

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



E-Learning Tools as Means for Improving the Teaching-Learning Relation

Augustin Prodan¹, Paulina Mitrea²,
Mădălina Rusu¹, Cornelia Revnic¹ and Remus Câmpean¹

¹*Iuliu Hațieganu University Cluj-Napoca*

²*Technical University Cluj-Napoca
Romania*

1. Introduction

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). 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. This is the reason why our approach to e-learning is so called blended learning, which combines traditional face-to-face and computer-based learning, with focus on principles of active learning.

The main goal of our work is to improve the teaching-learning relation by means of e-learning tools. For this purpose, we used Java and XML technologies to implement practical and intelligent e-learning tools (Prodan et al., 2002, 2004, 2006, 2008, 2010), which we began to incorporate in a Moodle based e-learning system (Martinez & Jagannathan, 2008).

We started our researches concerning e-learning technologies in the last years of the previous century, when we created a pool system consisting of e-courses to be used in a context of open learning. We observed 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. However, we considered this new type of open learning not as a substitute, but as a supplement for traditional learning (Prodan, 1998).

We rely on new paradigms of artificial intelligence (Bayesian Inference, Case Based Reasoning and Intelligent Agents) for creating e-learning scenarios to be used in a context of active learning. An e-learning scenario combines simulation and interactive visualization

and allows the learners to explore the knowledge bases with some well-defined learning purposes. For each application object, our system contains a simulation class and a visualization class. These classes are then configured to obtain a particular simulation with a specific visualization. In an e-learning scenario, visualization is an active part of the system, serving as an additional interface for modifying dynamically some parameters. The simulation and visualization classes are coded in Java, using XML format to describe the configurations for both the components and their relationships.

We use Internet as communication support in our e-learning technologies. Internet provides a general communication space, which we use as the base line infrastructure for creating e-learning spaces. We created e-learning spaces around various subjects, such as biostatistics (Prodan et al., 2004), pharmaceutical botany, analyzing medical images (Prodan et al., 2006, 2008, 2010), etc. Previously, each teacher used a private e-course organizational structure for learning management.

The main final result will be a student-centred, cost effective, Moodle based e-learning system (Martinez & Jagannathan, 2008), supporting the benefits of blended learning pedagogical models (Francis & Raftery, 2005) and allowing to improve the teaching-learning relation.

In section 2 we present a short literature review for some e-learning tools we intend to consider in our effort to enhance the teaching-learning relation. In section 3 we propose an e-learning environment based on a Java framework for designing and implementing e-learning tools. Section 4 shows some results and experiments with e-learning scenarios and finally section 5 states some conclusions and future work.

2. E-learning tools

E-learning comprises all forms of electronically supported learning and teaching, the support being provided usually by e-learning tools. In the teaching-learning equation, the learning side is more important. This is the reason why the word “e-learning” is more frequent used than “e-teaching”. Generally, the efficiency of the learning depends on teaching-learning relation and we think that it is possible to improve this relation by e-learning tools. We use Java and XML technologies to create original e-learning tools and we think the best way to follow for improving the teaching-learning relation is by using electronic portfolios and by incorporating all these e-learning tools in a Moodle based e-learning system. The subsection 2.1 illustrates some general ideas about electronic portfolios, while subsection 2.2 presents the reasons why we decided to approach a Moodle based e-learning system.

2.1 Electronic portfolios

Our students have different backgrounds, interests, levels of motivation and approaching to studying, therefore we considered that open learning is adequate to them. To improve the teaching-learning relations, we included electronic portfolios in our e-learning technologies. The Java framework provides the infrastructure for preparing e-learning scenarios based on practice and real world experiences, as practice is essential in learning activities. Our e-learning scenarios promote active learning, forcing the students to take part in real world activities, simulated on computer.

To better the human contact between students and instructor, we approached new strategies labelled as blended learning, combining e-learning and traditional face-to-face classroom instruction, with a focus on active learning. We implement e-learning scenarios relying on learner centred paradigm, where learners are encouraged to develop skills and strategies in

their own way. Learning is not considered simply as an outcome of teaching, because it is an activity having as input the results of teaching and training. Blended learning should be viewed as a fundamental redesign of the instructional model. A blended learning context can provide the independence and increased control essential to developing critical thinking. Along with the increased control that a blended learning context encourages is a scaffolded acceptance of responsibility for constructing meaning and understanding (Garrison & Kanuka, 2004). To be a critical thinker is to take control of one's thought processes and gain a metacognitive understanding of these processes, i.e., learn to learn.

In education, portfolios are described as a meaningful collection of students' work stored in a traditional folder (Farr, 1990). There are various types of portfolios, organized according to the purpose they are used for:

- Learning portfolios - used for supporting the learning processes and the on-going professional development;
- Teaching portfolios - used for supporting the teaching processes;
- Assessment portfolios - used in evaluation processes;
- Employment portfolios - used in seeking jobs.

The electronic portfolio was introduced later and represents the same collection of students' work, only this time the storage is not the traditional folder, but an electronic storage environment, such as web pages, files organized in folders and stored on a CD or on a dedicated server, etc (Barret, 2001). An electronic portfolio simply means that the portfolio is technology based. There are many benefits of electronic portfolios, versus traditional portfolios. Electronic portfolios take up little physical space, can hold a great deal of information and may be accessed with minimal effort. A learning portfolio is a collection of student work over a period of time, resulted from activities in the e-learning environment. Electronic portfolios may improve the teaching-learning relations (Rusu & Prodan, 2006). The students have a constant project to work on being actively involved in the learning process, and their motivation has increased visibly. Both the teachers and the students agreed that portfolios provide a better assessment than the traditional testing: learners can have their learning process assessed and are offered the chance to reflect on their own work. Portfolios allow constructive feedback from tutors, increasing cooperative learning and students' motivation. Portfolio assessment helps students enjoy the assignments, while enabling them to learn more easily and take an active part in their development.

2.2 Moodle based e-learning environment

This year we decided to approach Moodle in our e-learning environment. The main reason for such a decision is that Moodle was meant to embody better a student-centred paradigm, enabling to improve the teaching-learning relations. Moodle is an open source LMS (Learning Management System), so we benefit from the availability of the source code and the right to modify it. Also, Moodle gives us a lot of flexibility and is inexpensive, allowing us to face the present-day budget crisis. Other important qualities of the Moodle result just from his title. The word Moodle is an acronym, standing for Modular Object-Oriented Dynamic Learning Environment. Being modular, we can add and remove modules, considering our actual needs and requirements, and our previous experience. The programming paradigm used in Moodle is Object-Oriented as in Java. We created in Java an e-learning infrastructure and e-learning scenarios, focusing on real life objects. It is no difficult to move all our Java classes into Moodle, the programming paradigm being the

same. Moodle is a powerful tool able to provide dynamic web pages, employed to create dynamic teaching-learning relations. In our previous e-learning experiments, we used this valuable characteristic. To implement the e-learning scenarios, we used Java technologies for dynamic processes and XML technologies for dynamic content (data and documents).

A valuable pedagogical advantage of the Moodle, arising at both teaching and learning level, is constructivism. Teachers construct their own teaching strategies and cooperate for enhancing these strategies, using a feedback with reactive suggestions. Also, learners use e-learning modules that allow peer to peer interaction and cooperation, acting and thinking cooperatively.

3. E-learning environment

We defined and implemented 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 a context of open learning. This framework provides the infrastructure for preparing e-learning scenarios based on practice and real world experiences, as practice is essential in learning activities. Our e-learning scenarios promote active learning, forcing the students to take part in real world activities simulated on computer. Also, we designed e-learning tools based on bootstrapping methods (which are quite valuable for reasoning in uncertain conditions), with the purpose to simulate laboratory experiments in both didactic and research activities.

