



**HAL**  
open science

# VIRTUAL STORIES AUTHORIZING TOOLS FOR PEDAGOGICAL PURPOSE

Jean-Pierre Gerval, Mircea Popovici, Jacques Tisseau

► **To cite this version:**

Jean-Pierre Gerval, Mircea Popovici, Jacques Tisseau. VIRTUAL STORIES AUTHORIZING TOOLS FOR PEDAGOGICAL PURPOSE. International Conference on Computers and Advanced Technology in Education, Jun 2003, Rhodes, Greece. hal-02999137

**HAL Id: hal-02999137**

**<https://hal.science/hal-02999137>**

Submitted on 10 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## VIRTUAL STORIES AUTHORIZING TOOLS FOR PEDAGOGICAL PURPOSE

Jean-Pierre GERVAL (1), Mircea POPOVICI (1,2), Jacques TISSEAU (1)

*(1) Ecole Nationale d'Ingénieurs de Brest (ENIB)  
Technopôle Brest-Iroise – BP 30815 - 29608 Brest cedex - FRANCE  
Tel: +33 (0)2 98 05 66 31, Fax: +33 (0)2 98 05 66 29,  
E-mail: {gerval,popovici,tisseau}@enib.fr*

*(2) OVIDIUS University of Constanta  
124 Boulevard Mamaia – 8700 Constanta – ROMANIA  
Tel: +40 241 618070, Fax: +40 241 618372, E-mail: [dmpopovici@univ-ovidius.ro](mailto:dmpopovici@univ-ovidius.ro)*

### Abstract:

This paper presents two developments that were achieved within the framework of a distributed virtual reality environment for children. This virtual environment supports cooperation among members of a dispersed team engaged in a concurrent context. The first development is a user-friendly interface that enables teachers to fit with children pedagogical requirements and generates new virtual environments according to teacher's specifications. The second tool aims at a better children cooperation by means of avatar's behaviours. These tools have been implemented with Java and VRML languages. The distribution of virtual worlds is obtained using DeepMatrix as environment server.

### Keywords:

Distributed Virtual Reality, Virtual Environments, Virtual Reality Modeling Language, Java, External Authoring Interface.

## 1. INTRODUCTION

This paper describes tools that were developed in the framework of a pedagogical project called EVE: Environnements Virtuels pour Enfants (Virtual Environments for Children).

The EVE application has already been presented in [1]. In this previous paper, we were describing the virtual world and its implementation.

Nowadays our new target is to describe the way teachers can create any kinds of stories according to children pedagogical needs.

We also present new developments that make children communication easier by means of using avatar's behaviours.

## 2. THE PEDAGOGICAL CONTEXT

The EVE application has been developed in order to help primary school children to learn reading. It offers children an attractive approach of learning through two games that implements emulation and cooperation.

This first game is a self-training step according to a global learning method of reading. The target is to make sentences where a picture illustrates each sentence (Fig. 1).

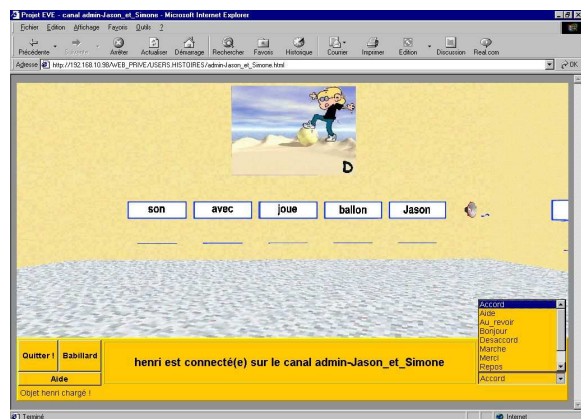


Fig. 1. First game: sentence and picture

The child must discover the sentence using a disordered set of words and moving words to the right places.

The second game is carried out in a common virtual room (Fig. 2) where children will meet each other. Each child owns a personal avatar. A team of three children is supposed to build a story using the pictures won previously.

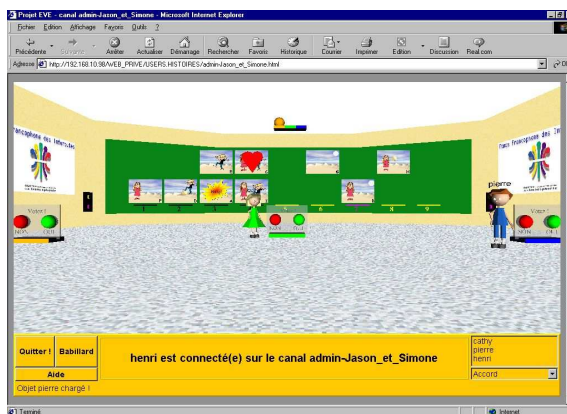


Fig. 2. Second game in a common virtual room

## 3. DESIGNING STORIES

### 3.1 The Story

A story is defined by an ordered set of nine sentences. As previously mentioned, a picture illustrates each sentence. A sound file is also linked to the sentence in order to enable children to hear the sentence.

