BRAIN. Broad Research in Artificial Intelligence and Neuroscience,
ISSN 2067-3957, Volume 1, October 2010,
Special Issue on Advances in Applied Sciences,
Eds Barna Iantovics, Marius Măruşteri, Rodica-M. Ion, Roumen Kountchev

THE STATE CHART METAPHOR FOR THE DESIGN OF COLLABORATIVE VIRTUAL ENVIRONMENTS

Ugo Barchetti, Alberto Bucciero, Anna Lisa Guido,
Luca Mainetti, Roberto Paiano, Andrea Pandurino

Abstract. Nowadays there are an increasing evolution of technological solutions providing the virtual collaboration among people geographically distributed to obtain some common goals. At the same time virtual collaborative environments are often used for learning purposes because increase the interaction among users. These systems are usually developed in a custom way according to the specific requirements without any kind of standardization in terms of definition of collaboration among involved users. Although the literature shows attempts to formally express learning processes through a standard language, there are not proposals that combine e-learning languages with typical concepts of 3D collaborative experiences. This paper propose a notation model able to design collaboration for e-learning virtual environments. In this paper we present also the architecture of a tool that helps the designer in the use of the notation model here proposed.

Keywords: *CVE, Metaphor, e-learning*

2000 *Mathematics Subject Classification*: 68U05, 68T35.

1. Introduction

The technology evolution brought the development of different on-line learning methods integrated into Learning Management Systems (LMS) providing tools for learning process management useful to teachers, tutors and students.

Although the e-learning systems get better over time, we think there are some aspects to develop in order to provide more suggestive and realistic experiences. From a educational point of view, the platforms must be not only quite simple tools, but a learning environment able to support educational and pedagogic contexts. Then the e-learning system must be considered as an environment through which the user builds his learning with a flow of interaction, meeting and processes established between he and the context. A possible way to make realistic the experience is to introduce the 3D virtual environment in which people will virtually meet together and will learn something about a specific topic. It is not very simple to combine the e-learning language and the concepts of 3D collaborative experience, so it is important to introduce a design process that brings in a structured way to the design and the following development of the learning experience using 3D virtual environment. The real problem is to describe the learning process in a way as close as possible to the way of thinking of the people that will benefit of it such as pedagogic people The research community tried to codify the learning process and to formally describe it through a standard language. Was born, so, Educational Modelling Language (EML)[1] term with the goal to create formal model of learning processes. EML is a semantic notation model oriented to contents and processes description, from a pedagogic point of view, in order to support reusability and interoperability among didactic objects, activities and so on. This notation can offer strategies to guarantee the didactic materials reusability through Learning Objects, but is not enough to describe the typical interactions of CVEs (Collaborative Virtual Environment). Starting from the acquired experiences in the CVE field, we think that it is important to develop an abstract language that takes into account the our know-how of previous experiences in e-learning and in the collaborative virtual environment. To obtain this goal, we are following two distinct approaches. The first oriented to users without any particular technical expertise. The second more oriented to users with technical or pedagogic skills. The first approach deals with a metaphor theatre and try to define the overall 3D e-learning experience using concepts related to the theatre. The second one, the goal of this paper, follows the e-

learning oriented literature that shows examples of CVE modelling languages as Petri Nets and the like. The approach allows to define the 3D e-learning experience using the statechart metaphor. Another goal of this paper is to present the overall architecture of a design tool that we design and develop. The tool helps the designer in the application of the approach here proposed for the 3D e-learning experience. We also show some screenshot of this tool.

2. State of the art

To understand our approach in the design of e-learning experience made up using a collaborative virtual environment as platform, it is important to understand the scientific panorama approach. Collaborative Virtual Real Environment (CVRE) [2] was developed by the Universidad du Chile and it design the virtual environment using the concepts of the real world: room, auditorium and so on.

