VISUALIZATION OF FOLKTALES ON A MAP BY COUPLING DYNAMIC DEVS SIMULATION WITHIN GOOGLE EARTH

Jean-François Santucci and Laurent Capocchi

SPE UMR CNRS 6134, University of Corsica, Campus Grimaldi, 20250, Corte, France

Keywords: DEVS, Discrete Event Simulation, Google Earth API, Dynamic variable structures.

Abstract: This paper deals with dynamic visualization of folktales on a map. The visualization is performed using a coupling of dynamic simulation with the Google Earth API. The simulation is based on the DEVS (Discrete EVent system Specification) formalism. We have defined a set of basic models to perform visualization of folktales using a software framework called DEVSimPy. DEVSimPy is a framework dedicated to DEVS modeling and simulation of complex systems. We also present in this paper a validation of the developed basic models based on folktales coming from the Corsican mythology.

1 INTRODUCTION

This paper deals with spatial visualization of folktales. This work is part of a global project (Santucci & De Gentili, 2009 ; Santucci, De Gentili & Thury-Bouvet, 2010) concerning the study of myths according to the method defined by Claude Levi-Strauss (Levi-Strauss, 1955) when dealing with Structural Anthropology. The goal is to offer anthropologists the possibility to visualize on a map a dynamic progress of the story involved by a given myth. The visualization is based on a Discrete EVent system Specification simulation (DEVS) dynamically linked with the Google Earth API (Brown, 2006). The interest of such a work for anthropologists is to be able to visualize a story stemming from a myth obtained after a DEVS structural anthropology simulation. From a computer science point of view the interest is to define a generic way to propose dynamic visualization of a given story. We choose to perform the dynamic features of the problem of visualization using the DEVS formalism. The spatial visualization on a map has been done using the Google Earth API. We describe in detail in this paper how a dynamic coupling between DEVS formalism and Google Earth has been realized. The proposed approach has been implemented using the DEVSimPy framework in order to develop a dynamic library of DEVS models. We have validated this library using a story stemming from a myth belonging to the Corsican oral culture. The outline of the paper is as follows: we first give in section 2 the background of the research. Section 3 deals with dynamic visualization of folktales. First an overview of the proposed approach is described. Then we explain how we implemented the simulation of variable dynamic structures which are used in order to perform dynamic visualization of folktales on a Google Earth map. Then we detail the description of the dynamic relationship between the DEVSimPy library of models and the Google Earth API. The concluding remarks and future work are given in section 4.

2 BACKGROUND

We present in this section the background of the work that is the main contribution of this paper. The dynamic visualization of folktales belongs to a method called Structural Anthropology (Levi-Strauss, 1955). We first briefly introduce in this section how to perform structural anthropology using a computer science approach. The DEVS formalism is summarized in sub-section 2.2. The DEVSimPy framework is detailed in sub-section 2.3.

2.1 Structural Analysis of Myths

The structural analysis of myths has been defined by Claude Levi Strauss in his books. He specially explained how to apply his method in his famous book series (Levi-Strauss,1969; Levi-Strauss,1971; Levi-Strauss,1978; Levi-Strauss, 1981). The method

Santucci J. and Capocchi L..
VISUALIZATION OF FOLKTALES ON A MAP BY COUPLING DYNAMIC DEVS SIMULATION WITHIN GOOGLE EARTH.
DOI: 10.5220/0003596901280133
In *Proceedings of 1st International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (SIMULTECH-2011), pages 128-133
ISBN: 978-989-8425-78-2

leans on the concept of myth transformation.

Myth transformation consists in performing the generation of myths from a first “reference” myth as Claude Levi Strauss calls it. The generation involves a set of basic transformations that are applied on the mythems of a given myth. A mythem has been defined in 1955 (Levi-Strauss, 1955) as a sentence of a given myth composed by a term and a function. For example the following sentence:” the ogre lives near Casta” is described by the term “ogre” and the function “lives-near-Casta”. All the sentences of a given myth can be expressed by a set of mythems. Then a set of transformations can be applied in order to generate a new myth. Seven basic transformations have been defined by Claude Levi-Strauss: homology, symmetry, opposition, homology, canonical formula, suppression of mythems, and addition of mythems. A set of work associating computer sciences and structural analysis of myth has been proposed in the past towards the objective of myth transformation modeling (Jason & Segal,1977) ; Richard & Jaulin, 1971). These works are attempts to formalize the analysis of tales developed by Claude Levi Strauss. These approaches are based on computer sciences in order to perform systematic myth transformations software. In order to find a solution to this problem we have defined a software myth transformation approach (Santucci and De Gentili, 2009; Santucci, De Gentili and Thury-Bouvet, 2010) based on the DEVS formalism which is summarized in sub-section 2.2. The structural anthropology of myth as defined by Claude Levi Strauss also proposes the analysis of a given myth according to his spatial representation in the landscape. Since it is issued from oral cultures, a myth is told and retold in front of people. Usually a myth involves a set of places well known by the people who are listening to the story. When the story is told, the landscape linked with the myth is mentally viewed by the listeners. In this paper we describe in detail how we are able to propose the dynamic visualization of a myth on a map using a computer.