A sentence is defined by an ordered set of words. As it is presented in figure 1, a picture is associated to each word.

The story data model is presented hereafter (Fig. 3).

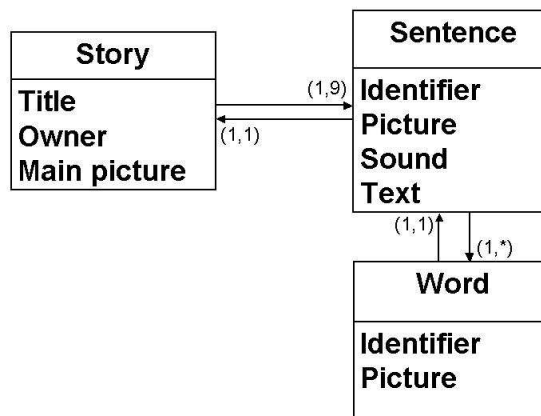


Fig. 3. Story data model

According to this data model, which is really simple, we have chosen to implement this model using a simple directory structure.

### 3.2 The User Interface

Our challenge is to provide teachers with a user-friendly interface (Fig. 4) (Fig. 5) that enables them to build their own stories.

Using this tool, teachers can create stories that fit children pedagogical requirements all along the school year.

Teachers can also involve children in designing new stories for their friends.

For each sentence, the teacher will choose:

- A gif file (the picture)
- A wav file (the sound)

and write the text of the sentence.

When this job ended, the teacher **saves** the story on the server. Before to proceed to data transfer, the application, automatically:

- Generates a gif file for each word taking into account word length in order to use the best characters font size;
- Transforms each wav file into a mp3 file in order to reduce the amount of data.

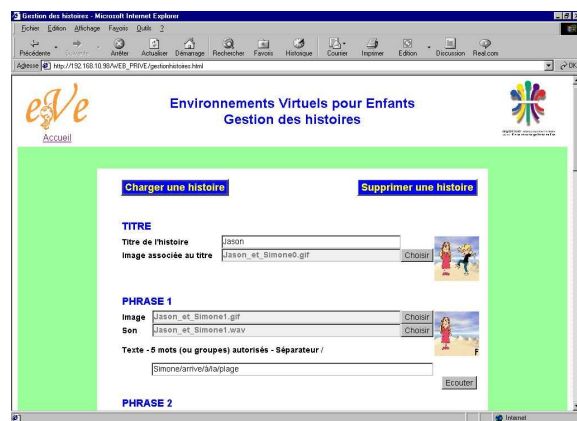


Fig. 4. User interface: top

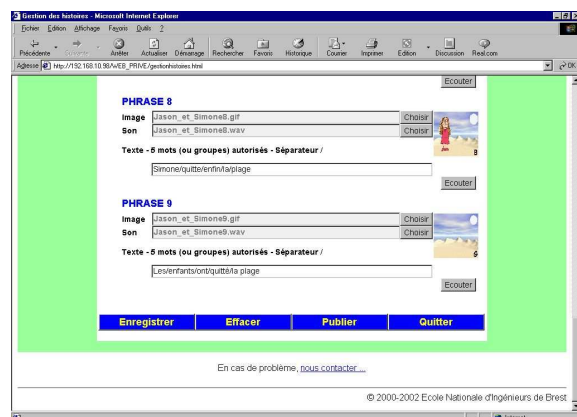


Fig. 5. User interface: bottom

The teacher can **clear** the interface, **load** or **delete** a story from the server, previously created.

A new virtual world that implements the new story is generated when the teacher decides to **publish** its story. On the server side, symbolic links to the virtual world are created in the new story directory structure.

At the same time, an E-mail is sent to a moderator who will read the new story and decide whether he accept it or not according to its content.

A second interface (Fig. 6) enables children or teachers to choose a virtual world.



Fig. 6. User interface: choosing a virtual world

The list of all published stories is showed on the left up corner of the page. The user:

- Chooses a story in the list;
- May click on the picture to view all pictures;
- Can press a button to read sentences - sentences disappear when the mouse is released.

This interface enables the teacher to read the story before he starts working with children.

Both user interfaces have been implemented in Java language with a special attention due to the fact that users are viewing the pages with Internet Explorer and running the Microsoft Virtual Machine.

## 4. AVATAR'S BEHAVIOURS

### 4.1 Pedagogical point of view

A first idea when developing the EVE project was to implement a chat that enables children to discuss and explain their choices especially during the second game. Previous experiments demonstrate that this choice was not suitable for inexperienced in reading and writing learners. Actually, teachers mainly used the chat!

We have decided to implement avatar's behaviours (Fig. 7) so that:

- It increases the realism of avatars that are virtual representations of children playing in a virtual world;
- It induces a better communication between children.

The implemented behaviours are:

- Avatar's realism: **Walk, Standby,**
- Children communication: **Hello, Help me! Thank you, I agree, I disagree, Laughing, Smiling, Good Bye.**

These previous lists are not exhaustive and should be easily completed according to current experiments with teachers and children.



