AN INTELLIGENT AND PRACTICAL EDUCATIONAL ENVIRONMENT

Augustin Prodan

ABSTRACT
This paper presents a Java framework for designing and implementing intelligent and practical e-learning tools, to be used by both the students and the teaching staff in their didactic and research activities, especially in a context of open learning. As the electronic technologies do not contain teaching methods and pedagogical characteristics, we have to simulate a high quality teaching-learning relation inside the e-course. Such a simulation must incorporate both the integrated knowledge of the best teachers and the best pedagogical strategies, obtaining this way an intelligent e-learning tool. At the same time, we prepare our e-courses based on practice and real world experiences, as the practice is essential in learning activities. An e-course consists of a set of e-learning scenarios, each e-learning scenario being generated by virtue of some well-defined learning objectives. To generate e-learning scenarios for a particular e-course, we have to create first a particular infrastructure containing knowledge, strategies and experiences from particular domains of the target e-course. We use Java technologies to create intelligent and practical e-learning scenarios based on new AI (Artificial Intelligence) paradigms, such as Case Based Reasoning, Bayesian Inference and Intelligent Agents. We approach the first e-course in domains of Statistics, Stochastic Modeling and Simulation.

KEYWORDS
Case-based Reasoning, e-learning scenario, learning process, open learning, teaching-learning relation, XML-based knowledge base

INTRODUCTION

This paper outlines the results of a research achieved by Department of Mathematics and Informatics at Iuliu Hatieganu University, with the purpose to build intelligent and practical e-learning tools. Our final aim is to create an electronic educational environment consisting of intelligent and practical e-learning tools to be used by both the students and the teaching staff in their didactic and research activities. We emphasize that the two characteristics, "intelligent" and "practical", are very important for our e-learning tools, especially in a context of open learning.

Generally, educational activities consist of teaching and learning processes. Teachers disseminate knowledge towards learners through teaching processes, while learners acquire knowledge through learning processes. The purpose of teaching methods is to facilitate the settlement of teaching-learning relations between the dual teaching and learning processes. A teaching process is really working if and only if the associated learning process is working. Therefore, we are successful in teaching if and only if our partners (students, colleagues, etc.) obtain successes in learning (Prodan, 1996). The quality of a teaching-learning relation is highly depending on pedagogical characteristics and on pedagogical context. These assertions are true in all educational contexts, from traditional face-to-face courses to new electronic educational environments offered by e-teaching and e-learning technologies. In a face-to-face course, pedagogical virtue is mainly radiated by physical presence of a good teacher. While pedagogical virtue of the teacher is partly maintained in an online e-course, the most of it is lost in traditional courses distributed to the students by means of WWW (World Wide Web) for open learning

(Prodan, 1998). However, we think that open learning is the most important component of the real learning, because the real learning is achieved for the most part through open learning. In an open learning context, learners have more control, they have the freedom to choose where, when and how to learn, each learner having his own pace. The benefits are for both slower and faster learners. The foreign students learning in a second language have extra time they need to understand the meaning of words by using dictionaries (Race, 1989). This is the reason why we consider that the e-courses distributed by WWW for open learning must have pedagogical characteristics and must be placed into a pedagogical context. Even if WWW with its associated two standards HTTP (HyperText Transfer Protocol) and HTML (HyperText Markup Language) is well suited to create electronic educational environments, a great deal of free documentation, guides and examples available over the Internet have little pedagogical characteristics, and consequently are of real help only to already educated specialists and to course designers. Internet provides a general communication space, which is the infrastructure for e-learning spaces. An e-learning space is created on the Internet around a subject, through e-courses. Who is the teacher and who are the learners in a learning space on the Internet? Any person who has access to Internet and knows how to use an e-course is a potential learner, if he is interested in that subject. The teacher could be identified as the person who creates the e-course. Who is able to create e-courses on the Internet? To create an e-course around a subject, there are necessary subject knowledge, pedagogical skills and adequate software infrastructure and skills. In most cases, an e-course is the result of collaboration between a subject expert and a software expert. The two experts may be the same person, especially when the subject belongs to Software Science. This is the reason why the most courses distributed on the Internet are with subjects in this domain. However, we must be careful when choosing subjects in Software Science, as the domains in software have a very high progress rate. The cost of an e-course is to be correlated with the life time of that course, and between the life time of a course and the progress rate in the field there is an obvious correlation.

In a context of open learning, it is very important to simulate a good teaching-learning relation inside the e-course, because the real teacher is absent during an open learning process, and the electronic infrastructure does not contain pedagogical characteristics. Such a simulation must incorporate both the integrated knowledge of the best teachers and the best pedagogical strategies, obtaining this way intelligent e-learning tools. We have to use CBR (Case-based Reasoning) (Leake, 1996), Java (Eckel 1998; Prodan and Prodan, 1997), JDBC (Java DataBase Connection) and XML (eXtensible Markup Language) technologies to elaborate intelligent and practical e-courses, with a view to create an intelligent e-learning environment on the Internet.

