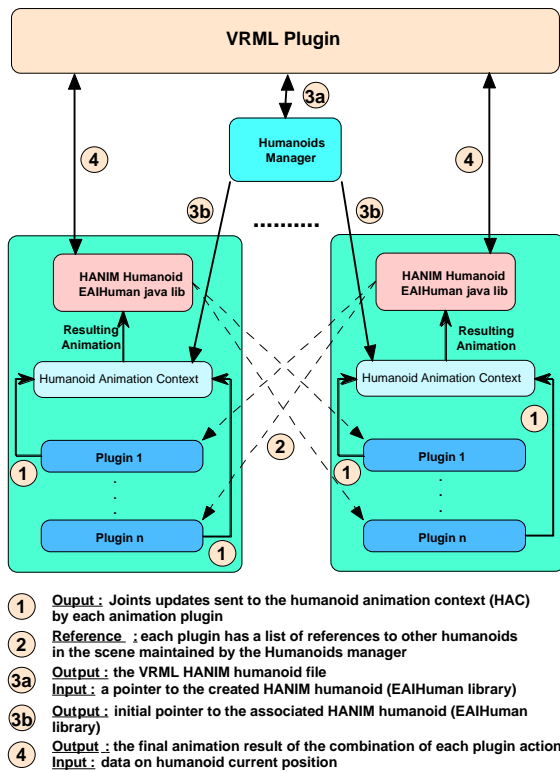


Figure 7: The HAC mechanism.

In addition of the virtual human it has in charge, the HAC is aware of all other virtual humans in the scene. This list of external bodies is communicated to each plugin. Then, the global position or the gesture of a given external body can be used as input by a WHD plugin to generate a specific animation (autonomous behaviour, virtual humans' interaction). The entire mechanism is described in Figure 7.

4. The Client-Server Architecture

There are many ways to communicate information between WHD and our database: shared memory, peer to peer connection or Client/server type architecture. We chose the last solution for its flexibility and its facility to interconnect heterogeneous systems. It also gives the ability to add new features related to new clients services.

The use of a shared memory would be faster than any network connections via sockets, but it implies that all applications run on the same computer. In such case, it is impossible to have two distant users interacting in the same environment and sharing the same database, as we will discuss later.

The use of peer to peer connection assumes that each application are able to be connected to other programs sharing information with it and others clients during simulations. This approach requires from user a good knowledge of all applications, and has a lack of flexibility

in case of multiple WHD sharing data from multiple databases.

Architecture Client/server consists in a central application (SERVER) with connections with other applications (clients). Each client communicates to others via the server, without knowing any physical locations of them, nor the way to communicate. The server can be considered as a black box, which externally offers services, and internally distributes information using a particular protocol for each client.

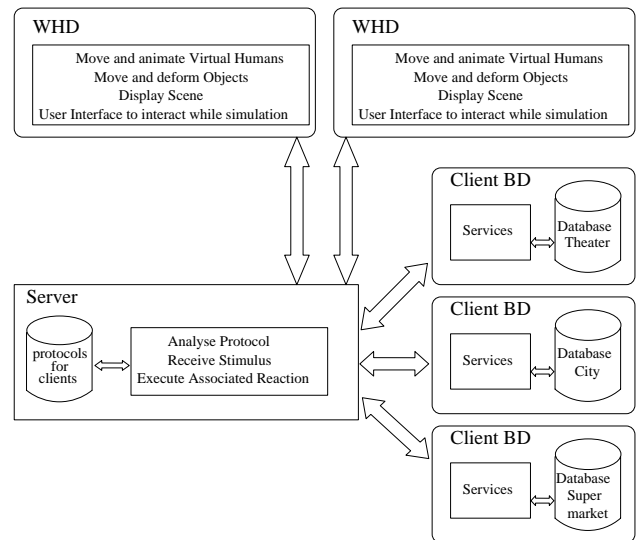


Figure 8. View of a configuration client/server

On the above figure, we describe a possible configuration, where two WHD clients access to services of two database clients. Then, it is possible to get as many databases we need to interconnect different scenes, and to update in real time any modifications about environment done by a virtual human in any WHD clients.

