

# Integration of Simulation and Multimedia in Automatically Generated Internet Courses

Manuel Alfonseca, Juan de Lara  
Dept. Ingeniería Informática, Universidad Autónoma de Madrid  
Campus de Cantoblanco, 28049 Madrid  
*E-Mail: {Manuel.Alfonseca, Juan.Lara}@ii.uam.es*

**Key words:** Continuous simulation, Internet, Multimedia, Automatic generation of courses, Java

**Abstract:** This paper describes the automatic generation of simulation-based Internet courses by means of an object-oriented continuous simulation language (OOC SMP), and a compiler for this language (C-OOL). Several multimedia extensions added to the language are also described. These extensions provide the student with a better understanding of the simulated models. The paper describes finally a course developed using the multimedia extensions.

## Introduction

Systems simulation [1] is one of the older branches of Computer Science: it was already advanced in the sixties and reached maturation in the seventies. Continuous simulation has been programmed traditionally either in special purpose languages, or in general purpose programming environments. According to their syntax, there are three main classes of continuous simulation languages: block languages [2], mathematically-oriented languages (such as *CSMP* [3]) and graph languages [4, 5].

Internet is becoming an important educational tool. New courses appear every day, [6,7], reaching from a simple translation of classroom notes, to more advanced materials including sophisticated elements, such as simulations, animated graphics, etc. In particular, the Java language [8] has made the courses more interactive, faster in

execution and easily transportable. The obvious interest of the field has created a need of appropriate tools to help in course elaboration, which should provide all the facilities offered by the Internet-based education [9,10].

The integration of simulation tools with multimedia elements makes it possible to express in a richer way the knowledge we are trying to provide, and gives the student a better comprehension of the problem, as explanations by means of video, images, texts, etc. are not statically included in an *HTML* page, but are dynamically synchronized with the simulation.

In his paper we describe the multimedia extensions added to our continuous simulation language, together with a course on Ecology that incorporates these elements.

### **The OOCSMP language**

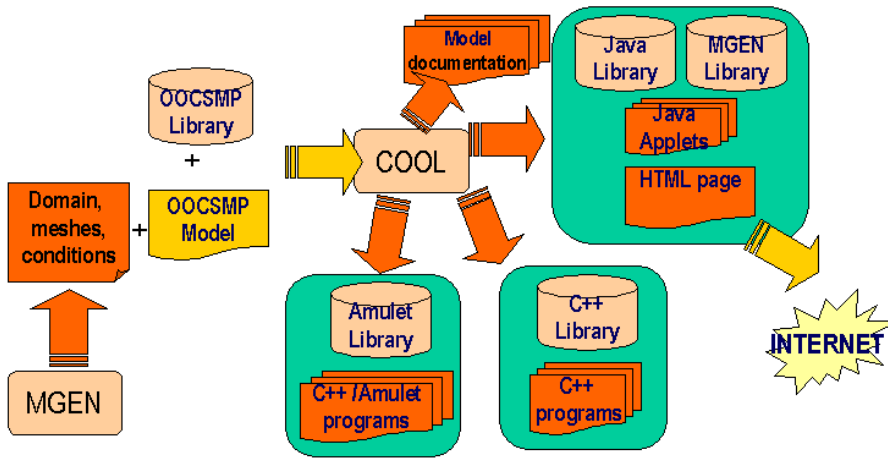
We have designed the *OOCSMP* language [11-13] as an object-oriented extension of the *CSMP* language. It is a true extension, meaning that *CSMP* programs are correctly compiled and executed with our *C-OOL* compiler. The object-oriented extensions added to the language make it possible to build very compact object-oriented models when the system to be simulated consists of a number of similar inter-actuating components.

The *OOCSMP* language has been used to build several courses in the web: a course on Newton's gravitation and the solar system, a course on Ecology [14], a basic course on Electronics [15], and a course describing the capacities of the language to solve partial differential equations [16]. All of them are accessible from the following address:

*[www.ii.uam.es/~jlara/investigacion](http://www.ii.uam.es/~jlara/investigacion)*

### **The C-OOL compiler**

*C-OOL* (a Compiler for the OOCSMP Language) is a compiler we have built to translate the *OOCSMP* language into C++ or Java. Its structure and the way it works is shown in figure 1:



**Figure 1:** Outline of the use of the C-OOL compiler.

Both the compiler and the Java libraries used at execution time have been extended to support the four multimedia primitives described in the following section.