In addition, we prepare our e-courses based on practice and real world experiences. We think that practice is essential in learning activities, because learning by doing increases substantially the effectiveness of learning processes. We elaborate practical e-courses to be used in schools, universities, research institutes, enterprises and at home, allowing the learners to work on experiments with real world items. We approach the first e-course in domains of Statistics, Stochastic Modeling and Simulation, using as infrastructure a set of Java class libraries created and implemented by Department of Mathematics and Informatics from Iuliu Hatieganu University (Prodan and Prodan, 2002).

INTELLIGENT AND PRACTICAL E-COURSES

We intend to create intelligent and practical e-learning tools destined to be used especially in a context of open learning. As the real teacher is not physically present during the open learning processes, we must replace him with a virtual one, that means we have to simulate a high quality teaching-learning relation. A good traditional teacher has the role of a guide and we think that we are able to simulate this role into our intelligent e-learning tools. In fact, we have to incorporate the teaching activities into our intelligent e-learning software. We think that it is necessary to point out the obvious fact that the electronic technologies do not contain teaching methods and pedagogical characteristics. This being the plain truth, we have to transpose both previously mentioned elements in an electronic format and to infuse them into e-learning tools, obtaining this way intelligent e-courses. An e-course consists of a set of e-learning scenarios, each e-learning scenario being generated by virtue of some well-defined learning objectives. To generate intelligent and practical e-learning scenarios for a particular e-course, we have to create first a particular infrastructure containing the knowledge from particular domains of the target e-course. For this purpose, we have to generate consistent JDBC and XML based knowledge bases, containing integrated knowledge of the best teachers. In addition, we will implement in Java a set of simulation algorithms describing real world phenomena and processes we have to include in e-courses (see Figure 1). We use Java technologies to generate intelligent and practical e-learning scenarios, based on new AI (Artificial Intelligence) paradigms, such as CBR (Case-based Reasoning), Bayesian Inference and Intelligent Agents. A learner can access the e-course and launch an e-learning scenario either locally, or via WWW in a context of distance learning.

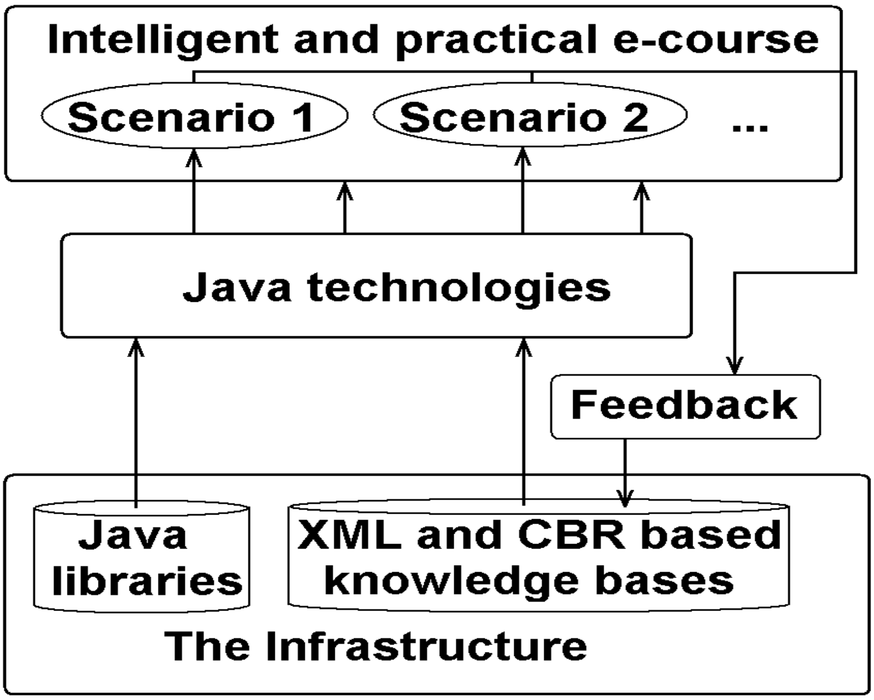

Figure 1. The Generation of the e-Learning Scenarios

A learning scenario is in fact like a traditional lesson, and the ideal solution is to simulate a teaching-learning relation with a virtual teacher able to interact with the learners and to instruct them. A good traditional teacher learns all the time from his previous didactic experiences. Based on this historical feedback, he always exploits prior specific successful episodes, and avoids prior failures. We introduce a similar feedback mechanism in our technology of elaborating e-courses. Following the traditional model, we store cases of positive experiences from previous e-learning scenarios in case bases created with JDBC, XML and CBR technologies (Leake, 1996). When generating a new e-learning scenario, we use a feedback mechanism based on our historical experience from previous e-learning scenarios (see Figure 1). The feedback information, collected from learners' remarks and from prior results and successes, is stored in case bases. The relevant cases are retrieved and adapted to fit new situations from new e-learning scenarios, or to improve the previous ones. In addition, our approach in creating an e-learning scenario relies upon a special sort of goal oriented intelligent agents. Although we did not identify exactly this type of agent when we consulted the much elaborated works (Wooldridge and Jenings, 1995; Nwana, 1996), we try to create and implement a goal oriented intelligent agent, able to incorporate knowledge, teaching methods and pedagogical characteristics into e-courses. We have to implement a simulation of some intelligence based actions and initiatives, that are to be incorporated into e-learning scenarios, with the purpose to map, to plan and to monitor the pace and the progress of a learning process.