4.1 The Server

Server is the central application, which distributes an information emitted from a client to other connected clients. Each client ignores the internal way of process of other clients, but know which services they can offer via a protocol with the server. In this protocol, a request specifies how to ask the server the list of the clients and their services.

Server does not know which clients will be connected to itself, nor the services they can offer. Indeed, it allows clients to use some commands in order to send message or to manage priority of received messages. When connecting, each client sends to the server each type of message he can emit (stimulus), and the sequence of server commands (reactions) than it must execute when it receives such a stimulus. All this information constitutes the protocol, which allows client and server to share data. The first configuration shows in Figure 9 that Client 1 can

offer Service A and Service B. and it can use Service C and D from the Client 2 via the SERVER. In return, Client 2 can access to Service A and B from Client 1. When a third Client opens a connection with the server (second configuration), the server informs others clients of its presence. Thus, they can ask individually the Server what services the new client is able to provide. When receiving this request, Server sends queering clients a message updating the list of accessible services for all clients. So both clients 1 and 2 can access service E, and Client3 can then access services A, B, C, and D.

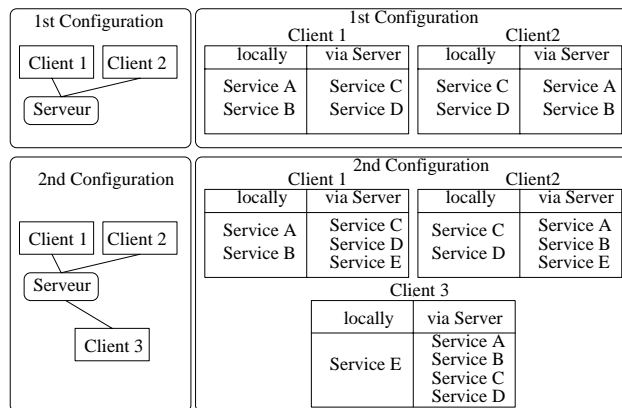


Figure 9. Situation of new client connection

4.2 The Clients

4.2.1 WHD Client

WHD Client allows to manage display, animation, deformation of objects [5] and virtual humans in a 3D scene. It can receive high level orders, but need to translate them in lower level order to execute them. A service from another client (like the CITY) can be needed in order to compute these low-level orders. As an example, a user needs to move a virtual human to a phone. WHD makes a request to the Database Client in order to know where the nearest phone is and the way to reach it. The computed path provides also the list of objects inside ENVs in order to avoid collision with any obstacles. Once the Database Client replied, WHD applies the way to this virtual human.

4.2.2 Database Client

Database client offers services relatives to environment, trajectory computation, labelled area localisation and availability of a place. It also allows the update of mobile objects positions, shared by multiple WHD clients at the same time. As WHD has internally no information about the scene environment, it cannot determine alone where are the areas where humans can freely move (walking on a sidewalk, instead of the road, which is reserved for engine).

4.3 The Protocol

At connection time to the server, a client sends his protocol to the server. Thus, the server knows how to react to future client's requests. This protocol consists of a list of stimulus and associated reactions. The stimuli are the messages sent from clients to server. The reactions are the sequence of commands that server executes when it receive specific stimulus. Reactions can be executed sequentially or in a parallel way, and each stimulus can be set with a specific priority. Reactions are commands known by server: to send messages, to update field with data, to wait data updating a field, or to set variables. Stimuli are always messages of buffer type (containing string of char), or integer type. Here is an example of protocol:

```
Stimulus: MSG_BUFFER SEND >$Name to >$Area
Reaction: SERVER COMMAND SET Sequential Priority 5
Reaction: WHD MSG_BUFFER where_is <$Name
Reaction: DATABASE MSG_BUFFER where_coord <$Area
Reaction: SERVER WAIT_FOR_UPDATE ( Coord for <$Name > >x1
>y1 >z1
Reaction: SERVER WAIT_FOR_UPDATE ( Coord for <$Area > >x2
>y2 >z2
Reaction: DATABASE MSG_BUFFER compute path for <$Name from
<x1 <y1 <z1 to <x2 <y2 <z2
Reaction: SERVER WAIT_FOR_UPDATE (Path for <$Name > >$Way
Reaction: VHD MSG_BUFFER apply <$Way to <$Name
EndReact.
```