An e-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 (Prodan et al., 2008). A good traditional teacher learns all the time from previous didactic experiences. Based on this historical feedback, the teacher exploits prior specific successful episodes, and avoids prior failures. We introduce a similar feedback mechanism in our technology of elaborating e-learning scenarios (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

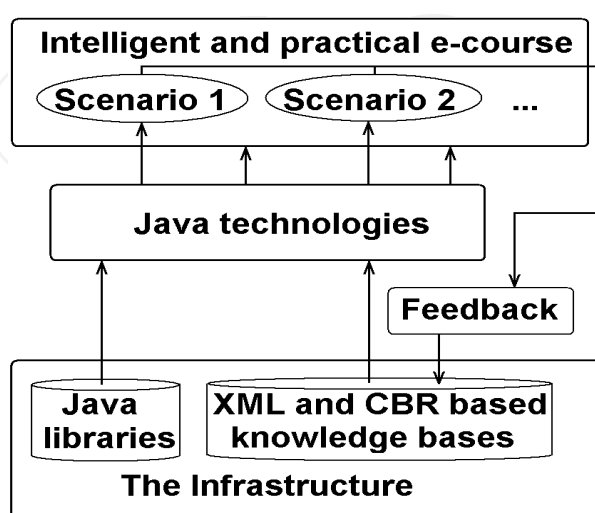


Fig. 1. The generation of the e-learning scenarios.

(Nwana, 1996), able to incorporate knowledge, teaching methods and pedagogical characteristics into e-courses (Kanuka 2008). We intend 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. Following the traditional model, the cases of positive experiences from previous e-learning scenarios are stored into case bases created with XML and CBR (Case-Based Reasoning) technologies (Leake, 1996).

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 created first a particular infrastructure containing the knowledge from particular domains of the target e-course. For this purpose, we generated consistent Java and XML based knowledge bases, containing integrated knowledge of the best teachers. In addition, we implemented in Java a set of simulation algorithms describing real world phenomena, processes and activities we have to include in e-courses. When generating a new e-learning scenario, we use a feedback mechanism based on our historical experience from previous e-learning scenarios (Figure 1). Also, we use Java technologies to generate intelligent and practical e-learning scenarios, based on new AI (Artificial Intelligence) paradigms, such as Case-Based Reasoning, Bayesian Inference and Intelligent Agents (Prodan et al., 2006, 2008, 2010). A learner can access the e-course and launch an e-learning scenario either locally, or via WWW in a context of distance learning.

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, our system provides 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, it is necessary 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 it is necessary 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.

3.1 Stochastic modelling and simulation

We approached e-courses in domains of biostatistics, stochastic modelling and simulation, using as infrastructure a set of Java class libraries for stochastic modelling and simulation, created and implemented by us (Prodan & Prodan Radu, 2002). The papers (Prodan et al., 1999, 2000) 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 modelling and simulation. In order to employ mathematics and statistics to analyze some phenomena processes and activities 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. We created and implemented a collection of Java class libraries for stochastic modelling and simulation. The stochastic models constructed accurately represent real world phenomena processes and activities, particularly in medicine, pharmacy and health care. We 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 (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 modelling and simulation (Prodan & Prodan Rodica, 2001).

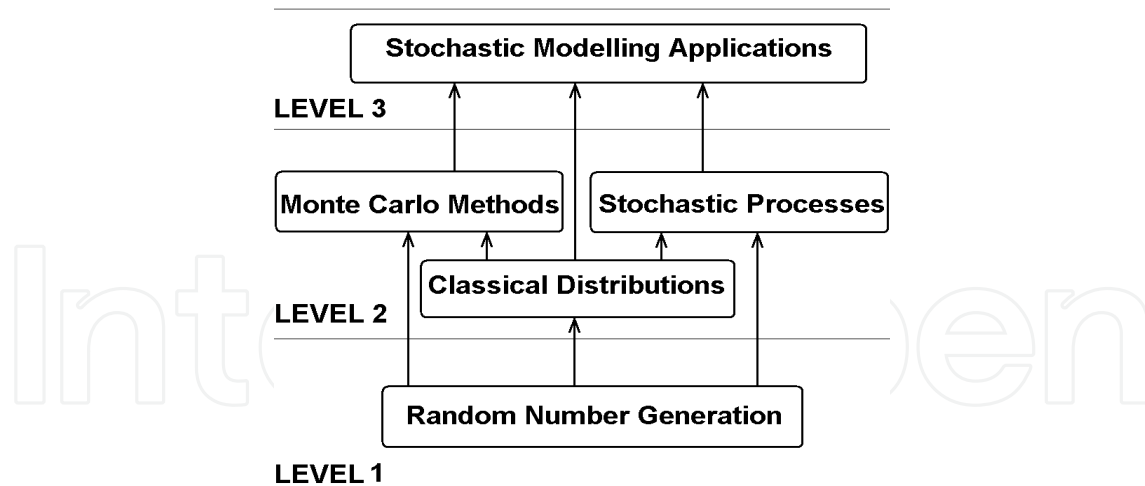


Fig. 2. The simulation levels.

The first two levels of simulation are the basis for the third level, which is devoted to applications. We used distributional models, stochastic processes and Monte Carlo methods to implement some stochastic modelling applications that accurately represent real world processes, phenomena and activities, particularly in medicine, pharmacy and health care (Prodan & Prodan Rodica, 2001; Gorunescu et al., 2002).

3.2 Java infrastructure for distributional models

The classical random variables are the simplest stochastic models, also called distributional models, which enter into the composition of other complex models. We propose a hierarchy of Java classes for modelling the classical distributions. Each distribution class encapsulates a particular `simValue()` method (Figure 3).

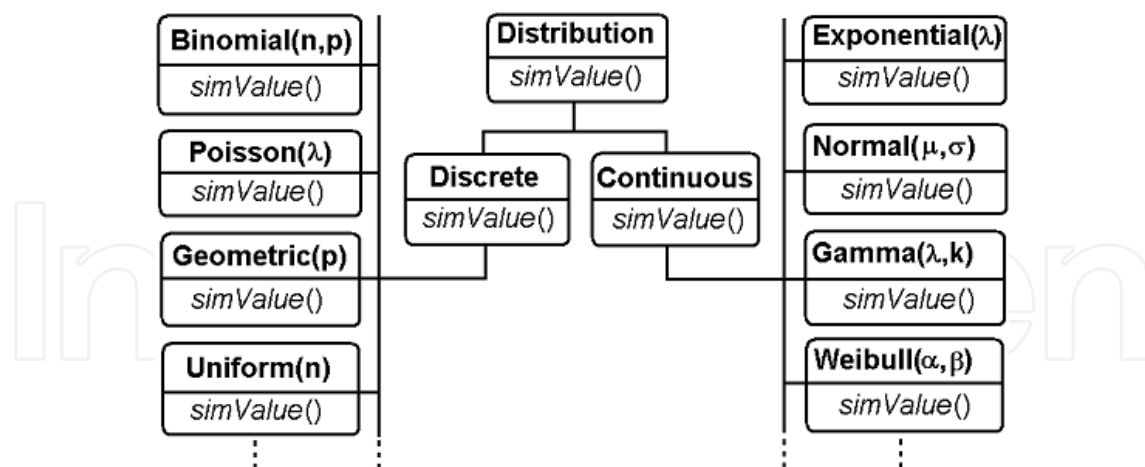


Fig. 3. The hierarchy of Java classes for distributional models.

Each distribution is determined by a set parameters and a distribution function. Based on these elements we defined a Java class for each distribution, obtaining a generic distributional model (Figure 4). This Java class is used to create one or more objects, which are instances with particular values for parameters. Also, a simulation algorithm is defined for each class, which is able to generate a specific value for the corresponding distribution. So, there is a set of simulation algorithms belonging to the whole hierarchy of Java classes,

these algorithms being implemented via the polymorphic method called `simValue()`. The particular implementation of the simulation algorithm for each class is based on one or more of the following techniques: the Inverse Transform Technique, the Acceptance-Rejection Technique and the Composition Technique (Ross, 1990).

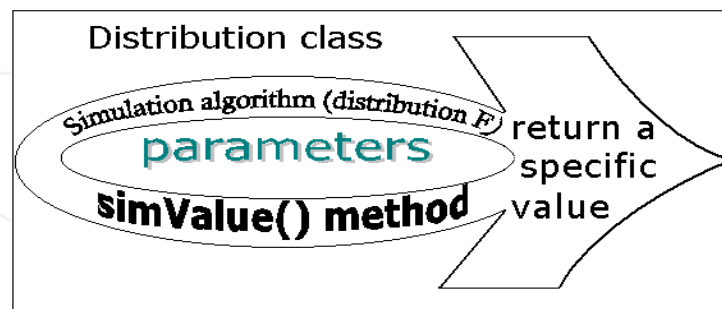


Fig. 4. The generic distributional model.