When generating an e-learning scenario, we focus on pedagogical context and have all the time in mind the following pedagogical characteristics:

- **Friendly interface** – When the learners have the initiative to enter the e-course, a teaching-learning relation is initiated through the user interface. Hence, it follows that if our purpose is to facilitate the learners' access to knowledge, we have to create a friendly, easy to learn, efficient and agreeable graphical user interface.
- **General information** – When a learner enters for the first time into an e-course, we have to visualize some general information about that e-course, the objectives of the e-course, specific information about e-learning scenarios and how to use them.
- **Objectives** – Each e-learning scenario is generated by virtue of some well defined learning objectives. When a learner launches a particular e-learning scenario, we have to visualize the objectives corresponding to that scenario. By doing so, we help the learners to see what it is expected to be able to do after they will traverse the respective e-learning scenario.
- **Orientation facilities** – When the learners use e-learning scenarios to navigate in an ocean of information and knowledge, browsing through XML based knowledge bases and hyperdocuments, they want to know at any time their position. We implement in e-learning scenarios many techniques to facilitate the navigation, such as maps, marked routes, bookmarks, diagrams, queries, etc.
- **Layered structure** – An e-course has a layered structure from simple to complex, allowing each learner to access an optimum layer, depending on his purpose and previous knowledge.
- **Customization** – The functionality of an e-learning scenario is adapted to ability and purpose of each learner. Customization can be either static or dynamic. Static customization is an easy task, being carried out by a set of parameters before run time. Dynamic customization is more difficult, because it is necessary to collect information about learner during e-learning scenario execution.
- **Global and local coherency** – To improve global coherency of an e-learning scenario, we implement adequate visualization and orientation techniques. As concerning local coherency, each link must have a well-defined destination and we have to minimize the fragmentation, to avoid the confusion and getting lost.
- **Learning by doing** – We think that practice is essential in learning activities, because learning by doing increases substantially the effectiveness of learning processes. We elaborate practical e-learning tools, allowing the learners to work on experiments with real world items.
- **Active learning** – The experienced learners can take part in activities of design and elaboration of some particular e-learning scenarios for beginners.
- **Homework assignments** – Previous studies show that most traditional learners appreciate the homework assignments. However, in traditional teaching-learning relations, the teachers do not have the means to react properly to the individual problems the learners have when working out the assignments. This problem is overcome in case of a simulated teaching-learning relation, because the virtual teacher is always present in a running e-learning scenario.
- **Virtual teacher evaluates the solution** – Before the final solution is sent to the virtual teacher, each learner can obtain some hints and suggestions to handle the problem. After the final solution is sent, the learner gets the outcome for self-evaluation.

THE INFRASTRUCTURE

The infrastructure used to build intelligent and practical e-learning tools consists of Java libraries incorporating a set of simulation algorithms (Prodan, Gorunescu and Prodan, 1999; Prodan, Gorunescu, Prodan and Campean, 2000; Prodan and Prodan, 2001) and some XML based knowledge bases containing the integrated knowledge of a good teacher. We approached the first intelligent and practical e-course in domains of Statistics, Stochastic Modeling and Simulation, using as infrastructure a set of Java class libraries for stochastic modeling and simulation, created and implemented by Department of Mathematics and Informatics at Iuliu Hatieganu University (Prodan and Prodan, 2002). The papers (Prodan, Gorunescu and Prodan; Prodan, Gorunescu, Prodan and Campean) demonstrate the advantages of stochastic models for representation of real world activities and focuses on a Java package, which includes a collection of classes for stochastic modeling and simulation. In order to employ mathematics and statistics to analyze some phenomena and processes of the real world, we first construct a stochastic model. Once the theoretical model has been constructed, in theory we are able to determine analytically the answers to a lot of questions related to these phenomena and processes. However, in practice is very difficult to get analytically the answers for many of our questions. This is the reason why we must implement the probabilistic mechanism using a programming language, then to perform a simulation study on a computer. Due to actual spread of fast and inexpensive computational power everywhere in the world, the best approach is to model the real phenomenon as faithfully as possible, and then rely on a simulation study to analyze it. The Department of Mathematics and Informatics from Iuliu Hatieganu University has created and implemented a collection of Java class libraries for stochastic modeling and simulation. The stochastic models constructed accurately represent real world phenomena and processes, particularly in medicine and pharmacy. We have to use these Java libraries as an infrastructure to build intelligent and practical e-learning tools, integrated in an electronic educational environment.

We have considered three levels of simulation. The first level consists of simulating random numbers, as they are the basis of any stochastic simulation study (see Figure 2). Based on the elements of the first level, we created a second level of simulation applied for distributional models, stochastic processes and Monte Carlo methods. We implemented a hierarchy of Java classes which model the classical distributions and we created a collection of Java class libraries for stochastic modeling and simulation (Prodan and Prodan, 2001). Each distribution is determined by a distribution function and a set of parameters. Using these elements, we defined for each distribution a simulation algorithm based on one of the following techniques: Inverse Transform Technique, Acceptance-Rejection Technique and Composition Technique (Ross, 1990). We implemented all these simulation algorithms through a polymorphic simulation method. To generate a single value from a particular distribution, it is necessary to import the corresponding distribution class from a library, then to instantiate an object of that class, and finally to call a polymorphic method based on this object. The first two levels of simulation are the basis for the third level, which is devoted to applications. We used distributional models and stochastic processes to implement some stochastic modeling applications that accurately represent real world processes and phenomena, particularly in medicine and pharmacy (Prodan and Prodan, 2001; Gorunescu, Gorunescu and Prodan, 2002).