An example of a possible use of such protocol is shown in the next section about integration.

5. Integration and discussion

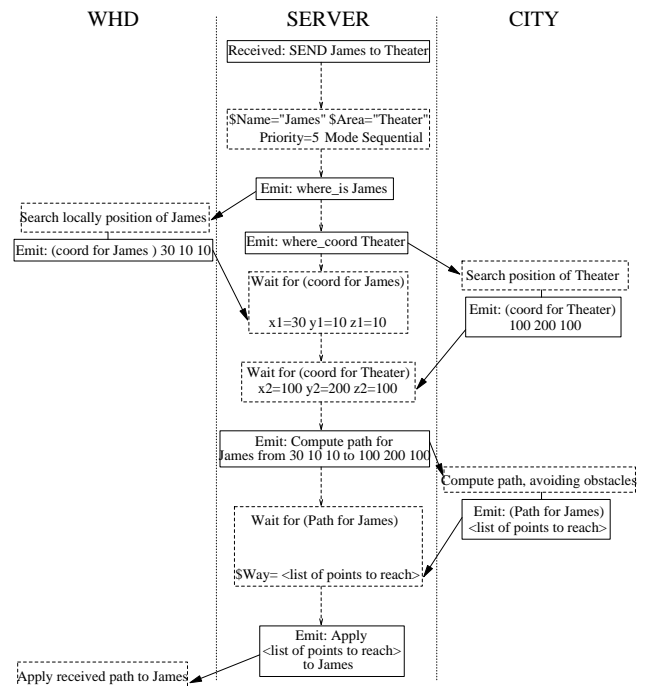


Figure 10. Message exchange during simulation

Figure 10 represents an example of integration and show message exchanges between the WHD application and the CITY via the server. When a client (like WHD) asks to send a humanoid named James to Theatre, SERVER first asks the current position of James to WHD, then the entry of the theatre to the Database client. The client waits until requested information is updated on server by the client that is providing the service. When data is updated, the SERVER gets the location of James. It sends a request to Database for computing a path using these 3D coordinates, and waits for the update of the path field. Once it is done, the path is sent to WHD, which applies it to the virtual human representing James.



Figure 11. View of the city within a WHD session.

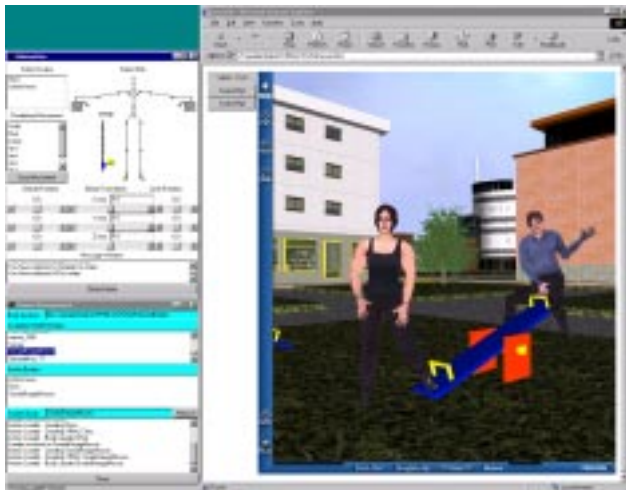


Figure 12. Real-time deformed virtual humans in the city with one of the WHD control panel.