2.2 DEVS Formalism

We briefly introduce the DEVS formalism (Zeigler, Praehoffer & Kim, 2000). In the DEVS formalism, one must specify: 1) basic models from which larger ones are built, and 2) how these models are connected together in hierarchical fashion.

An atomic model allows specifying the behavior of a basic element of a given system.

Basic models (called Atomic Models) are defined by the following structure:

$$AM = < X, S, Y, C, \delta_{ext}, \boldsymbol{\delta_{int}}, \lambda, ta >$$

Where,

- *X* is the set of input values,
- *S*: is the set of sequential states,
- *Y*: is the set of output values,
- $\boldsymbol{\delta_{int}}$, is the internal transition function dictating state transitions due to internal events,
- δ_{ext} the external transition function dictating state transitions due to external input events.
- λ is the output function generating external events at the output, and
- *ta* is the time-advance function which allows to associate a life time to a given state.

Connections between different atomic models can be performed by a Coupled Model (*CM*) (Zeigler, Praehoffer & Kim, 2000):

$$CC = < X, Y, D, \{M_d \ / \ d \in D\}, IC, EIC, EOC>$$

Where,

- X is the set of input values,
- Y is the set of output values,
- D is the set of model references, that is to say a set of names associated to the model’s components
- $\{M_d \ / \ d \in D\}$ is the set of coupled model’s components,
- IC, EIC and EOC define the coupling structure in the coupled system (IC defines the internal coupling; EIC is the set of external input coupling; EOC is the set of external output coupling.

A simulator is associated with the DEVS formalism in order to exercise instructions of a DEVS model to actually generate its behavior. The architecture of a DEVS simulation system is derived from the abstract simulator concepts (Zeigler, 1990).

2.3 DEVSimPy Framework

DEVSimPy (stand for DEVS simulator in Python language) is a collaborative Modeling and Simualtion (M&S) software. It is used in order to model and simulate complex systems based on the DEVS formalism in Python programming language.

The initial idea of DEVsimPy is to develop a Graphical User Interface based on the wxPython library (Sanner, 1999) around the PyDEVS kernel (Bolduc & Vangheluwe, 2001).

PyDEVS is an API of the modeling and simulation algorithms of DEVS implemented in python language. WxPython is a blending of the wxWidgets C++ class library with the Python

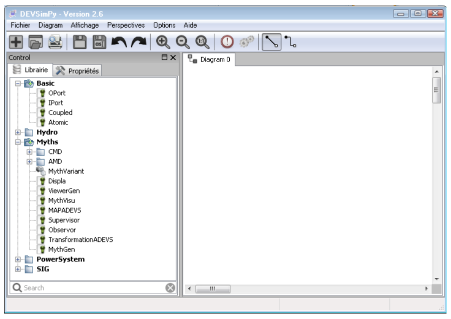


Figure 1: The DEVSimPy framework interface.

programming language.

The DEVSIMPY framework provides a friendly interface for discrete event modeling and simulation by integrating the basic classes of the PyDEVS kernel. Figure 1 gives an example of the window interface. You may notice in figure 1 that the window is split into two parts:

- The left part allows the user to visualize the classes which can be instantiated (using a drag and drop).
- The right part is dedicated to the design of coupled models with a simple drag and drop of classes belonging to the left part of the window.

We used this DEVSimPy framework in order to initiate a research concerning structural analysis of myths(Santucci & De Gentili, 2009 ; Santucci, De Gentili & Thury-Bouvet, 2010). The next step of this work concerns the visualization on a map of a given myth belonging to the set of myths already generated. This step is the main contribution of this paper and explained in section 3.