The hierarchy of Java classes for modelling the classical distributions is included in a JAR (Java ARchieve) library. To generate a single value from a particular distribution, it is necessary to import the corresponding distribution class from the library, then to instantiate an object of that class, and finally to call the polymorphic `simValue()` method based on this object (Figure 5). An instance of a particular class can be used to simulate a sequence of values for the corresponding random variable, by calling `simValue()` as many times as needed.

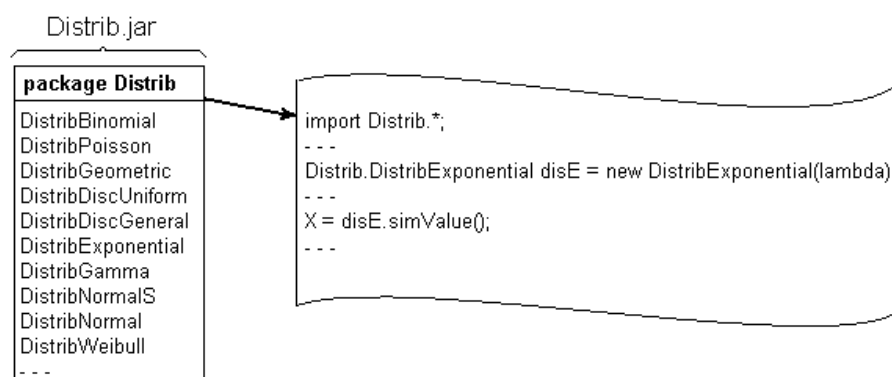


Fig. 5. The import of the distribution classes from the JAR library.

To make a simulation study for a distribution, it is necessary to generate more values, a sequence of values. One may choose to continually generate additional values, stopping when the efficiency of the simulation is good enough. Generally, one may use the variance of the estimators obtained during the simulation study to decide when to stop the generation of additional values. For example, if the objective is to estimate the mean value $\mu = E(X_i)$, $i = 0, 1, 2, \dots$, one may continue to generate new data values until one has generated a number of data values for which the estimate of the standard error is less than an acceptable value.

3.3 Bootstrapping e-tools for simulating laboratory works

Laboratory works play a fundamental role in learning environments, allowing learners to operate on experiments with real world items. Practice is essential in learning activities, because learning by doing increases substantially the effectiveness of the learning processes. Practical exercises are designed to prompt conceptual reasoning and critical thinking which

prompts the application of science to everyday situations. Heavy emphasis is put on detailed laboratory reports, written answers to exercises, completed homework assignments, tests and group work. Often we must repeat the same laboratory experiment many times, to make our utmost for all the students to participate and watch it. Students learn to ask questions, design experiments and analyze data, facts and theories. Also, statistical methods play a fundamental role in our university's didactic and research activities. Both the students and the teaching staff use traditional statistical methods to infer the truth from sample data gathered in laboratory experiments.

However, there are some reasons to minimize the number of laboratory experiments. In some cases, the repeated laboratory experiments mean the consumption of a great deal of substances and reactants. In other cases, there is a range of ethically motivated reasons to reduce the number of animals (frogs, guinea pigs, etc) used in experimentation. As the previous conditions are quite contradictory, we have to search for a solution that balances them. We perform an actual laboratory experiment only once for each group of students, then we repeat the same laboratory experiment by simulations as many times as necessary, each student being able to take part in analyzing data, making exercises and inferences.

We implemented bootstrapping methods into software tools, as applications in our simulation system (Figure 2), with the purpose to simulate laboratory experiments in both didactic and research activities (Prodan & Câmpăan, 2005). Usually, the result of a laboratory experiment is a set of output values v_1, v_2, \dots, v_n , named original, or actual sample. The experimenter (student, teacher, or researcher) uses this sample to perform data analysis and statistical inferences, then to draw conclusions. Bootstrapping methods involves repeating the original data analysis procedure with many replicate sets of pseudo-data. We implemented a general discrete distribution (Figure 10) to generate sets of pseudo-data, based on original set of data v_1, v_2, \dots, v_n . By resampling the original data, we generate artificial samples on which we make inference of interest as for the original sample. Using a bootstrapping e-tool and the computer power, the experimenter can repeat the original experiment without any consumption of substances and reactants, and without use of animals, obtaining pseudo-data as plausible as those obtained from the original experiment. Our bootstrapping tools use the computer power to obtain reliable standard errors, confidence intervals and other measures of uncertainty for a wide range of problems. Moreover, the bootstrap can produce a more accurate confidence interval than would produce based on normal approximation.

4. Results and experiments

In the first phase of our research we designed and implemented a hierarchy of Java classes for distributional models which we included in a JAR library, to be used as an infrastructure for stochastic modelling and simulation. We present in subsection 4.1 the main distributions we implemented in this library. Based on this infrastructure we created e-learning scenarios to be used in a context of active learning, relying on practice and real world experiences. Subsection 4.2 shows some of these e-learning scenarios.

4.1 The implementation of distributional models

We implemented a hierarchy of Java classes for distributional models (Figure 3). Java is the standard programming language for Internet applications (Prodan & Prodan Mihai, 1997); this is the reason why we used this language to create the infrastructure for e-learning scenarios. We present below Java classes implemented for some distributions, as well as simulation examples.

Binomial distribution

```

package Distrib;
public class DistribBinomial extends DistributionDiscrete {
    public DistribBinomial(int n, double p) { // Constructor
        this.n = n;
        this.p = p;
    }
    public double simValue() { // Simulation of a value
        double U, rap, pp, F;
        U = Math.random();
        rap = p/(1-p);
        pp = Math.exp(n*Math.log(1-p)); // probability X=i
        F = pp; // F(i)=P{X<=i} distribution function
        double k = 0;
        while (U > F) {
            pp = (rap*(n-k)/(k+1))*pp;
            F = F + pp;
            k++;
        }
        return k;
    } // simValue()
    public double initRecursion() { // Initial value for recursion
        return Math.pow(1-p, n); // P{X=0}
    }
    public double valRecursion(int k) { // Value for recursion
        return (double)(n-k)*p/(k+1)/(1-p);
    }
    int n;
    double p;
} ///;

```

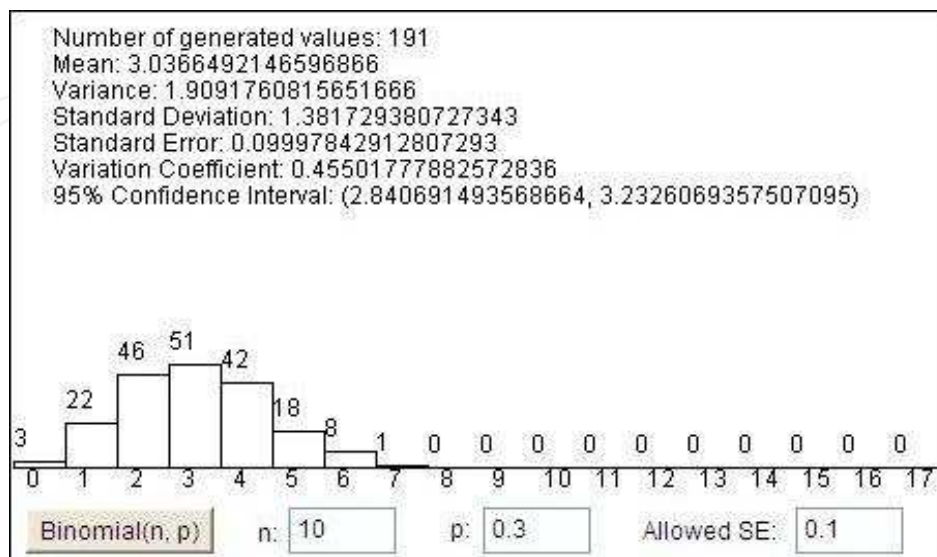


Fig. 6. Simulation from binomial distribution.