The model allows to define people, space, tools and rules: people interact in the virtual place using available tools (chat, blog,..). The modality of user interaction with the objects is controlled using rules that define the role that a person may have in a specific place; for example a office manager will be responsible only in its office and he/she will be a generic user in other spaces. CVRE has a graphic notation that does not uses the metaphor and so it is not useful. Process Modeling Language (PML) [3] comes from UML and it is based on two levels: High Level Uml-Based Diagram (it uses P-activity to design the activities and P-class to design objects) and Low Level Process Language an Object-Oriented language to provide a representation of the classes defined in the P-class using methods described in the P-activity. Nimmit (Notation for Modeling Multimodal Interaction Techniques) [4] [5], is another approach to design the interaction between human and computer, it is based on a graphical notation oriented to the virtual environment. NiMMiT uses the concepts of state and events: a state answer to a set of events that activate a task or a set of tasks and allow to change the state in a new state. The state change applies only if the task end with success. Considering the semantic gap between designer and developers of CVE, a new techniques was thought, the InTml (Interaction Techniques Markup Language) [6] a domain specific language born to define several aspects of a virtual environment (device, objects, interactions). InTML uses a dataflow architecture where objects and device are part of the flow and are linked each other. The filter object allows representing the interaction and in its simple form is made up of a set of input and

output. The filters may be defined and customized by designer. Each filter may use input and other filter to send its output. As for NiMMiT, inTML allows to export in XML format the design to use in the engine. The graphical notation of inTML is more rigid and inflexible.

3. The state chart metaphor

The real problem in the design of virtual environment in the e-learning is not to define a notation or a formal language useful to describe the e-learning session and the 3D environment where the e-learning session will takes place, but to define a language useful for different types of user that will works together in order to obtain advantages from it and, at the same time, able to provide all the needed details. In a definition of a e-learning experience there are several type of users involved:

- Storyboard Artist: the task of this user is to define the goal to reach and how it is possible to reach it using the natural language or using some draft;
- Designer: he/she creates several environment and geometries that characterize the e-learning experience;
- Pedagogic people: he/she describes the interaction useful to obtain feedbacks from user involved in the learning experience;
- Technical users: they put in the virtual environment the materials and the input coming from designer and pedagogues.

There is an interoperability problem between these different types of user because each of them has its own language and its own goal in the definition of a collaborative virtual environment e-learning session. Starting from this consideration we think to a new language that uses a metaphor easy to understand by different types of user and that allows us to define, using two different analysis layer, all the details useful to generate an e-learning collaborative virtual environment. In order to select the metaphor we observe that:

- It is important to have a formal description of the different layer and of the different points-of-view of the static view of the virtual world.
- It is important to design in a separate way the dynamic part of the experience.

We think that the metaphor that best meets these two problems is the statechart metaphor. Using the state chartmetaphor it is possible to structure the language in two separate level: to define and to design a static view of the world, it is important to define a set of objects for example Rooms, Camera, light and so on. The use of a statechart metaphor is useful in the description of the dynamic view: the virtual environment is characterized from several events that follows each other: the most intuitive way to describe a transaction would be to imagine it as a transition of states. The proposed language is made up of more levels that allows, starting from the high level, to go down to a more detailed level that brings to the definition of all the detail of the collaborative experience.

4. High level view, low level view

The proposed language is based on two levels: one that allows to define, using the state machine, how the collaborative metaphor are organized (in a virtual environment a collaborative metaphor define how several objects, both avatars and objects, interact each other). Each collaborative metaphor is represented as a state of a state machine. It is possible to define the condition of input and output of each state and, in another detail level, it is possible to change the point of view describing how the single object participate to the metaphor. This representation, easy to understand from the user involved in the design, it is useful to provide a clear and complete idea of the collaborative metaphor of the e-learning environment to design (Figure 1).

The transition from one state to another is made up with input and output rules. When we define the behavior of the collaborative session, we must define the entity, the objects, the events and the action of the collaborative metaphors and how the collaboration happen. In the low level view, it is possible to describe the behavior of each state defined in the high level view providing both a conceptual and a physic meaning to the elements of the model.

In the table 1 we define some concept useful for the low level view. **User** is a stereotype that describe the user entity in the collaboration metaphor. **Object** is a stereotype to represent the 3D object in the collaboration metaphor. **Light** is a stereotype used to represent the different types of light to use. **Camera** is a stereotype to define the type of camera to use in the visualization. **Event** it defines the event related to an object. **Action** is a stereotype used to represent an action related to an event. **Connector** is a relation used to interconnect different event of the diagram.