Figure 11 and Figure 12 represent some views of the city with virtual humans. The represented interface concerns the manipulation of a virtual human body handling skeleton joints under WHD.

The database can be extended to a shared database with multiple users. In this case, all the database access

must be provided with a mono control access. All the clients modifying an object location must inform the database when the object is reserved and released.

Another use of the database can be a dynamic construction of a scene. A huge scene such as a city is segmented in a set of different parts (for example quarters, inside a building, a stair etc...) with their associated databases. Active parts of the 3D scene are dynamically loaded and displayed in direct relation with the user interaction (walkthrough, field of view) which is transmitted from WHD to SERVER. SERVER maintains a list of active parts of the city in real time. This list is the result of the information given by the CITY about the part of the scene that is able to be seen by the WHD user. CITY sends the scenes through SERVER to WHD. The principal drawback of such interaction between CITY and WHD is the huge amount of data in transit through the SERVER. Instead of sending the entire scene data, CITY can provide only WEB references on scene parts. This architecture can be enhanced by adding a predicative system, which specifies the neighbourhood of active scene parts. The new parts of the scene that could be soon active are eventually loaded (depending on CPU load) by a set of low CPU consuming threads. With such a system when a neighbour becomes active, the associated part is already loaded and WHD has just to update the scene graph with the new city part.

7. Conclusion and Future Work

In this paper, we have presented an integration of an Informed Environment database with a WEB application via a client/server. This integration is able to provide virtual humans with more knowledge by queering a database dedicated to urban life simulation. This application can have several utilisations using WHD as main application with different plugins or directly as new application. Here are some examples of these uses. A virtual guide can allow intelligent visits of 3-dimensional virtual places with environment knowledge and additional information attached like sounds or link with html pages. A virtual guide is an autonomous humanoid reacting to eventual queries from users. Some autonomous virtual humans can inhabit a virtual place to create a more realistic simulation. A possible more basic use, can be a visit without human representation but still providing information on the location of the user in the scene.

The SERVER with a WEB application such as WHD has the advantage of allowing multiple plugins to provide others services to give to virtual humans more complex behaviours [7]. A crowd module [24] can handle virtual humans with crowd behaviour. Also other applications can provide clients with complex rule based behaviours or have complex interaction with objects [21].

Other evolution of the simulation is the possibility to add in the database information about the sound associated to a place. New perception input could add more realism during the simulation.

Acknowledgements

This research was sponsored by the Swiss National Research Foundation. The authors would like to thank Frederic Sidler for designing the scenes and his collaboration.