Poisson distribution

```

package Distrib;
public class DistribPoisson extends DistributionDiscrete {
    public DistribPoisson(double lambda) { // Constructor
        this.lambda = lambda;
    }
    public double simValue() { // Simulation of a value
        double U, pp, F;
        U = Math.random();
        pp = Math.exp(-lambda); // probability X=i
        F = pp; // F(i)=P{X<=i} distribution function
        double k=0;
        while (U > F) {
            pp = lambda*pp/(k+1);
            F = F + pp;
            k++;
        }
        return k;
    } // simValue()
    public double initRecursion() { // Initial value for recursion
        return Math.exp(-lambda); // P{X=0}
    }
    public double valRecursion(int k) { // Value for recursion
        return (double)lambda/(k+1);
    }
    double lambda;
} ///;

```

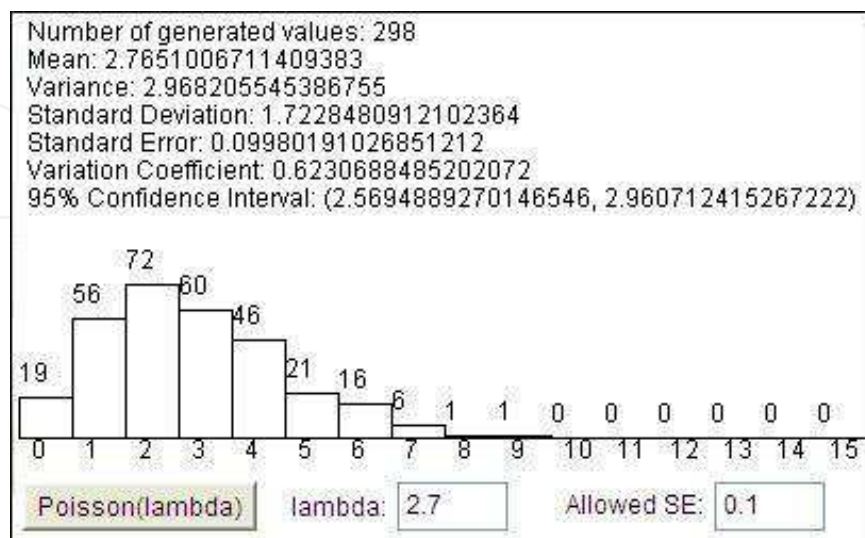


Fig. 7. Simulation from a Poisson distribution.

Geometric distribution

```

package Distrib;
public class DistribGeometric extends DistributionDiscrete {
    public DistribGeometric(double p) { // Constructor
        this.p = p;
    }
    public double simValue() { // Simulation of a value
        double U; // geometric random variable in (0,1)
        U = Math.random(); // Generate a random number
        return (int)(Math.log(U)/Math.log(1-p));
    } // simValue()
    public double initRecursion() { // Initial value for recursion
        return p; // P{X=0}
    }
    public double valRecursion(int k) { // Value for recursion
        return 1-p;
    }
    double p;
} ///;

```

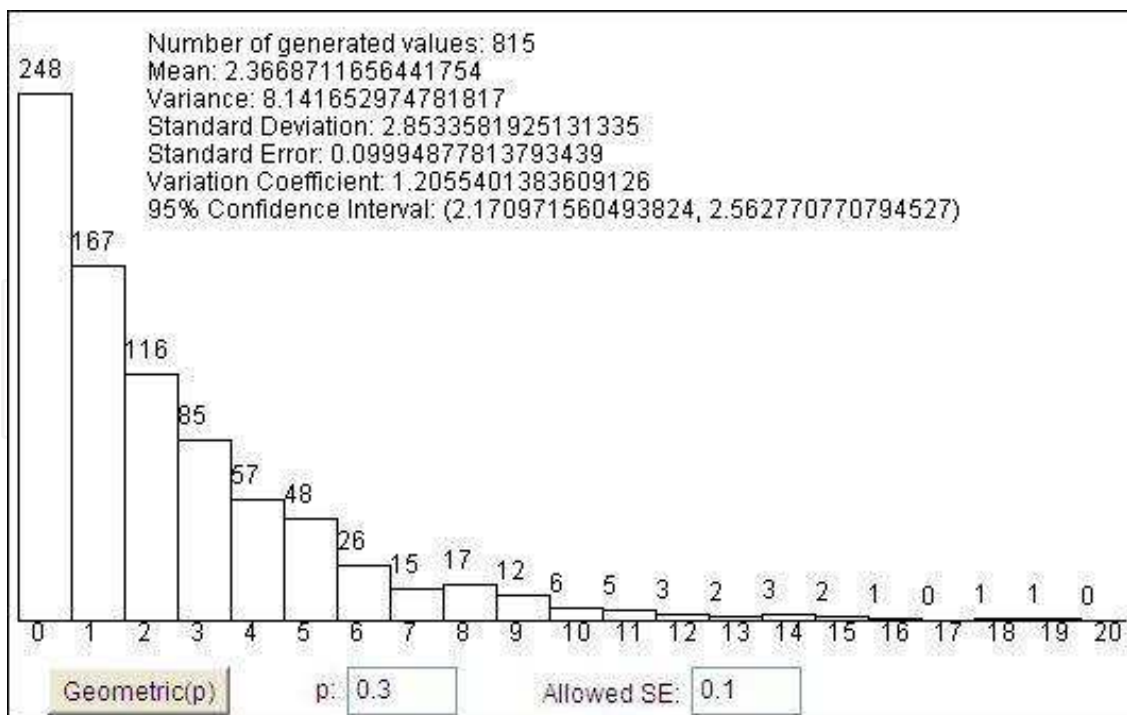


Fig. 8. Simulation from a geometric distribution.

Uniform discrete distribution

```

package Distrib;
public class DistribDiscUniform extends DistributionDiscrete {
    public DistribDiscUniform(int numb) { // Constructor
        n = numb;
        p = (double)1/n;
    }
    public double simValue() { // Simulation of a value
        int X; // Random variable
        double U; // Uniform random variable in (0,1)
        U = Math.random(); // Generate a random number
        return (int)(n*U);
    } // simValue()
    public double initRecursion() { // Initial value for recursion
        return p; // P{X=0}
    }
    public double valRecursion(int k) { // Value for recursion
        return (double) 1;
    }
    int n;
    double p;
} ///;

```

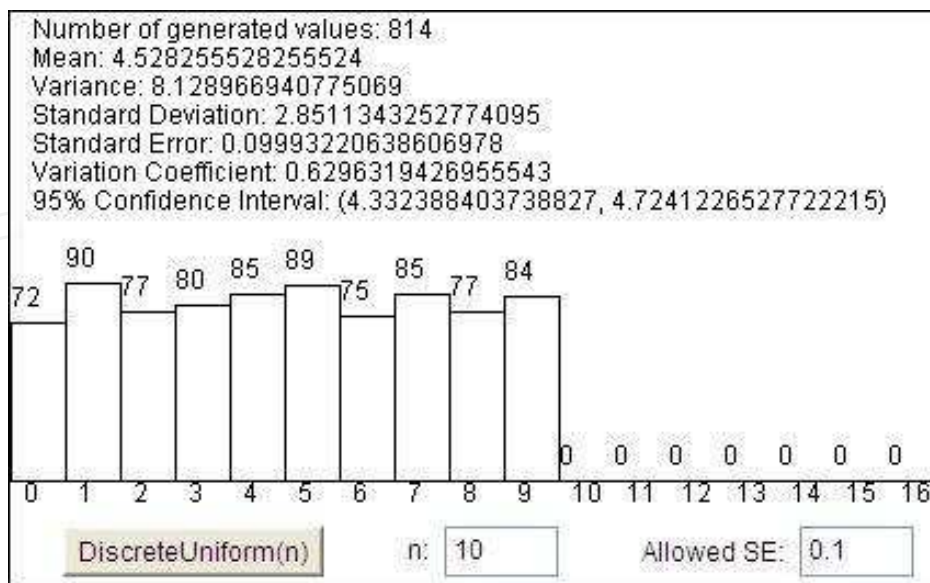


Fig. 9. Simulation from a uniform discrete distribution.