3 DYNAMIC VISUALIZATION OF FOLKTALES

The goal is to offer the possibility to dynamically visualize on a map the unfolding of a given myth. In this section after having pointed out the interest of such a study for the anthropologists we detail the proposed approach. We particularly explain how we have been able to deal with: (i) DEVS variable dynamic structures, (ii) an association of the Google Earth API and (iii) the DEVSimpY framework and dynamic visualization using Google Earth API.

3.1 Interest and Proposed Approach

The study of folktales is an important task belonging to the anthropologist domain. As we presented in section 2.1 one of the main contributor to this kind of study is Claude Levi Strauss. It is common for an anthropologist to study each myth individually by analysing a given myth according to its contextualization in the landscape. The goal is:

- To visualize the unfolding of a folktale own to the story skeleton
- To sequence the story on a map which summarizes the flow of events.

We decide to combine the simulation of models involved in the DEVSimPy framework (see section 2.3) with some features of the Google Earth API in order to perform the dynamic visualization of a given folktale on a map. The following problems have been solved:

- simulation of variable dynamic structures using DEVSimPy

- activation of Google Earth from the DEVSimPY framework

- dynamic refreshment on the map while the story skeleton is being simulated.

3.2 DEVS Modeling Overview

In order to perform the dynamic visualization of a folktale we have defined a DEVS modeling scheme by implementing the following set of specific atomic models: (1) Viewergen; (2) SIGviewer; (3) MythVisu; (4) PointMyth.

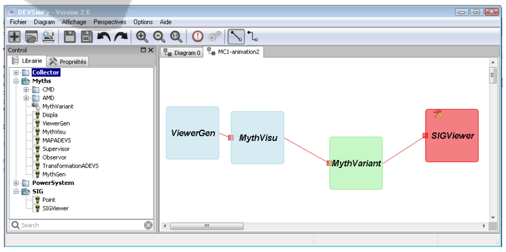
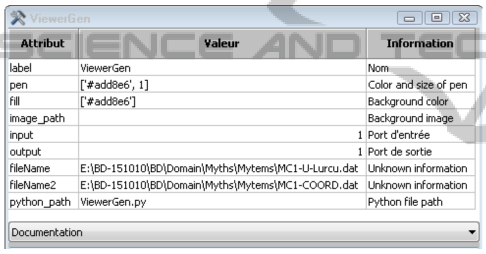


Figure 2: DEVS modeling for dynamic visualization.

We also defined a coupled model called MythVariant that is used as the initial variable structure, which will evolve when a new mythem has to be visualized on a map. Figure 2 presents the DEVS modeling scheme involving four DEVS models: the three atomic models (Viewergen, MythVisu and SiGviewer) as well as the coupled model MythVariant. Furthermore instances of the PointMyth atomic model will be dynamically added into the MythVariant coupled model when a new mythem is encountered by reading a myth.

The inter-connection between the four DEVS models is given in figure 2. The goal of the atomic model called ViewerGen consists in: (i) scanning a text file corresponding to the given myth under study; (ii) sending a message on its output for every mythem belonging to the given myth, where the message contains the term and the function involved in the mythem as well as the location of each of the mythems belonging to the myth. This information is located in the two files associated to the ViewerGen atomic model as seen on figure 3 (the files can be loaded using the two attributes: filename (term and function of the mythems) and filename2 (location of each mythem). The Viewergen atomic model is connected to the MythVisu atomic model as it may be seen in figure 2. The MythVisu atomic model is in charge of the management of variable dynamic structures: for each message received on its input port, the MythVisu atomic model is able to add an instance of the PointMyth atomic model in the MythVariant coupled model. A message containing the location of the point associated with the current mythem (as well as the term and the function involved by the mythem) is then sent to the SIGviewer atomic model. The implementation of the MythVisu atomic model is detailed in sub-section 3.3.


ViewerGen

Attribut	Valeur	Information
label	ViewerGen	Nom
pen	['#add8e6', 1]	Color and size of pen
fill	['#add8e6']	Background color
image_path		Background image
input	1	Port d'entrée
output	1	Port de sortie
fileName	E:\BD-151010\BD\Domain\Myths\Mytems\MC1-U-Lurcu.dat	Unknown information
fileName2	E:\BD-151010\BD\Domain\Myths\Mytems\MC1-COORD.dat	Unknown information
python_path	ViewerGen.py	Python file path

Figure 3: Properties of the MythVisu atomic model.

