1 Introduction
In the biomedical domain, there is a need to train people for the use and maintenance of diagnostic analytical devices. In the traditional training method, several students in a classroom with a live instructor manipulate the device alternately. Unfortunately in the biomedical industries, many new employees do not go through the classical training program. Moreover, these industries can not provide their devices for the employees training. We aim to use virtual reality and virtual environments in order to provide more freedom to users during training programs. For instance, using virtual reality, learners can train when they want, and where they want without any constraint.

Virtual environments can be combined with Intelligent Tutoring System (ITS) in order to adapt the learning situation depending on the learner activities [5]. In the literature there are plenty of ITS models [17, 11, 15], however they seems incomplete. One of the main lacks is the genericity, which means that we need to modify the ITS as soon as we change the environment [5]. For example, modifications of the ITS will be needed everytime we change the device or the exercise. Another lack is the possibility for the teacher to build the training by adding the concepts of objectives and prerequisites. These concepts can be provided by a pedagogical scenario.

The objective of this paper is to propose the most complete ITS called Chrysaor.
2.1 Major characteristics

As mentioned above, genericity is a major characteristic needed for our ITS. However, the actual methodology of ITS engineering is not ideal: each application is developed independently, tutoring expertise is hard-coded into individual applications, and there is a little reuse of tutoring components [13]. It is essential to set our ITS as much generic as possible.

Some elements, such as the significance of the pedagogical strategy, which can substantially vary according to the way the tutor want to intervene, when and how intervene [19], seem quite interesting. Based on this, we may want to add a disruptive [2] or a companion-component [11]. In the same way, some ITS behaviors could be replaced by a human contribution. So, we need some kind of modularity of our ITS behaviors.

In order to have a better efficiency, we want suitable assistances for each learner. So we have to work on the individualization of the ITS. The classical ITS are composed of a learner model which represents the learner’s knowledge at the problem level [18]. Our ITS needs to have a learner model in order to individualize the assistances.

However proposed assistances may not be ideal from the instructor point of view. Therefore we wish that our ITS can modify his behavior depending on what happens in the simulation: the ITS becomes adaptive [5].

Another of our goal is to enable the teacher to customize each exercise. He might have the possibility to write a pedagogical scenario. One of the most significant proposal of the community is the IMS Learning Design (or IMS-LD) standard [8]. In ISM-LD a scenario is considered as a sequence of pedagogical activities. Lately, the pedagogical scenario model called Poseidon [10] was proposed. It enables a description of all of the components in pedagogical scenarios including activities in virtual environment. For these reasons, several characteristics seems important in the conception of an ITS: individualization, modularity, genericity, adaptativity, scenarios.

2.2 Existing ITS

In the ITS domain, many research works were made during the past few years and most of them focus on two or three characteristics described in section 2.1 (Table 1.).

Pegase is the most complete ITS with all these characteristics, so we have decided to based our own ITS on it. Nevertheless Pegase lacks the pedagogical scenario characteristic and some other can be improved, like the modularity. In the next section, we expose the improvement made in Chrysaor model based on the modularity and the pedagogical scenario.

3 Model

The ITS Pegase is the most complete ITS according to the five characteristics detailed in the previous section, we have decided to based our own ITS on it. Nevertheless Pegase lacks the pedagogical scenario characteristic and some other features can be improve like modularity. To do
so, we choose to rely our ITS on Mascaret [12] like Pegase, in order to use the multi-agent system technology for the modularity. Moreover Mascaret enables to easily reify the expert or the pedagogical scenario knowledge so that the tutor could reason on it. Mascaret is a metamodel to describe virtual environments and the agents evolving in this environment. This metamodel (founded on the Unified Modeling Language : UML) provide an unified modeling language to describe the structure of the environment (entities, positions...), entitie’s and agent’s behavior.

3.1 Modularity

Mascaret is based on many agent concepts like agent communication, behavior and knowledge bases (Fig. 1). In order to improve the modularity of our ITS, we have used the multi-agent system technology so that we could work on these concepts.