General discrete distribution

```

package Distrib;
public class DistribDiscGen extends DistributionDiscrete {
    public DistribDiscGen(int n, double[] p) { // Constructor
        this.n = n;
        this.p = p;
        for (int i=0; i<p.length; i++) {
            if (pMax<p[i]) pMax=p[i];
        }
        c = pMax*n;
    }
    public double simValue() { // Simulation of a value
        Distrib.DistribDiscUniform dDU = new DistribDiscUniform(n);
        do {
            Y = (int)dDU.simValue();
            U = Math.random();
        }
        while (U > (double)(p[Y]/(c/n)));
        return Y;
    } // simValue()
    int n, Y;
    double pMax=0, c, U;
    double[] p = new double[n];
} ///;

```

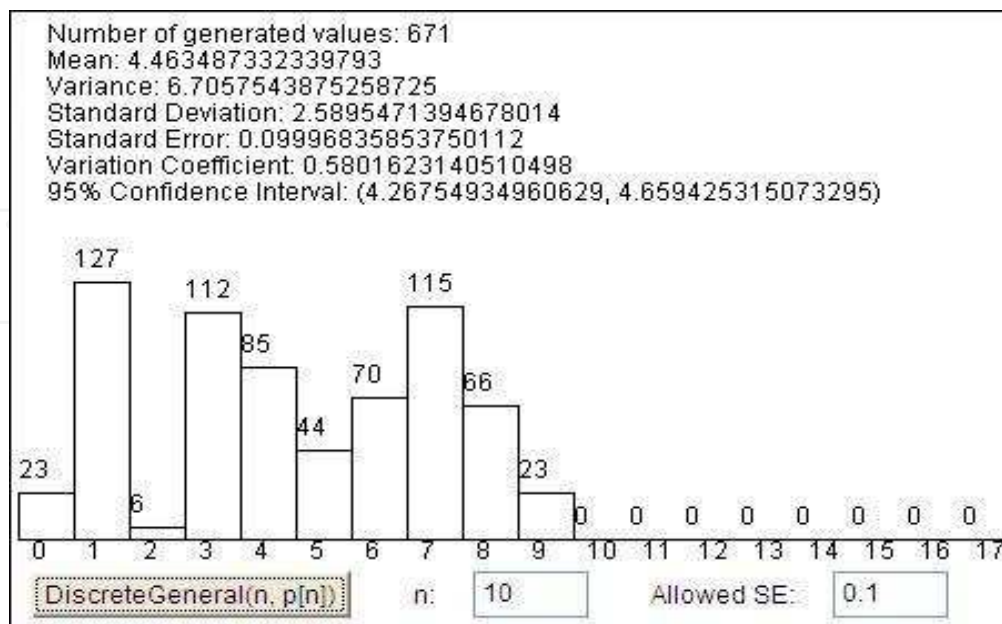


Fig. 10. Simulation from a general discrete distribution.

Normal standard distribution

```

package Distrib;
import Distrib.*;
import java.util.*;
public class DistribNormalS extends DistributionContinue {
    public double simValue() { // Simulation of a value
        double y1, y2, y, Z, U;
        Distrib.DistribExponential Y1 = new Distrib.DistribExponential(1);
        Distrib.DistribExponential Y2 = new Distrib.DistribExponential(1);
        Random R = new Random();
        do {
            y1 = Y1.simValue();
            y2 = Y2.simValue();
            y = y2-Math.pow(y1-1, 2)/2;
        } while (y<0);
        U = R.nextDouble();
        if (U<0.5) Z = y1;
        else Z = -y1;
        return Z;
    } // simValue()
} // //;

```

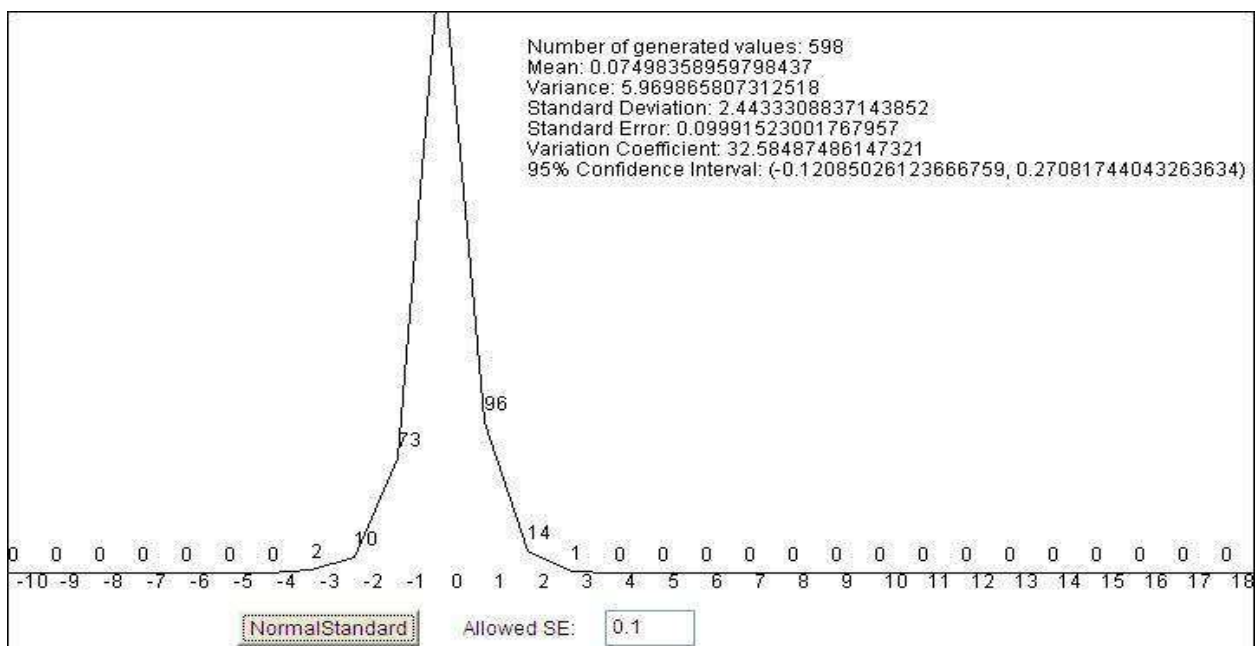


Fig. 11. Simulation from normal standard distribution

Normal distribution

```

package Distrib;
public class DistribNormal extends DistributionContinue {
    public DistribNormal(double miu, double sigma) { // Constructor
        this.miu = miu;
        this.sigma = sigma;
    }
    public double simValue() { // Simulation of a value
        double Z;
        Z = NS.simValue();
        X = miu+sigma*Z;
        return X;
    } // simValue()
    double miu, sigma, X;
    Distrib.DistribNormalS NS = new Distrib.DistribNormalS();
} ///;

```

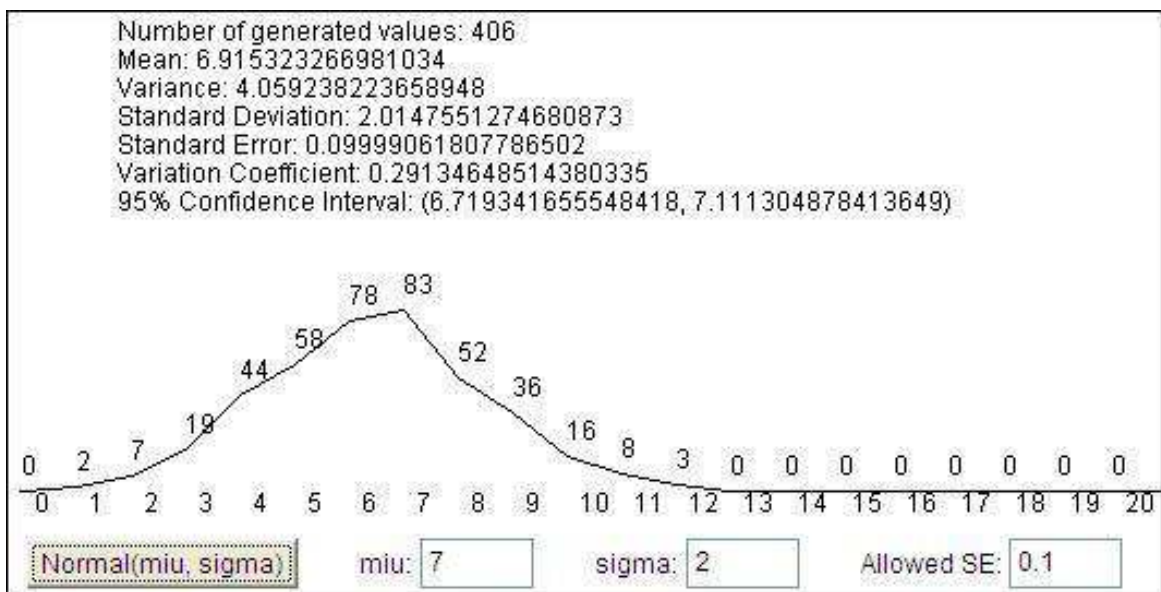


Fig. 12. Simulation from a normal distribution.