### Multimedia extensions to the *OOCSMP* language

We have added several multimedia extensions (images, video, audio and text) to the output formats associated to the *OOCSMP* language. They are described as follows:

- Image panel: A panel or window is displayed where several images may be shown subsequently. Each image appears when a certain condition is true. A condition may be any *OOCSMP* logic expression, using the logic operators  $<$ ,  $>$ ,  $<=$ ,  $>=$ ,  $\parallel$  (logic OR),  $\&\&$  (logic AND), which have been added to the language. The *OOCSMP* syntax of an image panel declaration is:

```
IMAGEPANEL [position]
    (, START( <OOCSMPexpression>), "file")+
    [, DEFAULT , "file" ]
```

Where `position` is one of the nine positions in the main panel ( $N$ ,  $S$ ,  $E$ ,  $W$ ,  $NE$ ,  $NW$ ,  $SE$ ,  $SW$ ) or *WINDOW*, which is shown in a separate window. The image in "file" appears in the indicated

position when condition `<OOC SMPexpression>` becomes true during the simulation. As many image-condition pairs as desired may be specified. If the `DEFAULT` clause is given, the associated file is shown when none of the other conditions holds. Several image panels may be created. Image formats accepted are gif and jpeg.

- Video panel (`VIDEOPANEL`): The same syntax as the previous, but videos are shown instead of images. Java Media Framework v2.0 [17] is used to show the video. Formats accepted are mov and avi.
- Audio (`AUDIO`): Similar syntax as the two previous, although in this case there is no associated graphic object, thus a position must not be specified. Java Media Framework v2.0 is also used here. This element accepts audio files in wav format.
- Text panel (`TEXTPANEL`): The same syntax as `IMAGEPANEL`.

### **A course on Ecology with multimedia elements**

Our course on Ecology [14] has been modified to incorporate the multimedia extensions, which have been specially useful in the pages where one species invades the ecosystem. Using them, it is possible to provide the student with detailed explanations of what is happening in the appropriate moments.

We shall add multimedia elements to the ecosystem model which displays a predator, a herbivore and a primary producer in a total equilibrium which is broken by the invasion of a new predator. This model is explained in detail at [14].

When the predator invasion takes place, predator and herbivore populations get smaller (there is more competition and more predators) while the population of primary producers increases until a new state of oscillating equilibrium is reached.

We will use the multimedia elements to underline three phases in the simulation:

1. From the beginning until the predator invasion (static equilibrium).

2. From the predator invasion until the oscillating equilibrium.
3. After the oscillating equilibrium has been reached.

We have to provide conditions that identify the three phases:

1. The first phase is defined by the condition `TIME>=0` and `TIME<50`, because the new predator invades the ecosystem at `TIME=50`. The associated multimedia panels should thus include the condition:

```
START ( (TIME >=0)&&(TIME<50) )
```

2. The second phase happens when `TIME > 50` and while the derivative of the predators and herbivore populations are negative (all three populations are getting smaller). Thus, the condition would be:

```
START( (TIME>=50)&& (Lion.XP<0)
      &&(Cheetah.XP<0)&&(Gnu.XP<0) )
```

3. Oscillating equilibrium is reached when none of the two previous conditions holds, i.e. we can use the `DEFAULT` clause.

Now we should prepare the multimedia elements to be used by our model. In our case, we shall add a text panel and an image panel.

Listing 1 shows the *OOC SMP* model. Figure 2 shows the state of the simulation before the predator invasion, and figure 3 displays the situation after the predator invasion, but before the ecosystem reaches oscillating equilibrium.

The model used in previous versions is the same, except for the declaration of the two new panels. As the figures show, the resultant applet is divided into 4 areas:

- To the northwest, a panel showing the populations of the four species in the ecosystem in the form of a bidimensional plot.
- To the southwest, a panel showing the percentage of the four populations in the form of an iconic graphic.
- To the northeast, a multimedia panel showing the trophic

chains in the different phases.

- To the southeast, a text explaining what is happening in the different phases.