![UML model of the knowledge base of Mascaret](image)

Figure 1: UML model of the knowledge base of Mascaret

We started to transform Pegase by providing it the multi-agent system technology. In the previous model, Pegase was represented by one unique agent. The pedagogical behavior of this agent was structured around several step (detect an error, propose an assistance, etc.) and set by several data (Learner model, Pedagogical model, etc.). Using Mascaret and the agent architecture, Chrysaor keeps this pedagogical behavior but it is divided in several agents (Error Agent, Pedagogical Agent) and several behaviors (classify the error, etc.). With this technology we are able to modify one behavior independently. Moreover, if a specific behavior is not efficient enough, it can be played by a human teacher.

Afterwards, we based our work on the agent communication, by adding the communication protocol FIPA-ACL (Agent Communication Language) to our system. Due to this standard communication protocol, we can now easily add or remove an agent or communicate with a new application on the network.

Similarly, we have tried to endow modularity to the agents' behaviors. Thus, we have chosen to describe the multi-agent system with an activity diagram because we based our model on Mascaret which enables to describe the behaviors with it. We defined the procedures and the roles within the organization. Thanks to the multi-agent system, we can easily change the organisation of the ITS agents: for example, we may want to add a disruptive or a companion-component. In the same way, some ITS behaviors could be replaced by a human contribution. In order to obtain this freedom to arrange the pedagogical strategies by adding or replacing some behaviors, we have increased the modularity of the ITS.

Our studies also focus on the agents knowledge base of our ITS. Indeed, the FIPA standard proposes to provide a knowledge base to the agent, but without giving a formalism for the knowledge base. One interest of the knowledge bases is to enable the agents to reason on the data contained in it. Agent Knowledge base is a crucial point of our ITS model and thankfully many researches are based on it. In Mascaret, the agents have a knowledge base. In our actual model, the knowledge could be expressed in two models: the expert model (entity, behavior, procedure) and the pedagogical model (agent, behavior, procedure). These two knowledge bases are described on
the same language provided by Mascaret which is founded on UML. Consequently, some models expressed in UML could be used as a knowledge base of Chrysaor agents. A pedagogical scenario can be described through a procedure, therefore we could use the pedagogical scenario as a knowledge base (Fig. 1). This improvement enables us to easily change the knowledge of each agent. In this context we wanted to have the pedagogical scenario to be a knowledge for the agents.

3.2 Pedagogical Scenario

As seen in section 2, Chrysaor lacks of pedagogical scenario knowledge. Previously, we said that one of the most significant proposals of the community is the IMS-LD standard which focuses on the organization of learning activities. We wanted to use the IMS-LD standard so we have chosen to couple Poseidon (a pedagogical scenario model) with Chrysaor. Poseidon is an implementation of IMS-LD for virtual reality.

First, we aimed to rebuild the structure of Poseidon in order to perform it with Mascaret, so it will be considered as a knowledge base for the agents. The main class of Poseidon is LearningSession (a pedagogical scenario) and includes (Fig. 2): a prerequisite list (Prerequisite), an objective list (LearningObjective), an environment (EducationalEnvironment), the activities of the scenario (EducationalScenario), and the pedagogical resources (Pedagogical Resource) linked to entities (LearningObject).

![Figure 2: UML class diagram of the package Poseidon](image)

The EducationalScenario inherits from a Mascaret Activity and the EducationalEnvironment is an instance of a Mascaret Environment, thus we can now use the pedagogical data as an agent knowledge, so that the agents could reason on it. For example, our pedagogical scenario could be composed of multiple exercises. In one of them, there are some observables like a TimeEvent (from UML:Activity) at a critical point of the procedure. Chrysaor can reason on this scenario and choose to apply or not the observables.

4 Application

The use of Chrysaor is illustrated by an application of a virtual biomedical analyzer which enables to learn the technical procedures of the device. Many biomedical analyzers can be found in hospitals and analytical laboratories. There is a need to train people for the use and maintenance of these analysers due to the employee turnover. We decided to base our virtual reality application on a real analyzer (Fig. 3).