The goal of the SIGviewer atomic model is to: (i) open the Google Earth window; (ii) print the point corresponding to the current mythem on the map. The implementation of the SIGviewer atomic model is detailed in sub-section 3.4

3.3 Variable Dynamic Structures

In order to perform dynamic visualization of a given myth we had to implement a way to perform DEVS simulation of variable dynamic structures. In order to deal with the modeling of dynamic variable structure system a set of scholars have proposed in the recent past to extend the DEVS formalism (Barros, 1997; Hu, Zeigler, Mittal,2005; Baati, Frydman, Giambiasi, 2007; Hui & Wainer, 2006 ; Barros, 2003). Our approach leans on these previous work and supports changes in structure by the introduction of a special atomic model that keeps in its internal state the structure of a network of models. Changes in the state are automatically mapped into changes in structure. We have called MythVisu the special atomic model in charge of the management of dynamic variable structures in the case of this application.

Figure 4 illustrates the empty DEVS coupled model called MythVariant while the MythVisu atomic model coding is given below:

```
1class MythVisu(MythDomainBehavior) :
2   def intTransition(self):
3       self.state['sigma'] = INFINITY
4   def extTransition(self):
5       mv=
self.OPorts[0].outLine[0].host
6       msg = self.peek(self.IPorts[0])
7       mv.componentSet = []
8    m=PointMyth.PointMyth(latitude..)
9   m.timeNext=m.timeLast= 0.
10      m.addInPort()
11      m.addOutPort()
12      mv.addSubModel(m)
13  self.state['status']='ACTIF'
14   self.state['sigma'] = 0
15  def outputFnc(self):
16      self.msg.value = 0
17      self.msg.time= self.timeNext
18  self.poke(self.OPorts[0],self.msg)
19  def timeAdvance(self): return
20   self.state['sigma']
```

The dynamic modification of the initial empty coupled model MythVariant (pointed out on figure 4) is realized by the δ_{ext} transition function of the atomic model called MythVisu as it may be seen in the code presented above (see statement line 12 `mv.addSubModel(m))`. The variable `m` is an instance of the PointMyth atomic class model and is computed by the statement line 8.

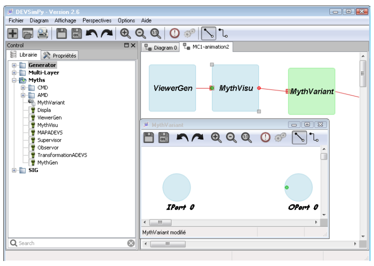


Figure 4: Illustration of the MythVariant coupled model.

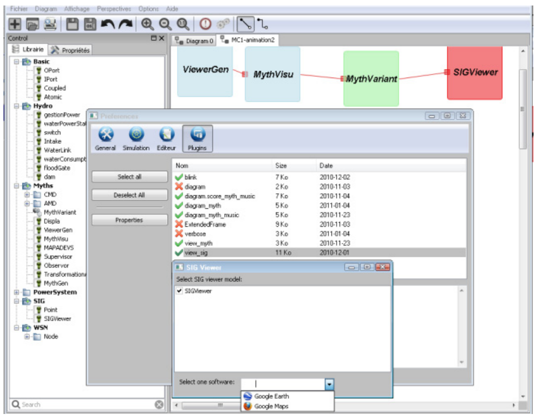
The class MythVisu described above points out four methods corresponding to the four basic functions of an atomic model. The DEVS $\boldsymbol{\delta_{int}}$, function of the atomic model MythVisu is coded through the intTransition method (described in lines 2 and 3) and consists in assigning the current value of the state variable sigma to infinity in order to indicate that the atomic model is waiting for an external event. The DEVS δ_{ext}, function of the atomic model MythVisu is coded through the extTransition method (cf. from line 4 to 14) which mainly consists in generating new atomic models in the associated coupled model called MythVariant. The DEVS λ function is coded through the outputFnc method of the atomic model MythVisu -from line 15 to 18) and consists in sending a message on the output port of the atomic model own to the statement of line 18. Finally the DEVS ta function is coded through the timeAdvance method of the atomic model MythVisu (lines 19 and 20) and consists in returning the current value of the state variable sigma.