Exponential distribution

```

package Distrib;
public class DistribExponential extends DistributionContinue {
    public DistribExponential(double lambda) { // Constructor
        this.lambda = lambda;
    }
    public double simValue() { // Simulation of a value
        U = Math.random();
        X = -1/lambda*Math.log(U);
        return X;
    } // simValue()
    double lambda, X, U;
} ///;

```

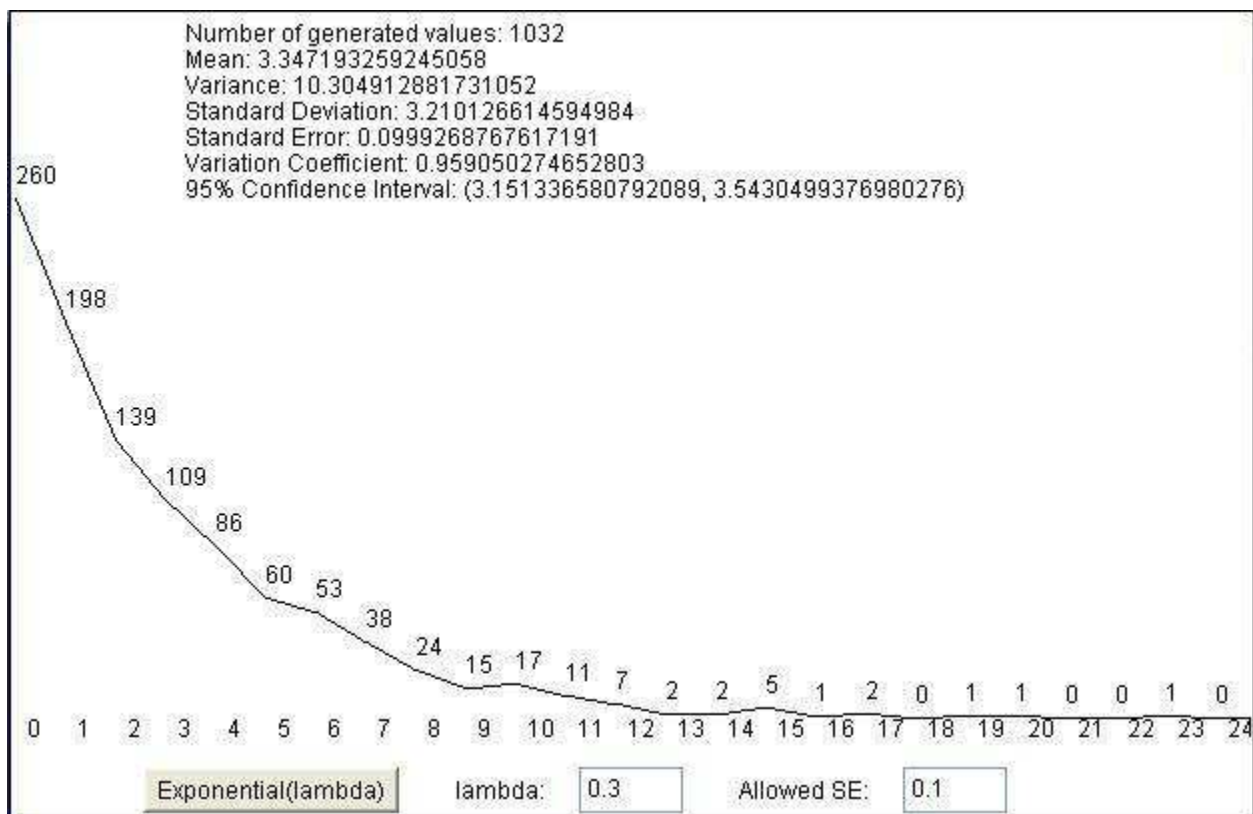


Fig. 13. Simulation from an exponential distribution.

Gamma distribution

```

package Distrib;
public class DistribGamma extends DistributionContinue {
    public DistribGamma(double lambda, int n) { // Constructor
        this.lambda = lambda;
        this.n = n;
    }
    public double simValue() { // Simulation of a value
        double U, X=0;
        for (int k=1; k<=n; k++) {
            U = Math.random();
            X = X-Math.log(U)/lambda;
        }
        return X;
    } // simValue()
    double lambda;
    int n;
} ///;

```

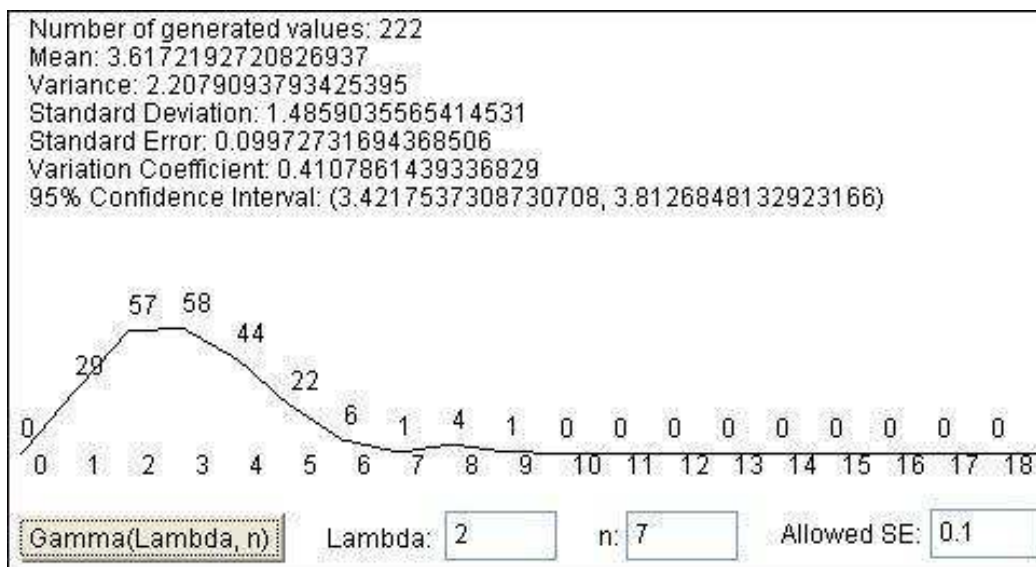


Fig. 14. Simulation from a gamma distribution.

Weibull distribution

```

package Distrib;
public class DistribWeibull extends DistributionContinue {
    public DistribWeibull(double alfa, double beta) { // Constructor
        this.alfa = alfa;
        this.beta = beta;
    }
    public double simValue() { // Simulation of a value
        double U;
        U = Math.random();
        X = 1/alfa*Math.pow(-Math.log(U), 1/beta);
        return X;
    } // simValue()
    double alfa, beta, X;
} ///;

```

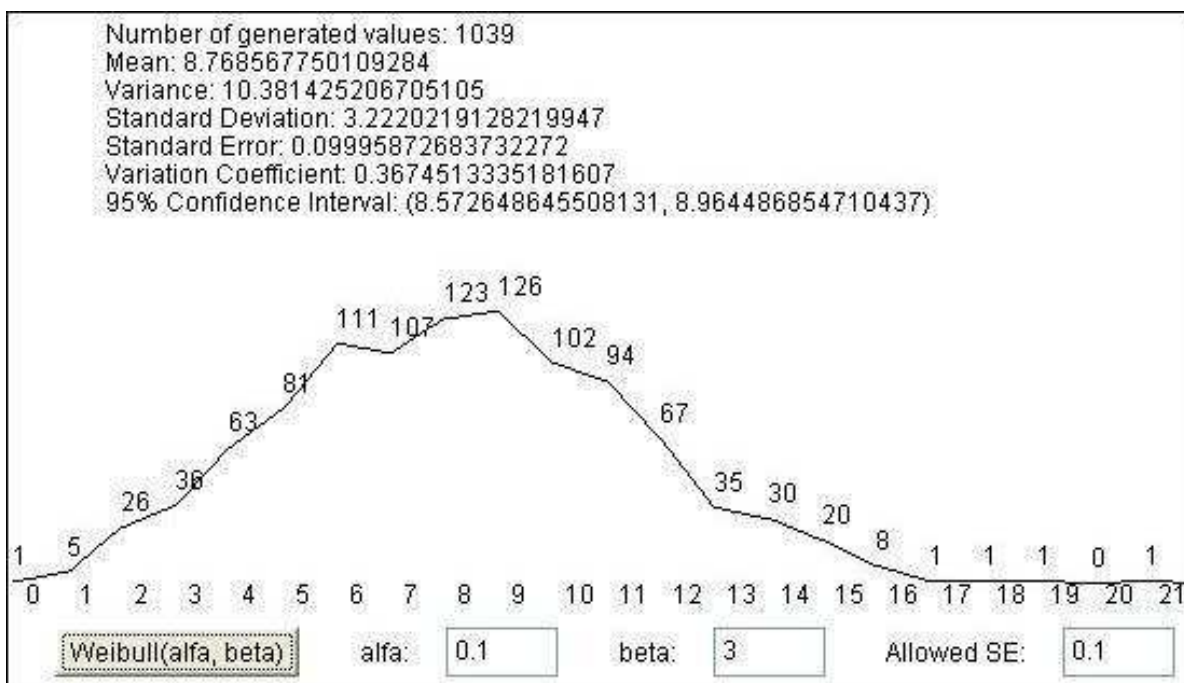


Fig. 15. Simulation from a Weibull distribution.