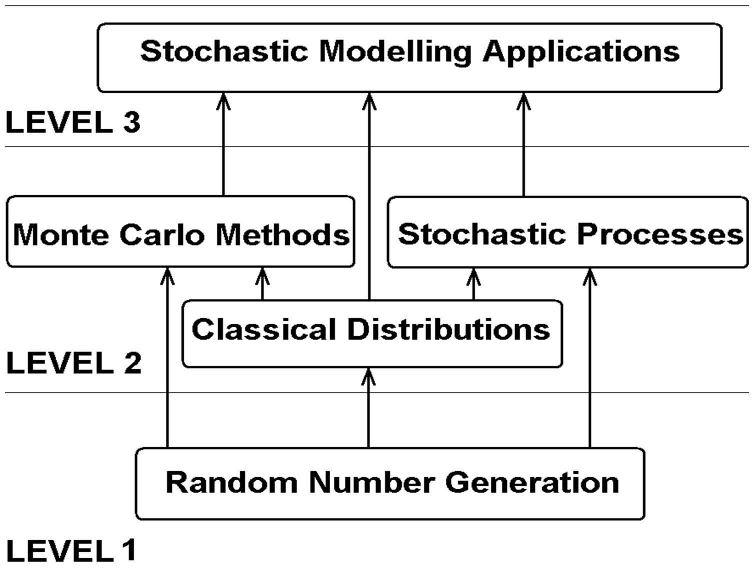

Figure 2. The Simulation Levels

We modeled the flow of patients through chronic diseases departments (Gorunescu, Gorunescu and Prodan, 2002). The planning of medical service within a chronic healthcare department is a complex problem the staff has to face, because patients of long-term services occupy the beds for long periods of time and a high quality medical care costs a lot of money. Under these circumstances, a balanced policy between a high quality service measured by the number of beds and suitable costs becomes a necessity for the administration in order to get full value for the money they have spent. With this end in view, using the simulation of distributional models and stochastic processes, we modeled the patient flow through chronic diseases departments. The use of stochastic compartmental analysis, which assumes probabilistic behavior of the patients around the system, is considered a more realistic representation of an actual situation rather than simpler deterministic model (Taylor, McClean and Millard, 1998). In order to simulate the model, we have split it into two parts: the arrival of patients and the in-patient care. Patients are initially admitted into acute care consisting of diagnosis, assessment and rehabilitation. The most of patients either are released and therefore re-enter the community, or die after such a period of acute care. However, a certain number of patients may be considered to be unable to look after themselves, and therefore pass from acute into long-stay care where they may remain for a considerable amount of time, or they will eventually die. The arrival of patients is modeled as a Poisson process with a parameter estimated by using the inter-arrival times (Prodan, Gorunescu and Prodan). These times are independent exponential random variables, each with parameter λ and with the corresponding density function $f(t) = \lambda e^{-\lambda t}$, $t \geq 0$. The in-patient care time is modeled by the application of a mixed-exponential distribution, where the number of terms in the mixture corresponds to the number of stages of patient care. A common scenario is that there are two stages for in-patient care: acute and long-stay, composing in this case two exponential distributions with parameters λ_1 and λ_2, representing the corresponding access rate for each stage. In this case, the mixed-exponential phase-type distribution has the density function $f(x) = \rho\lambda_1 e^{-\lambda 1t} + (1\text{-}\rho)\lambda_2 e^{-\lambda 2t}$, which implies a mean care time of $\rho/\lambda_1+(1\text{-}\rho)/\lambda_2$ days per patient (Prodan, Gorunescu, Prodan and Campean). This model enables us to study the whole system of geriatric care and can be used to look at the time patients spend in hospital and the subsequent time patients spend in the community. Interesting real phenomena can be studied during simulation experiences, such as rejection at entrance due to a brimful department, the resources being limited. Both medical staff and hospital administrators agreed that such a model could be used to maximize the efficiency of the chronic diseases departments and to optimize the use of hospital resources in order to improve hospital care.

E-LEARNING TOOLS FOR STOCHASTIC SIMULATION

Based on the infrastructure presented in previous section, we have to create intelligent and practical e-learning tools particularly for Statistics, Stochastic Modeling and Simulation. To create an intelligent e-learning tool, it is necessary to integrate and to insert into these tools the knowledge of the best teachers and the best pedagogical strategies. An e-learning scenario combines simulation and interactive visualization and allows the learners to explore the knowledge bases with some well-defined learning purposes. We define a simulation class and a visualization class for each application object. These classes are then configured to obtain a particular simulation with a specific visualization. The e-learning tools implemented by us incorporate Bayesian inference as a way of representing uncertainty, expressing beliefs and updating beliefs, using experimental results. In an e-learning scenario, visualization is an active part of the system, serving as an additional interface for modifying dynamically some parameters. We have to write all simulation and visualization classes in Java and use the XML format to describe the configurations, defining both the components and their relationships.