3.4 Google Earth Invocation

The Google Earth invocation leans on the two following atomic models: PointMyth and SIGViewer. The link between the DEVSimPy software and Google Earth is performed through a KML (Keyhole Markup Language) file (Google Inc, 2007). KML enables to build and to organize points, lines and other information on a Google Earth map. The δ_{ext} function of the SIGviewer atomic model allows the management of the placement of marks on Google Earth using a KML file as it may be seen below:

```
1def extTransition(self):
2  activePort = self.myInput.keys()[0]
3  msg = self.peek(activePort)
4  point = msg.value[0]
5 kml_tree = KML.KML_Tree(self.fn)
6  ......
7  kml_tree.add_placemark(….)
8  kml_tree.write(self.fn)
9 self.state['sigma'] = 0
```

Furthermore the dynamic visualization will be performed using the plug-in *view_myth* of DEVSimPy.
This plug-in allows the user to associate the initialization of the software which will be



```
Nom                       Size    Date
blink                     7 Ko    2010-12-02
diagram                   2 Ko    2010-11-03
diagram.score_myth_music  7 Ko    2010-11-04
diagram_myth              5 Ko    2011-01-04
diagram_myth_music        5 Ko    2010-11-23
ExtendedFrame             9 Ko    2010-11-03
verbose                   3 Ko    2011-01-04
view_myth                 3 Ko    2010-11-23
view_sig                  11 Ko   2010-12-01

Select SIG viewer model:
✓ SIGViewer
Select one software:
```

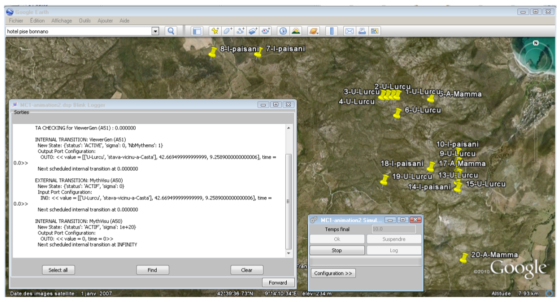
Figure 5: ViewMyth plug-in initialisation.

concerned with the dynamic visualization. The user has to choose between Google Earth and Google Map as it may be seen in figure 5.

3.5 Dynamic Visualization

The dynamic visualization of the placement of points on the map is realized by setting a refreshment of the KML file associated with the SIGviewer atomic model like described above. The file is modified every time an event is sent to the SIGviewer atomic model own to the δ_{ext} function presented in sub-section 3.4. The detection of an event is performed through the statement pointed out line 3 of the previously detailed code of δ_{ext}. Then the KML file is modified according to the code presented above in sub-section 3.4 (form line 5 to line 9). Furthermore the KML file can be checked by the Google Earth API periodically in order to have dynamic refreshment by enabling the auto-update function of the Google Earth API. Using this option the KML file source is regularly reloaded at an interval that we have specify. The modification of the KML file is performed during the DEVS simulation of the overall DEVS modeling presented in sub-section 3.2.

However one of the main interests of the proposed approach is to offer the possibility of a dynamic printing on a map the unfolding of a folktale. We therefore have to perform a step-to-step simulation in order to activate the refreshment of the KML file associated with the SIGViewer using the previously introduced method. We use a special plug-in called Blink belonging to the DEVSimPy features allowing to perform a step-to-step simulation as described in figure 6.



```
        TA CHECKING for ViewerGen (A51) : 0.000000

        INTERNAL TRANSITION: ViewerGen (A51)
         New State: {'status': 'ACTIVE', 'sigma': 0, 'NbMythems': 1}
         Output Port Configuration:
          OUT0: << value = [['U-Lurcu', 'stava-vicinu-a-Casta'], 42.669499999999999, 9.2589000000000006], time =
0.0>>
         Next scheduled internal transition at 0.000000

        EXTERNAL TRANSITION: MythVisu (A50)
         New State: {'status': 'ACTIF', 'sigma': 0}
         Input Port Configuration:
          IN0: << value = [['U-Lurcu', 'stava-vicinu-a-Casta'], 42.669499999999999, 9.2589000000000006], time =
0.0>>
         Next scheduled internal transition at 0.000000

        INTERNAL TRANSITION: MythVisu (A50)
         New State: {'status': 'ACTIF', 'sigma': 1e+20}
         Output Port Configuration:
          OUT0: << value = 0, time = 0>>
         Next scheduled internal transition at INFINITY
```

Figure 6: Step-to-step simulation.

Using the Forward button of the Blink Logger window we are able to control the generation of the KML file. By setting an appropriate period of refreshment (for example 10 second) the user is able to see the placement of every mythem on the screen and read the required information concerning the folktale. There are therefore two ways of using the simulation: (i) by selecting the step-to-step simulation -the user has in this case chosen the Blink plug-in as seen in sub-section 3.5 and is able to discover the unfolding of the myth interactively by clicking on the forward button; (ii) by performing a complete simulation and own to the enumeration of the generated points on the map the user is able to read the unfolding of the myth on the map by clicking on the different points of the story one after the other.