Fig. 7. Avatar's behaviours

The child may select a behaviour, which will be broadcasted (gesture and sound) to other children through its personal avatar.

#### 4.2 Technical point of view

The server implementation is based on the DeepMatrix software [2] from GEOMETREK. This software enables users to enter 3D websites where they can interact with other users and objects. DeepMatrix implements client-server architecture (Fig. 8).

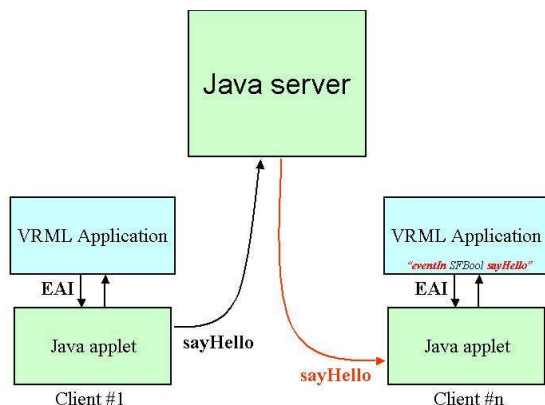


Fig. 8. Client-server architecture

Clients are Java applet running in a HTML Browser. Java applet loads VRML (Virtual Reality Modeling Language) [3] contents into the Browser VRML plug-in and adds avatar representation to the virtual

world. The communication between VRML world and Java applet is made by use of the EAI (External Authoring Interface).

Avatars are implemented in VRML (Fig. 9) using a PROTO structure. Behaviours are designed with the 3DStudioMax software. This software enables to export data into VRML format. The resulting code is then inserted into the avatar PROTO structure.

When a child selects a specific behaviour the Java applet sends a message to the server (Fig. 8). The server broadcasts an event for example: "sayHello" to all other clients Java applet. EAI enables a two-way communication between the Java applet and the VRML plug-in. The PROTO structure receives an eventIn for example: "triggerHello".

```

PROTO OrangeBoyAvatar
{
  ...
  eventIn SFBool triggerHello
  ...
}
{
  ...
  # Insert here 3DStudioMax VRML code
  # that describes avatar animations, for example:

  DEF rightHand Transform
  {
    ...
    DEF rightHandTimer TimeSensor {cycleInterval 2 }
    ...
    DEF rightHandInterpolator PositionInterpolator
    {
      ...
    }
    ROUTE rightHandTimer.fraction_changed
    TO rightHandInterpolator.set_fraction
    ROUTE rightHandInterpolator.value_changed
    TO rightHand.set_translation
    ...
  }
}
DEF S Script
{
  ...
  eventIn SFBool helloBehaviour IS triggerHello
  eventOut SFTIME helloTime
  ...
  url "vrmlscript:
  function helloBehaviour (value, eventTime)
  {
    helloTime = eventTime;
  }"
}
...
ROUTE S. helloTime TO rightHandTimer.startTime
...
OrangeBoyAvatar {}

```

Fig. 9. Avatar VRML structure

According to the previous example, when the eventIn “**triggerHello**” occurs, we need to catch its date in order to start avatar animation. This task is performed through the “**helloBehaviour**” function that returns the eventOut value “**helloTime**”.

Finally, the date is routed to VRML objects that are involved in the selected behaviour:

“ROUTE S. **helloTime** TO **rightHandTimer.startTime**”.

Remember that these objects were generated using the 3DStudioMax software.

## 5. CONCLUSIONS AND FUTURE WORKS

This application is accessible with a standard Web Browser (Internet Explorer) and VRML plug-in without any specific or additional software:

<http://www.enib.fr/eve>

and is currently tested with teachers and children from nine French primary schools.

On a pedagogical point of view, the user interface offers new perspectives such as the learning of foreign languages.

On a technical point of view, current experiments will help us to design and implement new avatar behaviours according to end users needs.

We are also working on a more sophisticated mechanism, which is the integration of streaming audio and video into the virtual world. This should make increase the performance of children cooperative work.

Virtual Reality enables new products developments that will help teachers and children in their tasks of teaching and learning through friendly interfaces.

## ACKNOWLEDGEMENTS

Thanks are due to:

- The Intergovernmental Agency of the Francophonie, which had funds the EVE projetct in the framework of the Francophone Information Highway Fund;
- The Brittany Regional Council, which funds this work in the framework of the MEGALIS Project.

## REFERENCES

- [1] J.P. Gerval, M. Popovici, M. Ramdani, O. El Kalai, V. Boskoff, J. Tisseau, *Virtual Environments for Children*, IASTED International Conference, Computers and Advanced Technology in Education, CATE 2002, Cancun, Mexico, May 20-22, 2002
- [2] Gerhard Reitmayr, Shane Carroll, Andrew Reitmeyer, Michael G. Wagner, *DeepMatrix – An Open Technology Based Virtual Environment System*, White Paper, October 30, 1998, <http://www.geometrek.com/developers/>
- [3] VRML Standard Version 2.0, ISO/IEC CD 14772, 1996, <http://vrml.org/VRML2.0/>