E-Learning Scenarios for Distributional Models

We have to create each e-learning scenario by looking at problems that can be put in a probabilistic framework. Every new concept is developed systematically through completely worked out examples from current medical and pharmaceutical problems. In addition, we have to introduce in each e-learning scenario specific probability models that fit out some real life problems, by assessing the probabilities of certain events from actual past databases.

As an example, we propose an e-learning scenario for students in Medicine. A learner that traverses such a scenario, will be able to apply a binomial distributional model in studying the chance of patients suffering from a particular type of cancer, to survive for at least a six month period after diagnosis. We would have to appeal to previous studies and information from actual databases to assess the chances of a patient surviving. This might indicate, for instance, that the probability of survival is p = 0.3, and consequently the complementary probability of death is $q = 1 - p = 0.7$. In real life, we are frequently interested what might happen to a group of patients we are studying. For example, we may formulate the following problem, as a piece of the current e-learning scenario:

"Of the 11 *patients in a particular cancer program, what is the chance of* 7 *or more of them surviving at least six months past diagnosis?"*

If p_k is the probability that k patients survives (where $k \leq 11$), to solve the previous problem we have to compute the sum $P = p_7+p_8+p_9+p_{10}+p_{11}$. The binomial distribution B (11, 0.3) can be applied to calculate these probabilities. If the learner is a beginner in Probabilities and Statistics, maybe he needs to get immediately an explanatory text about binomial distribution B (n, p). Then he has the possibility to directly apply the formula and compute the probabilities for binomial B (11, 0.3). Using the formula $p_k = 11!/(k!(11-k)!)(0.3)^k (0.7)^{11-k}$, for $k \leq 11$, he will obtain the values $p_7 \approx 0.017$, $p_8 \approx 0.003$, and $p_9 \approx p_{10} \approx p_{11} \approx 0$. Finally, the learner can give the solution for the previous problem: $P = 0.017 + 0.003 = 0.02$. The e-learning scenario may be resourceful in showing additional information when necessary. The learner may ask whenever for any information he need, but an intelligent software tool may have also the initiative to show some information when it considers that is an adequate moment. We think that Bayesian inference is suitable for deciding when the learner needs some supplementary information. We have to combine Bayesian inference with a sort of intelligent agents, to prepare and to configure a suggestive visualization, based on a friendly and efficient dialogue with the learner. As an example, for any learner may be useful to see a visualization of the previous probabilities p_k, where $k \in \{0, 1, 2, ..., 11\}$, in a very suggestive column format, and to recognize the solution to previous problem shown with dashed columns (see Figure 3).

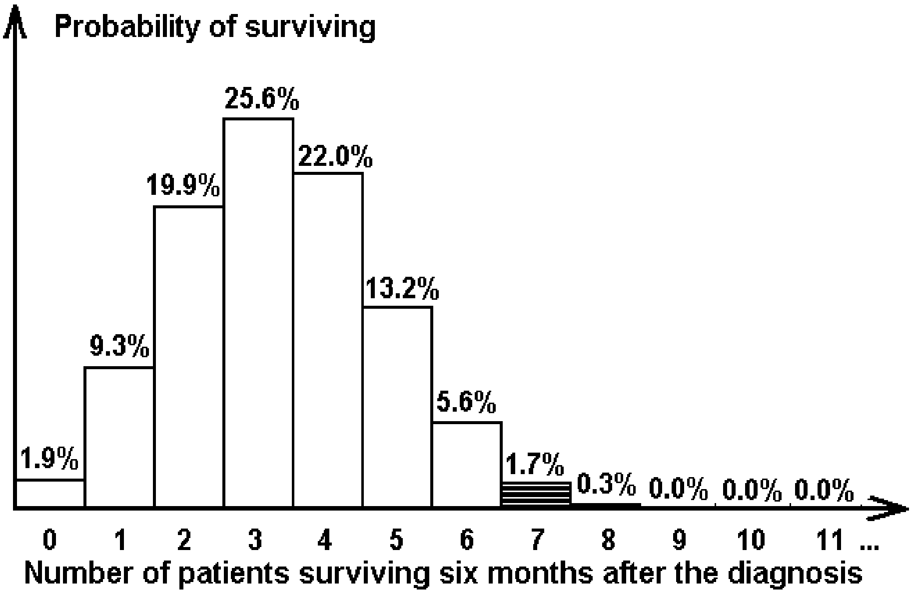

Figure 3. A graphical solution for patient surviving problem, in case of binomial distribution B (11, 0.3).

The previous distribution model, which is binomial distribution B (11, 0.3), may also be applied in an e-learning scenario for students in Pharmacy. For example, suppose they have to test the effect of a dose of digitalis on frogs. In this case, the problem can be formulated as follows:

"Injection of a certain dose of digitalis per unit of body weight into a large number of frogs causes the death of 30% *of them. What is the probability that the number of deaths will be* 7 *or more, when this dose is injected into each of a group of* 11 *frogs?"*

It is obvious that we may use the same statistical template to create a similar e-learning scenario for students in Pharmacy, but with specific texts.