4.2 Experiments with e-learning scenarios

Based on the simulation system presented previously (Figure 2), we create and implement various models as applications, to be used as e-learning scenarios. We create e-learning scenarios 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, pharmaceutical and health care problems. In addition, we 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.

4.2.1 E-learning scenarios for distributional models

As an example, we propose an e-learning scenario for students in medicine and health care. 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 needs to get immediately an explanatory text about binomial distribution $B(n, p)$. Then it is possible 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 $k \leq 11$, one 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, but 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 $k \in \{0, 1, 2, \dots, 11\}$, in a very suggestive column format, and to recognize the solution to previous problem shown with dashed columns (Figure 16).

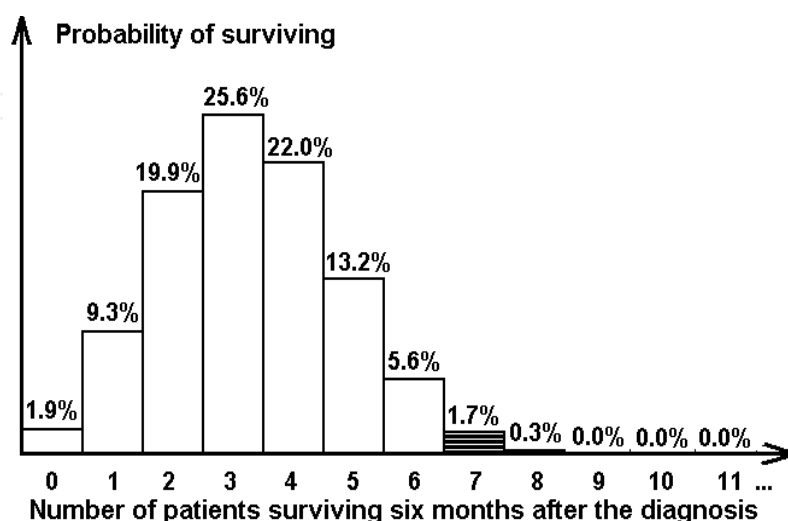


Fig. 16. A graphical solution for the patient surviving problem, based on the binomial distribution $B(11, 0.3)$.

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.

4.2.2 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 $N = 995$ 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 (Figure 17). There is an obvious relation between theoretical values and the values presented in Figure 16, as in the previous scenario we used the binomial model $B(11, 0.3)$.

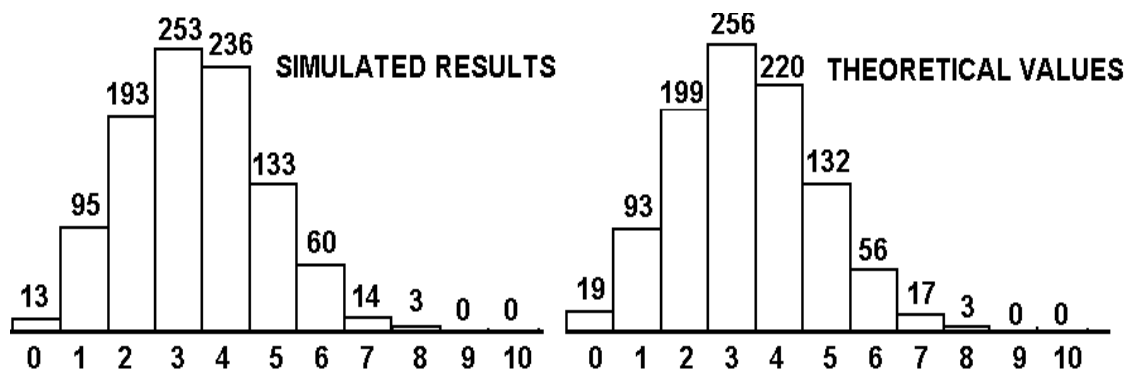


Fig. 17. Simulated versus theoretical values in case of the binomial distribution $B(11, 0.3)$.

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, i.e., 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 $\mu = E(X_i)$, $i=0, 1, 2, \dots$, the learner should continue to generate new data until are generated n data values for which the estimate of the standard error, $SE = s / \sqrt{n}$, is less than an acceptable value given previously 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 18 shows a comparison of the results for two simulations from binomial distribution $B(11, 0.3)$, first simulation with allowed standard error of 0.1, and second simulation with allowed standard error of 0.05.

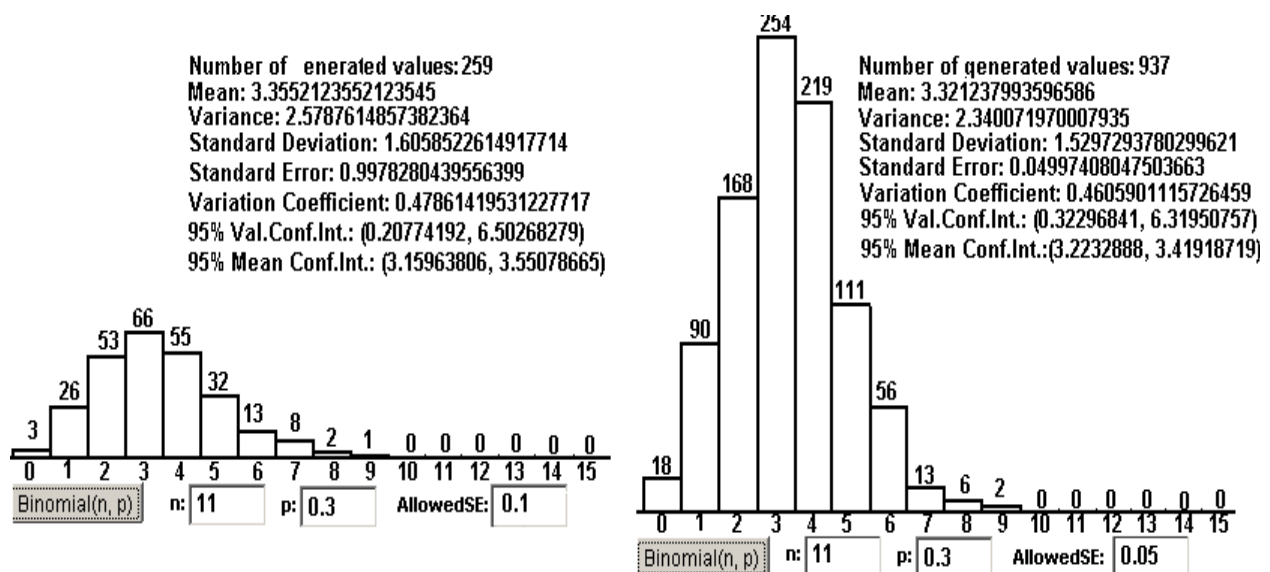


Fig. 18. A simulation with *Allowed SE* = 0.1, versus a simulation with *Allowed SE* = 0.05, in case of binomial distribution $B(11, 0.3)$

The number of generated values is determined by the value *Allowed SE* introduced by the learner, as the simulation stops when the condition $s / \sqrt{n} \leq \text{Allowed SE}$ is true. For the left side simulation, with allowed standard error of 0.1, are generated 259 values, while for the right side simulation, with allowed standard error of 0.05, are 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 after the decimal point. 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 $\text{int}(x)$, 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 neighbouring 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 19 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 19, graph a), and the second with a precision of two decimal digits (Figure 19, graph b).