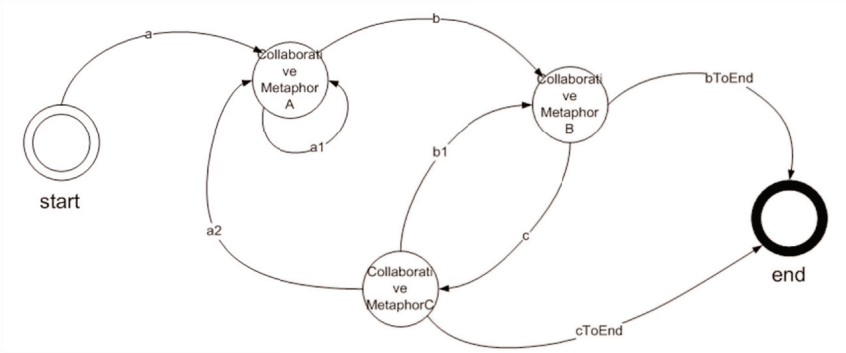

Figure 1: High Level View Example

Table 1: Low level view main concepts

Stencil	Description
	User
	Object
	Light
	Camera
	Event
	Action
	Connector

5. The design tool

The use of a metaphor to design 3D Virtual Learning Environment is an efficient way to approach the problem of the CVE design but it is important to have a support tool to apply the metaphor. The importance of the use of a design tool is twofold:

1. The tool suggests to the designer the primitives to use according to the context. For example if the designer is drawing the high level diagram, he/she can uses only the state and connection primitives and he/she does not use the others primitives. When the designer goes down to the low level view, he/she will be connected to the right high level view and he/she will be driven in the use of the primitive useful for its context.

2. The design of a 3D Virtual Learning Environment is an useful way to formalize the requirement of the user and to have a common semantic in order to allow both designer and developer to understand the same language. But, the importance of a tool is in its feature to translate in a machine readable format the design choices. In this way it is possible, using an appropriate engine for the generation of 3D collaborative experience, to automatic configure a 3D collaborative virtual experience. This is an important aspect that we take in consideration when we design the tool.

Taking in consideration the two important aspects considered above, we define the architecture of the design tool (Figure. 2). We observe that there is a database where we store the instances of each diagram and the details of the different users that, according to the metaphor, can participate in the definition of a specific model (Storyboard Artist, Designer, Pedagogic people, Technical users). The engine of the design tool accesses to the database using a database access object layer (DAO). The DAO has the task to abstract and encapsulates all accesses to the data source; it manages the connection with the data source to obtain and store data. The output of the design will be an XML file: the xml file will be the input for a CVE engine that will automatically generate the collaboration virtual environment. It is important to note that the CVE engine may use a set of template useful for the designer. The editor must present to the user these templates in order to help him in its design task. The core of the design tool architecture is the Design tool engine where there