E-Learning Scenarios for Simulation Studies

To make a simulation study for a distribution, it is necessary to generate more values, a sequence of values. The learner has the possibility to interactively select a distribution, to set some parameters, to visualize the results of a simulation and to form some conclusions. For example, the learner may choose to generate N=1000 values from previous binomial distribution B (11, 0.3) and to visualize them in a column format. In addition, the learner may visually verify the results of a simulation by comparing them with theoretical values, computed by theoretical formulae (see Figure 4). There is an obvious relation between theoretical values and the values presented in Figure 3.

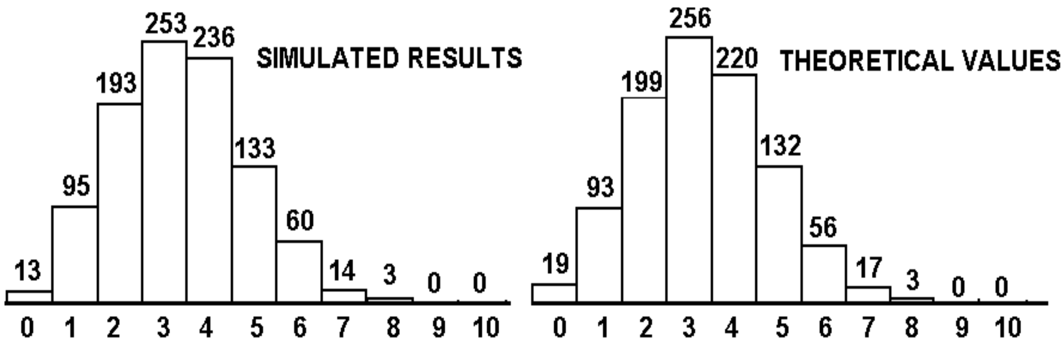

Figure 4. The simulated results versus theoretical valuesin case of the binomial distribution B (11, 0.3).

The learner may choose to continually generate additional values, stopping when the efficiency of the simulation is good enough. Generally, the learner may use the variance of the estimators obtained during the simulation study to decide when to stop the generation of additional values**,** that is when the efficiency of the simulation study is acceptable. The smaller this variance is, the smaller is the amount of simulation needed to obtain a fixed precision. For example, if the objective is to estimate the mean value $\mu = E(X_i)$, i=0, 1, 2, ..., the learner should continue to generate new data until he has generated n data values for which the estimate of the standard error, SE = $s / \sqrt{n}$, is less than an acceptable value given by him and named *Allowed SE*, s being the sample standard deviation. Sample means and sample variances are recursively computed. The final values of these parameters, the confidence interval and other statistics are showed as the result of the simulation study. Figure 5 compares the results of two simulations in case of a binomial distribution B (11, 0.3).

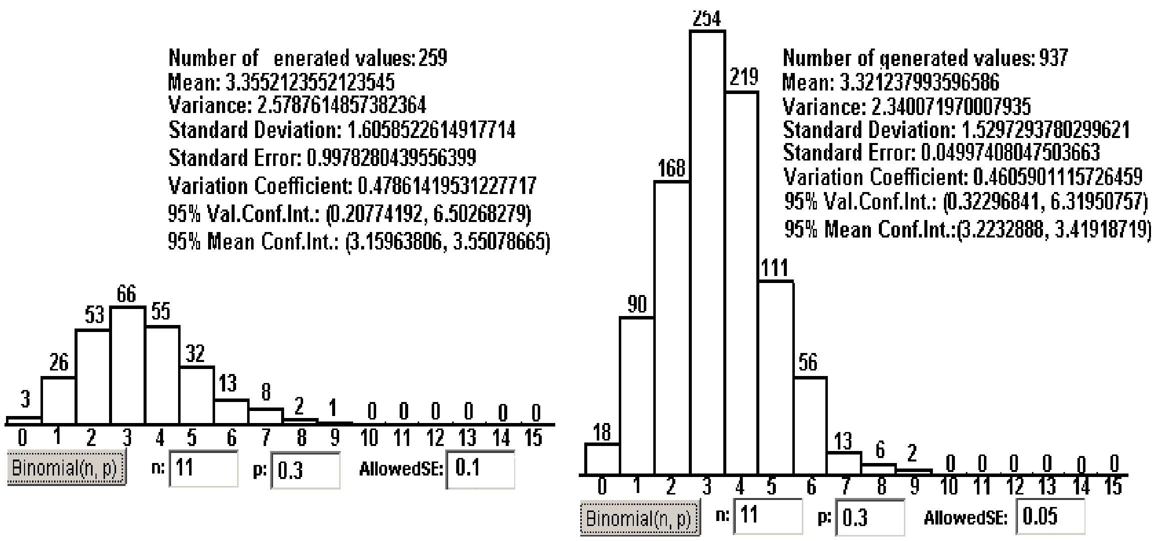

Figure 5. A simulation with *Allowed SE* = 0.1, versus a simulation with *Allowed SE*=0.05, in case of binomial distribution B (11, 0.3)

The number of generated values is determined by the value *Allowed SE* introduced by the learner, as the simulation stops when the condition $s / \sqrt{n} \le$ *Allowed SE* is true. The left graph is obtained as a result of a simulation with *Allowed SE* = 0.1, being generated 259 values, while the right graph is the result of a simulation with *Allowed SE* = 0.05, being generated 937 values.