References

- [1] A. Antonicelli, G. Sciscio, R. Rosicarelli, G. Ausiello, T. Catarci and M. Ferranini, "Exploiting Pompei culturage Heritage: the Plinius project", *Eurographics'99*, Sept. 1999, short paper pp. 100-103.
- [2] M. A. Aufaure Portier, P. Berthet and J. L. Moreno (1994), "Optimized Network Modelling for Route Planning", *Proceeding of thver1.dvie 4th Eur. Conf. on GIS*, EGIS94 Paris 1994 pp.1817 - 1824.
- [3] C. Babski, MPEG4 player on the Web for HANIM bodies, <http://ligwww.epfl.ch/~babski/StandardBody/mpeg4/>
- [4] C. Babski, WHD web Page, <http://ligwww.epfl.ch/~babski/StandardBody/WHD/>
- [5] C. Babski, D. Thalmann, "A Seamless Shape For HANIM Compliant Bodies", *Proc. VRML 99*, ACM Press, Paderborn Ger., 1999, pp.21-28.
- [6] C. Babski, D. Thalmann, "Real-Time Animation and Motion Capture in Web Human Director", *Proc. Web3D & VRML2000 Symposium*, Monterey CA, to appear Feb 2000.
- [7] A. Caicedo and D. Thalmann, "Intelligent decision-making for virtual humanoids". *5th European conference of artificial life, Workshop Artificial life in virtual environments*, Sept. 99, Lausanne Switzerland.pp.28 - 36
- [8] CosmoPlayer VRML browser. <http://cosmosoftware.com/products/player/brief.html>
- [9] A. Chavez, A. Moukas and P. Maes "Challenger: A Multi-agent System for Distributed Resource Allocation". *Proc. of AA'97*. ACM Press, Feb. 97, Marina del Rey, CA USA, pp. 323-332
- [10] P.R. Cohen, A.J. Cheyer, M. Wang, S.C. Baeg, "An Open Architecture", in AAAI Spring Symposium, pp1-8, March 94.
- [11] S. Donikian (1997) VUEMS : "A virtual Urban Environment Modeling System "; *Computer Animation'97* pp.127-133.
- [12] N. Farenc, S. R. Musse, E. Schweiss, M. Kallmann, O. Aune, R. Boulic, and D. Thalmann, "A paradigm for controlling Virtual Humans in Urban Environment Simulation", to appear in *Applied Artificial Intelligent Journal*, Special Issue on Intelligent Virtual environment, 2000.
- [13] N. Farenc, R. Boulic and D. Thalmann, "An Informed Environment dedicated to the simulation of virtual humans in a urban context", ", *Eurographics'99*, Milan Sept. 1999.
- [14] T. Fuji, K. Imamura, T. Yasuda, S. Yokoi and J. Toriwaki (1995), "A Virtual Scene System for City Planning", *Computer Graphics: Development in Virtual Environments 1995* pp.485-496.
- [15] Funkhouser, A. Thomas, "RING: A Client-Server System for Multi-User Virtual Environments", *ACM SIGGRAPH Special Issue on 1995 Symposium on Interactive 3D Graphics*, Monterey, CA, 1995, pp. 85-92
- [16] D. Guzzoni, A. Cheyer, Julia, Luc and Konolige, Kurt, "Many Robots Make Short Work". SRI International 1996 AAAI Robot Contest.
- [17] HANIM Specification. <http://ece.uwaterloo.ca/~h-anim/>
- [18] K. R. Howard (1997), "Unjamming Traffic with Computers"; *Scientific American* October 1997.
- [19] R. Ingram, S. Benford and J. Bowers (1996), "Building Virtual Cities: applying urban planning principles to the design of virtual environments", *VRST96* pp.83-95.
- [20] W. Jepson, R. Liggett and S. Friedman (1996), "Virtual Modeling of Urban Environments", *Presence* Vol. 5 Winter 1996 p.83-95.
- [21] M. Kallmann and D. Thalmann, "Modeling Objects for Interaction Tasks", *EGCAS'98 - 9th Eurographics Workshop on Animation and Simulation*, 1998, Lisboa, Portugal.
- [22] M. Masuch, B. Freudenberg, B. Ludowici, S. Kreiker and T. Strothotte, "Virtual Reconstruction of Medieval Architecture", *Eurographics'99*, Sept. 99, short paper pp. 87-90.
- [23] P. L. Mokhtarian (1997), "Now That Travel Can Be Virtual, Will Congestion Virtually Disappear?", *Computers Scientific American* October 1997.
- [24] S. R. Musse, F. Garat and D. Thalmann, "Guiding and Interacting with Crowds in Real-Time", *Workshop on Computer Animation and Simulation*, *Eurographics'99*, Milan
- [25] J. Silvetti, C. Smith, T. Dumbleton, S. Williamson and G. Taylor, "Site and Structure at the Vatican: from the earliest settlement to the present", *Eurographics'99*, Sept. 1999, short paper pp. 92-95.
- [26] D. Thalmann, N. Farenc, R. Boulic, "Virtual Human Life Simulation and Database: Why and How", *Dante'99*, Nov. 1999, Japan, to appear.
- [27] VRML 2.0 specifications. <http://www.vrml.org/Specifications/>
- [28] WorldView VRML browser. <http://www.intervista.com/>