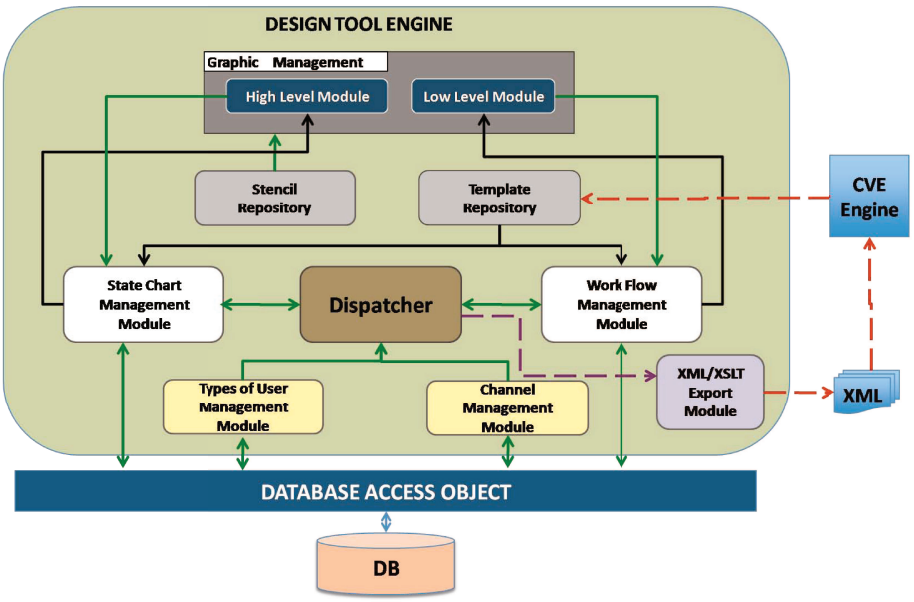

Figure 2: The design tool architecture

are all the modules. We can see in figure 2 that there are two colors for the arrows: green arrows and black arrows. Green arrows will be detailed later in this paper; black arrows indicate that the templates coming from the CVE engine, will be provided to the state chart management system module and to the workflow management system module: these modules present, using the graphic management module, the template to the designer. The use of template is very important: templates allow to re-use design aspect already used in others situations. The core of the design tool engine is the *Dispatcher module*. The dispatcher module links together (green arrows) the information coming from the Workflow Management Module and the information coming from the State Chart Management Module. In this way the engine composes the design of each diagram by linking together the high level view and the low level view. The language based on the state chart metaphor that we describe in the previous sections is oriented to several types of users: the dispatcher provides the right type of primitives to the specific user that is working (green arrows). The dispatcher, finally, has the task to adapt the design in order to make possible its use to the specific channel. The CVE, in fact, may be deployed to a different device with specific characteristics and the XML file,

output of the design phase, must consider the devices characteristics. The *state chart management module* and the *workflow management module* have the task to manage respectively, the high level view and the low level view. This means to present the right primitives for each view and to help the user in the use of the primitives. In these modules, there is the implementation of the interconnection rules between each element and the list of the properties of each element that will be used in the xml file. Both modules will interact with the db (green arrows) in order to store all the design choices of the user. The two modules *Type of User Management Module* and *Channel Management Module* has the task to collect the information about users and channels configuration coming from the database and to provide these information to the Dispatcher. The business logic of the dispatcher will present, according to the language, the primitives to the user. The two modules interact with the database. The *graphic management module* has the task to manage all graphic aspects that is the possibility to move objects inside the work area, the possibility to drag and drop elements, to link together two or more elements and so on. The stencil repository will contain all the graphical primitives defined in the previous section and all the graphic methods useful to provide the possibility to draw in the work area. When the overall design has been completed by all types of users involved, the *XML/XSLT module* will convert in a machine readable format the design and obtain as output the XML file. If the final CVE will be deployed on the different device, the XML/XSLT module will provide different XML files. The tool, that we design and develop, has been written in Java and we use the RCP platform to manage the graphical aspects of the tool. As a development tool we use Eclipse RCP that is a pre-configured environment for the use of the RCP platform.

6. The design tool use

In this section we will show a possible use of the language through the use of the editor. We will show how to apply the state chart metaphor in order to design a specific e-learning collaborative virtual experience. We suppose to design a learning session where people, represented through their avatar, are involved in two different games. Initially, the learning session is described by the Storyboard Artist that plans the activities. In the specific case the storyboard artist plans a meeting among involved avatar (welcome), a discussion about a specific topic (discussion) and then two games: Treasure Hunt and

Quiz. At the end of the games there is a wrap-up of what people learned during the session. The Storyboard Artist uses the statechart metaphor through the editor (Figure 3) and designs the high level view.

In the figure 3 it is possible to see also the different areas of the editor:

1. The *work area*, central part of the tool. In the figure there is the high level view area but, if the designer work in the low level view, there will be a tab bar. In each tab the user will define the metaphor according to the state chart that he/she has been previous defined;

2. The *Object area*: on the left it is possible to see all the Objects that the user may use in the design. If the designer work in the low level view, the Object area will display elements represented in tab.1;

3. The *Solution area*: on the right the designer has the list of the state in the state chart metaphor; a click to each file will open the low level view tab;

4. The *menu area*: on the top it contains the access point to a typical editor operation (save, undo, redo, etc.).

At this point the designer creates the environment and geometries that characterize the e-learning experience: the editor that we present does not provide these features because any designer can use the proper software (open source or commercial) useful for the definition of the geometries and environment. The Pedagogic people, define the low level view of each state of the state chart metaphor and describe the interaction useful for the e-learning session. If we consider the Treasure Hunt state, the pedagogic people will make the design of the interaction as showed in the figure 4. In the example, the Treasure Hunt is related to the learning session topic. We consider the Henry VIII as topic of the learning session then the Treasure Hunt can be defined using the objects related to that period. The avatars aim to search these objects .

The low level view of the Treasure Hunt metaphor is represented in the figure 4. In the figure 4 it is possible to see how each avatar must click to the object and answer to a question that the system asks. In each session there are also two avatar named Guide and Helper. These avatars do not have interaction with other avatars because they will give support to other avatars involved in the session.