4.1 Virtual analyzer

We based our application called VirtualAnalyzer on the real analyzer and we have implemented a routine procedure called basicUseProcedure which is composed of 120 basic actions and an execution time of approximately 40 minutes for a beginner. VirtualAnalyzer (Fig.
4) is a virtual reality application for the use of the biomedical analyzer, in which the learner have to do some reagents reconstitutions and use them in the analyzer in order to start a biomedical test. There are plenty of possible actions and only one role for this procedure. In reality, users can directly interact with the analyzer (open a door, insert a reagent, etc.) or interact with the associate computer (launch test, open the drawer, etc.) We choose to simulate this computer interface with Android and execute it under a touchpad which communicate with the virtual analyzer. In the event of the learner making an error, an assistance will be proposed by the ITS, and the human teacher will choose one of them, like increase transparency of all the environment elements apart from the correct object (Fig. 5).

With the use of our ITS in a virtual environment, learners can train at the execution of a procedure, and repeat it as much as they want. Like for the ITS model, this application uses the generic models, i.e. the structure of the environment, objects, organizations and procedures present in the application are described by a Mascaret model (close to UML model).

4.1.1  Environment

Previously, we exposed that an environment could be an agent knowledge. Thus, all the data of the environment could be a knowledge for the agents : the entities (e.g. a reagent and its volume) and the organization (e.g. the basicUseProcedure). Moreover, we have described some procedures using an activity diagram (Fig. 6). All these information can be used as an agent knowledge.

4.1.2  Pedagogical Scenario

Using the new Poseidon, we have described a pedagogical scenario which can be split in multiple pedagogical exercises. We can see an example of exercise in Figure 7: an activity diagram with all the actions for the learner, and some observable (e.g. a TimeEvent) for the teacher coupled with a pedagogical action.
4.2 ITS

Our pedagogical scenario is composed of two exercises. In the first one, the tutor presents all the important parts of the analyzer, and the learner has nothing to do. In the second part, the learner has to do a routine procedure. There are some observables like a TimeEvent or an imposed assistance at some critical points of the procedure. In our example, we choose to have the tutor play the teacher role of the pedagogical scenario. As though we have explained in section 3.2, the pedagogical scenario could be a knowledge of the tutor. For the first exercise, the tutor reason and choose the best way to present the parts of the analyzer (for example, grow up a part). For the second exercise, the learner does an action and the tutor has all the knowledge about the correct action (expert model) or the observables (pedagogical scenario). For example, the learner does a wrong action, so the tutor reason and propose an assistance like increase transparency of all the environment elements apart from the correct object. Later, if the learner takes too long to do the correct action the tutor could choose to apply or not the observable define in the pedagogical scenario.

4.3 Experimentation

With the help of some cognitive psychologists, we would like to evaluate the transfer of the learning skills acquired in a virtual environment to the real life. To confirm the utility and the efficiency of VirtualAnalyzer and Chrysaor, we set up the experimentation based on a cognitive psychology literature scientific analysis [3]. This experimental working is designed and holded by a cognitive psychologist. This research have a double goal: to check how the learning of a procedure takes places within a virtual environment for training (VirtualAnalyzer), and to estimate the tutor (Chrysaor) contribution for the learning. Behavioural measurements (time to perform a task, number of instructions consulting, number of incorrect actions) will be collected as thirty people will carry out a procedure. The thirty people will be divided into two independents groups: one group will performs the procedure on virtual environment and one group will perform the same procedure on virtual environment increased by the tutor.

5 Conclusion and futurs works

In this paper, we presented all the important characteristics needed for an ITS. On this basis, our first studies focused on the modular part of our ITS. In order to bring off these studies, we based on Mascaret metamodels. First we use the multi-agent system technology on Pegase. Now we can easily add a new agent, or remove one in order to play its role by a human. In the same way, we can easily change the behaviors of each role. Then we have modified Mascaret and use
the environments as some knowledge of the agents. The genericity and the modularity of Pegase have been increased. At least, we have incorporate Poseidon in our model in order to use it as some knowledge for our ITS. The application of the virtual environment figuring a biomedical analyzer and containing a procedure enables us to test Chrysaor and the use of the new Poseidon.

The incoming experimentation is very important because we can improve our model according to the experimental results.

Our future studies will focus on the learner model, in order to individualize more and more the simulation for the learner. For example we started to work on including some data from external questionnaires (like Hollnagel [7]) in our learner model. With these works, we will be able to have a more specific pedagogy for each learner.

6 References