This applet, together with the other pages in the course on Ecology, can be found at the following Internet address:

<http://www.ii.uam.es/~jlara/investigacion/ecommm/africa1.html>

```
TITLE Three species , invasion of predator
INCLUDE "Species.csm"
* Actual species
Species Cheetah("Cheet", "http://www.ii.uam.es/~jlara/wcat.gif",
4,-.028,.0014,.0,50)
Species Lion ("Lion", "http://www.ii.uam.es/~jlara/lion.gif",
2,-.02, .001)
Species Gnu ("Gnu", "http://www.ii.uam.es/~jlara/bovin.gif",
20,-.02,.0001,.016666666)
Species LGrass ("LGrass", "http://www.ii.uam.es/~jlara/leafs.gif",
400, .01, 0, .0005)
Species Ecosystem := Cheetah, Lion, Gnu, LGrass
Ecosystem.STEP()
Cheetah.ACTION(Gnu, 1, 1)
Lion.ACTION (Gnu, 1, 1)
Gnu.ACTION (Lion, - .6, 0)
Gnu.ACTION (Cheetah,- .4, 1)
Gnu.ACTION (LGrass, 1, 1)
LGrass.ACTION (Gnu, - 1, 1)
* Timer and show data
TIMER delta:=0.01,FINTIM:=900,PRdelta:=.5,PLdelta:=5
PLOT [N], Cheetah.X, Lion.X, Gnu.X, LGrass.X, TIME
ICONICPLOT [C], Cheetah.X, Lion.X, Gnu.X, LGrass.X
TEXTPANEL [E], START((TIME>=0)&&TIME<50), "inicio.txt",
START((TIME>=50)&&(Lion.XP<0)&&(Cheetah.XP<0)&&(
Gnu.XP<0)), "inv.txt",
DEFAULT,"equilib.txt"
IMAGEPANEL [NE], START((TIME>=0)&&TIME<50), "inicio.gif",
START((TIME>=50)&&(Lion.XP<0)&&(Cheetah.XP<0)&&(
Gnu.XP<0)), "inv.gif",
DEFAULT,"equilib.gif"
METHOD ADAMS
```

**Listing 1:** A model of an ecosystem invaded by a new predator

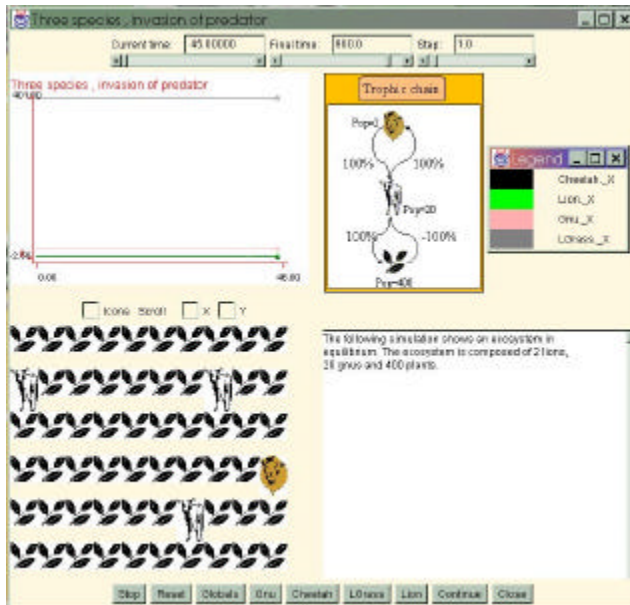


Figure 2: State of the simulation while the first condition holds

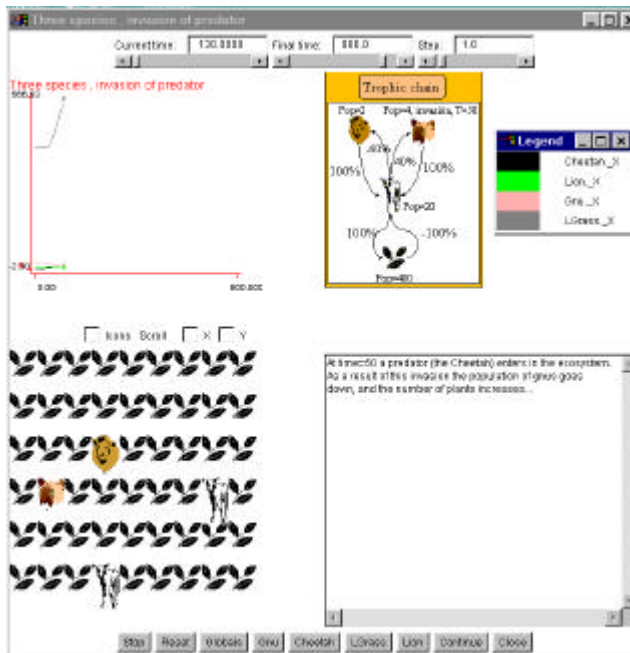


Figure 3: State of the simulation while the second condition holds

## Conclusions and Future Work

The inclusion of multimedia elements in our language enhances student comprehension of the simulated model. The mechanisms that synchronize the multimedia elements with the simulation (the clauses `START` and `DEFAULT`) make it possible to display each element in the appropriate moment, thus holding the student attention better than a static presentation in *HTML* pages separated from the simulation.