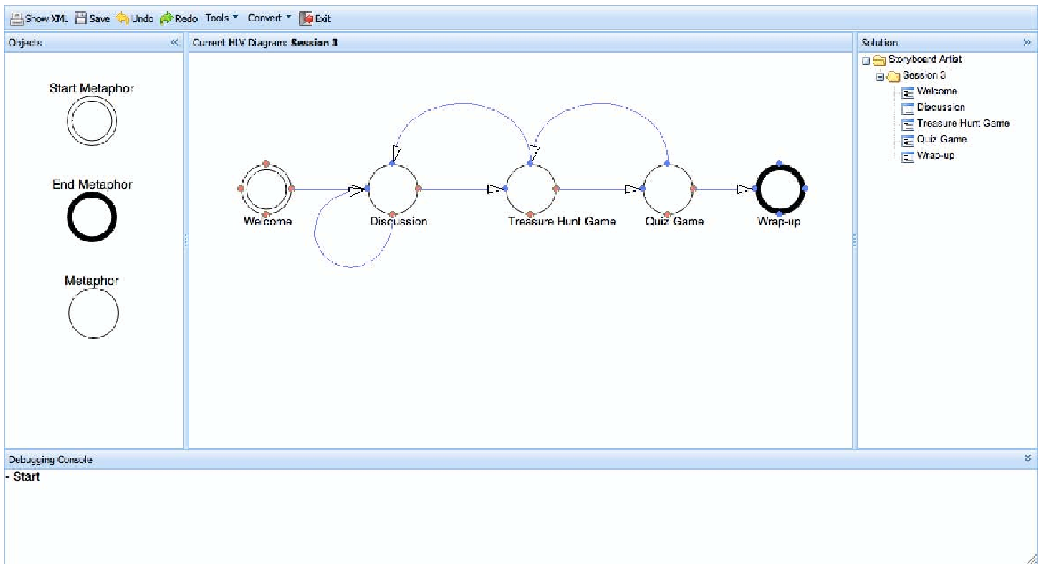

Figure 3: High level view designed by the storyboard artist

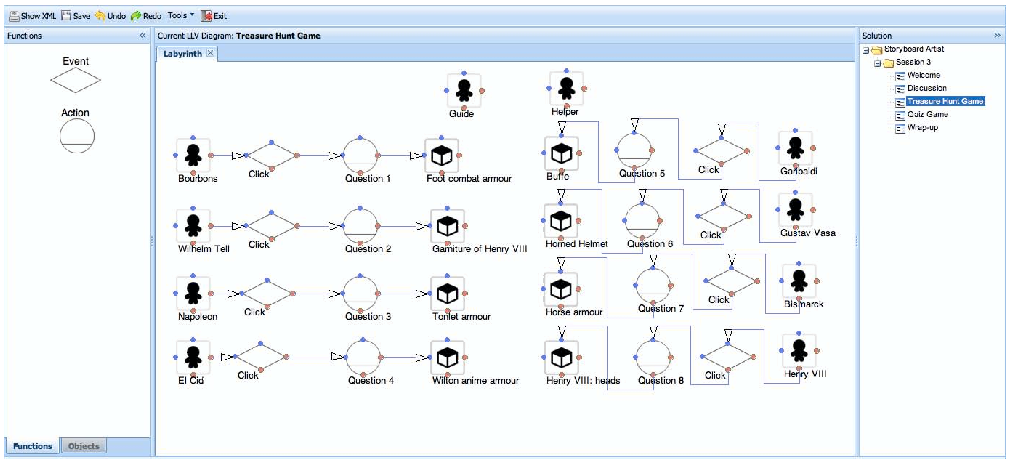

Figure 4: Low Level view designed by pedagogic user

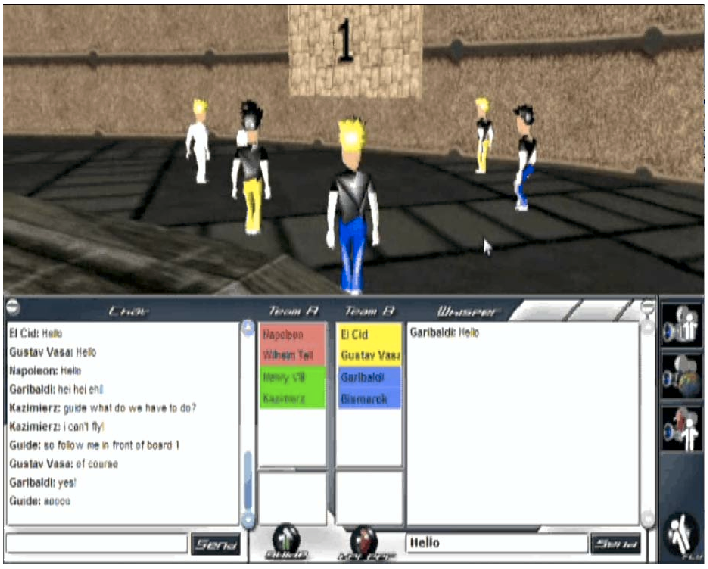

Figure 5: Screen shot of the generated collaborative virtual environment

The output of the design is an xml file. The technical user has to customize the designed session in order to put in the virtual environment engine in order to generate the session. The customization regards technical parameters (for example spatial coordinate, intensity of the light and so on). A screen shot of the generated session is in figure 5.

7. Conclusions

The importance of the CVE in the e-learning brings to an important consideration: it is necessary to have an approach to the design of the CVE oriented both to describe all the characteristics of the CVE and to cover the e-learning aspects. It is important, therefore, to define a language useful to define a CVE in a simple way. We have chosen in our research work the state chart metaphor useful to represent how the collaborative metaphor are related together and, in a detailed view, the definition of the specific characteristic of each collaborative metaphor. On a first experimentation of the state chart metaphor usage seems that different users (pedagogist, designer, storyboard artists and technical) may use without problems the metaphor and can describe the overall learning session. Our next step is to develop an editor in order to simplify the usage of the proposed language and to export the design in a machine readable

format. This will be the input for an engine that generates the collaborative e-learning sessions. We present in this paper also an editor that helps the designer in his/her task and provides useful features to export the overall design in one or many xml files (multiple xml files will be obtained when the CVE will be deployed to many devices). This will be the input for an engine that generates the collaborative e-learning sessions. We have already developed a prototype of this editor.