When simulate from a continuous random variable *X*, a generated value $x \in X$ is approximated with a given *precision* expressed by the number of decimal digits to be considered. The learner has the possibility to choose a precision of one, two, or even more decimal digits. If a coarse approximation is accepted, no decimals are considered and the real value *x* is approximated by *int*(*x*), that is by integer part of the *x*. In this case the continuous random variable *X* is rudely approximated by a discrete one, and the results of a simulation can be graphically expressed in a segmented line format, each segment joining the top sides of two neighboring columns. If a precision of one decimal digit is selected, a more refined segmented line can more precisely visualize the results of the same simulation. With a precision of two decimal digits, a more refined visualization is obtained. The higher this precision is, the higher is the *resolution* realized in visualization. Figure 6 compares two visualizations for the same set of generated values from the standard normal distribution N (0, 1), the first visualization being with a precision of one decimal digit (Figure 6, graph a), and the second with a precision of two decimal digits (Figure 6, graph b).

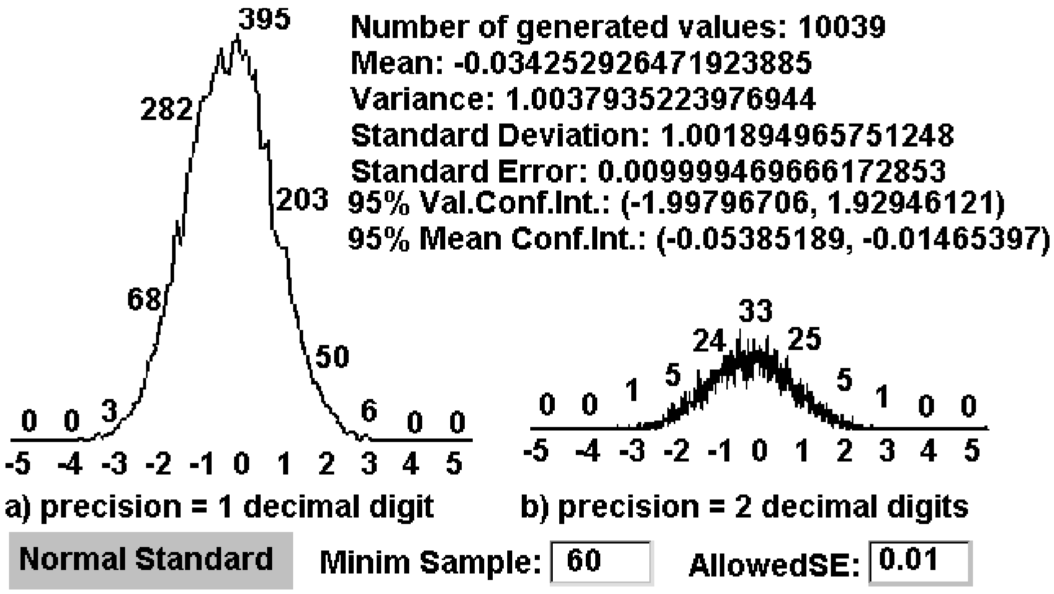

Figure 6. The same set of generated values from the Standard Normal distribution N (0, 1), visualized with a precision of one decimal digit, versus a precision of two decimal digits.

As can be seen in this figure, when the precision grows with one decimal digit, the resolution grows ten times. With a precision of one decimal digit, ten numbers are considered between two successive integers, while if the precision is of two decimal digits, one hundred numbers are considered between two successive integers. When necessary, intermediate resolutions can be considered. Figure 7 shows a similar comparison for the exponential distribution with parameter $\lambda = 0.3$.