The cost of adding multimedia elements to a course generated with *OOC SMP* is very small. One has just to identify the appropriate conditions associated to each of them, and prepare the images, video, texts, etc., the remainder of the model is not changed.

It is also unnecessary to modify by hand the *HTML* pages generated by the compiler, as those elements previously included in those pages can now be introduced as dynamic multimedia elements inside the simulation applet.

In the future, more multimedia elements will be added to the language, such as *HTML* panels and animation. The student activities may also be monitorized either through a data base or with special control task control mechanisms [18]. The system will also be enhanced with a graphic environment to build *OOC SMP* models, to plan the course pages, and to synchronize the multimedia elements. With this environment, *OOC SMP* and *C-OOL* would become full-fledged author tools.

We are starting to work with distributed objects, using *RMI*. *C-OOL* will generate distributed code for several machines. In this way, the performance of some simulations will be substantially improved.

## References

- [1] Monsef, Y. 1997. "*Modelling and Simulation of Complex Systems*", SCS Int., Erlangen.
- [2] Alfonseca, M. 1974. "*SIAL/71, a Continuous Simulation Compiler*", in "*Advances in Cybernetics and Systems*", Ed. J. Rose, Gordon and Breach, London, Vol. 3, 1974, 1319-1340.



- [3] IBM Corp. 1972. *"Continuous System Modelling Program III (CSMP III) and Graphic Feature (CSMP III Graphic Feature) General Information Manual"*, IBM Canada, Ontario, GH19-7000, 1972.
- [4] Karnopp, D. *"Bond Graph Models for Electrochemical Energy Storage: Electrical, Chemical and Thermal Effects"*, Journal of the Franklin Institute, Vol 324, 1990, pp. 983-992.
- [5] Legasto A.A. Jr., Forrester, J.W., Lyneis, J.M. editors, *"Systems Dynamics"*, North Holland, 1980.
- [6] Thomson Publishing. 1997. *Internet Distance Education with Visual C++*. <http://www.thomson.com/microsoft/visual-c/teacher.html>
- [7] GNA The Globewide Network Academy. 1997. <http://www.gnacadey.org>.
- [8] <http://java.sun.com>
- [9] Aviation Industry CBT Committee Computer Managed Instruction. 1977. *Computer Managed Instruction Guidelines and Recommendations*, AGR 006, Version 1.1, AICC. <http://www.aicc.org/agr006.htm>.
- [10] Schutte. 1997. *Virtual Teaching in Higher Education: The New Intellectual Superhighway or Just Another Traffic Jam?*. In Internet at: <http://www.csum.edu/sociology/virexp.htm>
- [11] M. Alfonseca, E. Pulido, J. de Lara, R. Orosco. 1997. *"OOCSMP: An Object-Oriented Simulation Language"*, Proc. 9th European Simulation Symposium ESS'97, pp. 44-48, 1997.
- [12] M. Alfonseca, J. de Lara, E. Pulido. 1998. *"Semiautomatic Generation of Educational Courses in the Internet by Means of an Object-Oriented Continuous Simulation Language"*, Proc. 12<sup>th</sup> European Simulation Multiconference ESM'98, pp. 547-551, 1998.
- [13] M. Alfonseca, R. Carro, J. de Lara, E. Pulido. 1998. *"Education in Ecology at the Internet with an Object-Oriented Simulation Language"*, Proc. Eurosim'98 Simulation Congress, Fed. European Simulation Societies, ed. K.Juslin, pp. 118-123, 1998.
- [14] Alfonseca, M., de Lara, J. and Pulido, E. 1998. *"Educational simulation of complex ecosystems in the World-Wide Web"*, Proceedings ESS'98, Nottingham, pp. 248-252.
- [15] Alfonseca, M., de Lara, J. 2000. *Automatic generation of a web course on electronics with associated documentation*, to appear in EUROMEDIA'00
- [16] de Lara, J., Alfonseca, M. 1999. *Simulating Partial Differential Equations in the World-Wide-Web*. EUROMEDIA'99. Munich, pp 45-52.
- [17] <http://java.sun.com/products/java-media/jmf/2,0>
- [18] Alfonseca, M., García, F., de Lara, J., Moriyón, R., *"Generación automática de entornos de simulación con interfaces inteligentes"*, ADIE n° 12, Octubre-Diciembre 1998.