8. Acknowledgments

We would like to thank Professor Paolo Paolini for his tangible support in the L4A, a national research project founded by MIUR (Ministero dellIstruzione dellUniversit e della Ricerca).

References

[1] Rawlings A., Van Rosmalen P., Koper R., Rodiguez-Artacho M., Lefrere P. Survey of Educational Modelling Languages (EMLs), CEN/ISSS WS Learning Technologies Workshop, 2002.

[2] Guerrero, L.A. Collazos, C.A. Pino, J.A. Ochoa, S.F. Aguilera, F. , Designing collaborative virtual environments based on real spaces to promote community interaction. In Web Congress, 2003. Proceedings. First Latin American ISBN: 0-7695-2058-8.

[3] Rossi,D., Turrini, E. Using a process modeling language for the design and implementation of process-driven applications. In Proceedings of International Conference on Software Engineering Advances (ICSEA 2007), (Cap Esterel, French Riviera, France, August 25-31, 2007).

[4] Vanacken D., De Boeck , J., Raymaekers, C., and K. Coninx NiMMiT: A notation for modeling multimodal interaction techniques. In Proceedings of the International Conference GRAPP06, (Setbal, Portugal, Febrary 25-28, 2006).

[5] Joan De, Vanacken, Davy, Raymaekers, Chris and Coninx, Karin High-Level Modeling of Multimodal Interaction Techniques Using NiMMiT. In Journal of Virtual Reality and Broadcasting, 4, 2 (Sept. 2007).

[6] Figueroa, P. Green, M. Hoover, H. J. InTml: A Description Language for VR Applications In Proceedings of Web3D'02, (Tempe, Arizona, USA, February 24-28, 2002).

Ugo Barchetti, Alberto Bucciero, Anna Lisa Guido,
Luca Mainetti, Roberto Paiano, Andrea Pandurino
Department of Engineering Innovation
University of Salento
Via per Monteroni, 73100 Lecce, Italy
email: {*ugo.barchetti , alberto.bucciero, annalisa.guido, luca.mainetti, roberto.paiano, andrea.pandurino*} *@unisalento.it*