4 CONCLUSIONS

We have presented in this paper set an approach to dynamically visualize the unfolding of a folktale. This work is part of a multi-disciplinary research concerning Structural anthropology of myths and computer science: the goal is to offer a software approach to help an anthropologist to perform Claude Levi Strauss myth analysis method. This method is based on the concept of mythem. Each folktale (or myth) is decomposed into a set of mythems which are the basic elements of a given story. The proposed approach leans on an association of the DEVS formalism and the Google Earth API. We have detailed the set of DEVS models which has been defined in order to: (i) dynamically read a text file containing the story where each line of the file corresponds to a mythem ; (ii) dynamically print on a map each mythem using the Google Earth API. We introduced a software framework called DEVSimPy which has been used in order to implement the set of DEVS models required in order to perform the dynamic visualization of myths. We also described the validation of these DEVS model using a concrete folktale belonging to the Corsican mythology. The future work will consist in deeply collaborating with an anthropologist in order to model the 813 myths defined by Claude Levi Strauss in his *Mythologiques Series* and a set of more than 200 myths coming from the Corsican Mythology and known by the anthropologist. Our future work will also concern the dynamic simulation of a given myth in a 3D environment.

REFERENCES

Baâti, L., Frydman, C. & Giambiasi, N. (2007), LSIS-DME M&S Environment Extended by Dynamic Hierarchical Structure DEVS Modeling Approach. *ACM/SMS proceedings of the Spring Simulation Conference*, Norfolk, Virginia, USA.

Barros, F. J. (1997). Modeling Formalism for Dynamic Structure Systems. *ACM Transactions on Modeling and Computer Simulation*, 7(4), 501-514.

Bolduc, J. S. & Vangheluwe, H, (2001). *The modeling and simulation package PythonDEVS for classical hierarchical DEVS*. Technical report MDSL-TR-2001-01, McGill University, Montréal, Canada

Brown, M.C. (2006*), Hacking Google Maps and Google Earth*, Canada: Wiley Publishing, Inc

Google Inc., KML 2.1 Reference, 2007, http://earth.google.com/kmlkml_tags_21.html.

Hu, X., Zeigler B. P. & S. Mittal, (2005). Variable Structure in DEVS Component-Based Modeling and Simulation, *Simulation,* 81(2), 91-102

Hui, S. & Wainer, G. (2006). A Simulation Algorithm for Dynamic Structure DEVS Modeling. *Proceedings of the Winter Simulation* Conference, WSC 06, 815 – 822.

Jason, H. & Segal, D. (1977). *Patterns in Oral Literature*. The Hague: Mouton Publishers

Lévi-Strauss, C. (1955). The structural study of myths. *Journal of American Folklore, 68(270)*, 428-444.

Levi-Strauss, C. (1969). *Introduction to a Science of Mythology 1. The Raw and the Cooked,* New York: Harper & Row

Levi-Strauss, C., (1973). *Introduction to a Science of Mythology 2. From Honey to Ashes,* New York: Harper & Row.

Levi-Strauss, C. (1978). *Introduction to a Science of Mythology 3. The Origin of Table Manners,* New York: Harper & Row.

Levi-Strauss, C. (1981). *Introduction to a Science of Mythology 4. The Naked Man.* New York: Harper & Row

Richard, P. & Jaulin, R. (1971). *Anthropologie et calcul,* Paris: Union Générale d'édition.

M. F Sanner (1999) , "Python: a programming language for software integration and development," *J. Mol. Graphics Mod* 17, pp. 57–61.

Santucci J. F., De Gentili E. (2009). Dynamic variable structure modeling and simulation of the Claude Levi-Strauss's mythical thought morphodynamics *Proceedings de la ACM SPRINGSIM'09 Conference, 22-27 Mars 2009, San Diego, USA.*

Santucci J. F. , De Gentili E., Thury-Bouvet G. (2010) Discrete Event Modeling and Simulation of the mythical thought morphodynamics involved in Claude Levi Strauss structural analysis, *Chapter 8 in Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models , Information Science Publishing, E.G. Blanchard & D. Allard (Editors),* pp. 152-178.

Zeigler, B. P. (1990). *Object-Oriented Simulation with Hierarchical, Modular Models*, London: Academic Press.

Zeigler, B. P., Praehofer, H. & Kim, T.G. (2000). *Theory of Modeling and Simulation. Second edition*, London: Academic Press.