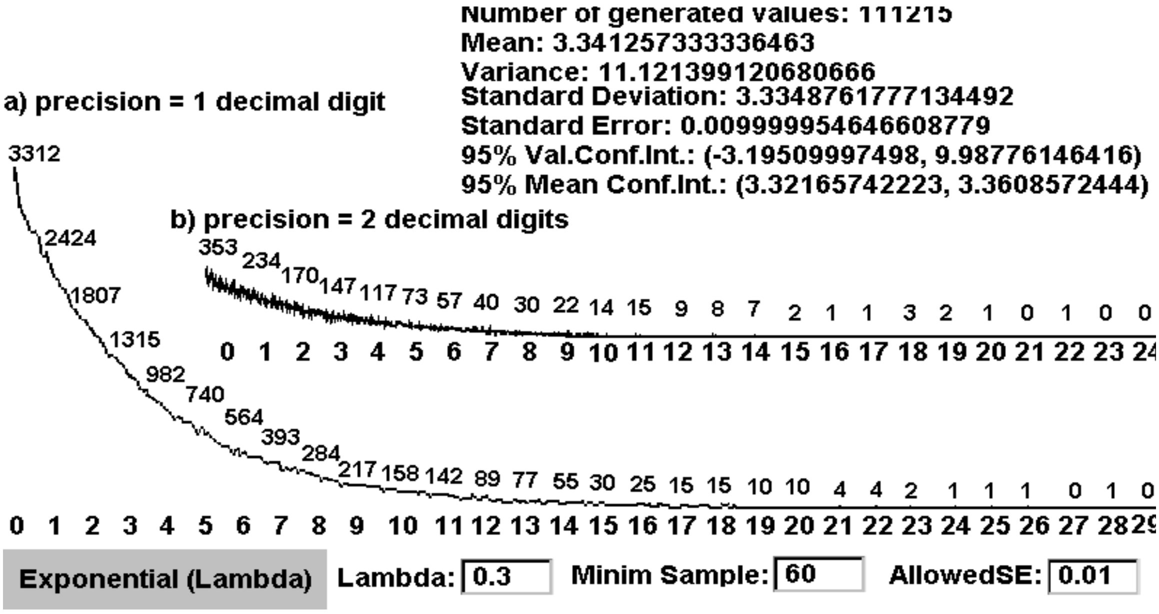

Figure 7. The same set of generated values from the exponential distribution E (0.3), visualized with a precision of one decimal digit, versus a precision of two decimal digits.

FUTURE WORK

As a future work, we will improve the feedback mechanism with new CBR strategies, with the purpose to exploit previous successful experiences and to avoid previous failures. We will implement software tools which will include Bayesian inference as a way of representing uncertainty, expressing beliefs and updating beliefs, using experimental results. These software tools will assist us, as human beings, to learn from past experiments and revise our knowledge on issues around us. Also, we will create agent based e-learning scenarios, able to map, to plan and to monitor the pace and the progress of learning processes. In near future we intend to cooperate with experts in medicine and pharmacy to create various intelligent and practical e-learning tools for these domains.

CONCLUSIONS

In this paper, we presented a Java framework for design and building the intelligent and practical e-learning tools, to be used by both the students and the teaching staff, especially in a context of open learning. We use an infrastructure consisting of Java class libraries for stochastic modeling and simulation. The learner has the possibility to interact with specific Java applets, to select specific values for some parameters, obtaining this way specific simulations and visualizations. As the real teacher is not physically present during an open learning process and the electronic technologies do not contain pedagogical characteristics, we have to simulate a valuable teaching-learning relation and to incorporate the teaching activities in our e-learning software, obtaining this way intelligent e-learning tools. We use Java, XML and CBR technologies to elaborate e-courses consisting of intelligent and practical e-learning scenarios, allowing the learners to work on experiments with real world items.

REFERENCES

Eckel, B. (1998). Thinking in Java (MindView Inc., Prentice Hall PTR). Available from http://www.mindview.net/Books/TW/

Gorunescu, M., Gorunescu, F., and Prodan, A. (2002). Continuous-time Markov Model for Geriatric Patients Behavior. Optimization of the Bed Occupancy and Computer Simulation, Korean Journal of Computational and Applied Mathematics, 9 (1), 185-195.

Leake, D. P., (1996). Case-Based Reasoning: Experiences, Lessons, and Future Directions, AAAI Press / MIT Press, Menlo Park, CA.

Nwana, H. S., (1996). Software Agents: An Overview, Knowledge Engineering Review, 11(3), 205-244.

Prodan, A., (1996). Quelques barriers a franchir dans la relation enseignement-apprentisage, Flash Informatique, E.P.F. Lausanne, 23-26. (http://sawww.epfl.ch/SIC/SA/publications/FI96/).

Prodan A. and Prodan, M., (1997). Java Environment for Internet, ProMedia plus Ed., Cluj-Napoca, Romania.

Prodan, A., (1998). Towards a new type of open learning, Clujul Medical, LXXI (3), 424-431.

Prodan, A., Gorunescu, F., and Prodan, R., (1999). Simulating and Modeling in Java, Workshop POOSC'99 (Parallel/High-Performance Object-Oriented Scientific Computing), Lisbon, Technical Report FZJ-ZAM-IB-9906, Forschungszentrum Julich GmbH, 55-64.

Prodan, A., Gorunescu, F., Prodan, R. and Campean, R., (2000). A Java Framework for Stochastic Modeling, Proceedings of the IMACS'2000, (16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation), Lausanne, CD Ed. Rutgers University, New Brunswick – NJ, USA, ISBN 3-9522075-1-9.

Prodan, A. and Prodan, Rodica, (2001). Stochastic Simulation and Modeling, Proceedings of the ETK-NTTS'2001 (Exchange of Technology and Know-how – New Techniques and Technologies for Statistics), Crete, 461-466.

Prodan, A. and Prodan, R., (2002). A collection of Java class libraries for stochastic modeling and simulation, Proceedings of the ICCS'2002 (International Conference on Computational Science), Amsterdam, Springer, Sloot et al.(Eds.), Part. I, LNCS 2329, 1040-1048.

Race, P., (1989). The Open Learning Handbook, Nichols Publishing, London, UK.

Ross, S. M., (1990). A Course in Simulation. Macmillan Publishing Company, New York, 2nd edition.

Taylor, G. J., McClean, S., Millard, P., (1998). Continuous-time Markov Models for Geriatric Patient Behavior, Appl. Stochastic Models and Data Analysis, 13, 315-323.

Wooldridge, M and Jenings, N., (1995). Intelligent Agents: Theory and Practice, Knowledge Engineering Review, 10 (2), 115-152.

Augustin Prodan
Mathematics and Informatics Department
Iuliu Hatieganu University
3400 Cluj-Napoca
Romania
E-mail: aprodan@umfcluj.ro