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 20 shows a similar comparison for the exponential distribution with parameter $\lambda = 0.3$.

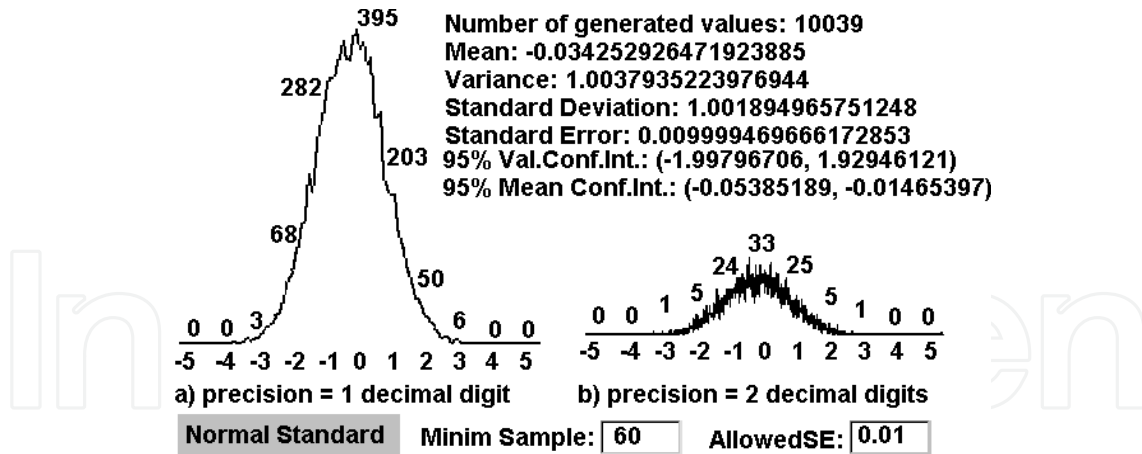


Fig. 19. The same set of generated values from the Standard Normal distribution $N(0, 1)$, visualized with a precision of one decimal digit, versus a precision of two decimal digits.

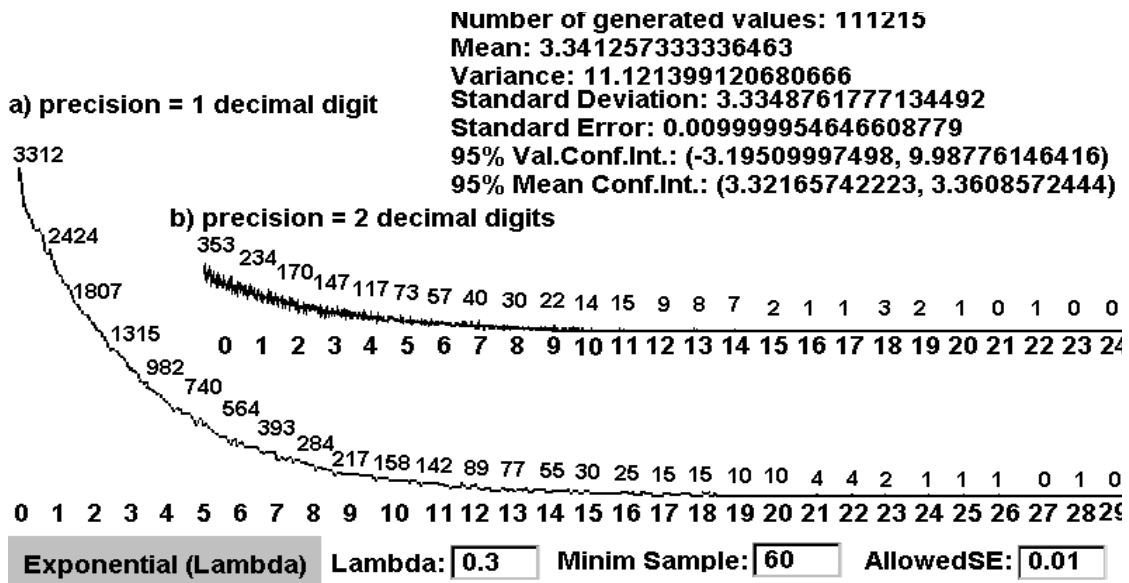


Fig. 20. The same set of generated values from the exponential distribution $E(0.3)$, visualized with a precision of one decimal digit, versus a precision of two decimal digits.

4.2.3 A Markov model for geriatric patients' behaviour

We modelled the flow of patients through chronic diseases departments (Gorunescu et al., 2002) and based on these models we created e-learning scenarios. The planning of medical service within a chronic healthcare department is a complex problem the staffs 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 modelled the patient flow through chronic diseases departments. The use of stochastic compartmental analysis, which assumes probabilistic behaviour of the patients around the system, is considered a more realistic representation of an actual situation rather

than simpler deterministic model. 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 modelled as a Poisson process with a parameter estimated by using the inter-arrival times (Prodan et al., 1999). These times are independent exponential random variables, each with parameter λ (the rate of arrivals) and with the corresponding density function $f(t) = \lambda e^{-\lambda t}$, $t \geq 0$. Figure 21 shows a simulation of the Poisson arrivals at rate $\lambda = 2.75$ per day.

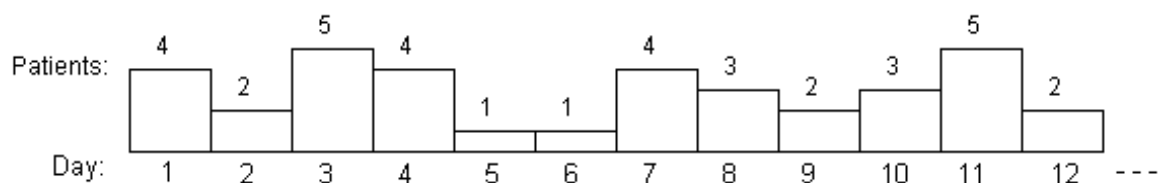


Fig. 21. The Poisson arrivals at rate $\lambda = 2.75$ per day.

The in-patient care time is modelled as a mixed-exponential phase-type 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 $f(x) = \rho \lambda_1 e^{-\lambda_1 x} + (1-\rho) \lambda_2 e^{-\lambda_2 x}$, which implies a mean care time of $\rho/\lambda_1 + (1-\rho)/\lambda_2$ days per patient (Prodan et al., 2000). Figure 22 shows a simulation for the in-patient care time with parameters $\rho = 0.93$, $\lambda_1 = 0.04$ and $\lambda_2 = 0.001$, resulting a mean care time of 93.25 days per patient.

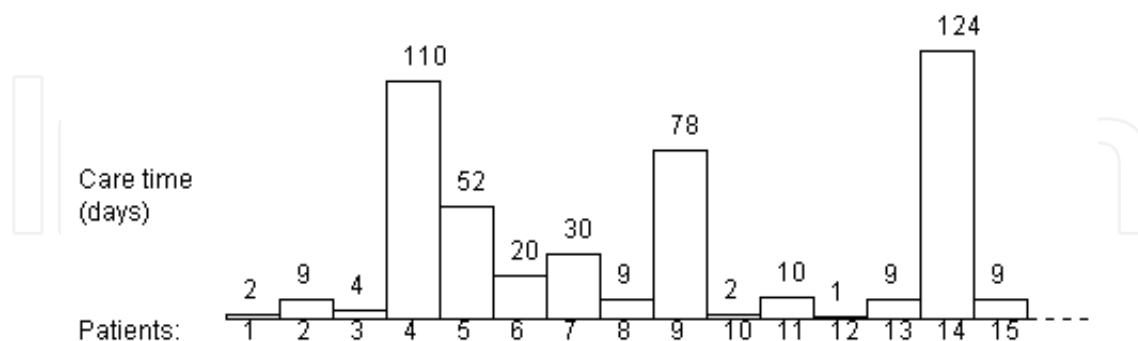


Fig. 22. The simulated results for in-patient care time.

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. We used this model in training health care teams, based on e-learning technologies.

4.2.4 E-learning tools for wound image understanding

Medical images are valuable in both didactic and research activities, for students in medicine, pharmacy and health care. Computerized image processing contains methods for non-invasive wound evaluation, allowing an accurate diagnosis in a large category of patients with damaged and wounded skin. Generally, wounds have a non-uniform mixture of yellow slough, red granulation tissue and black necrotic tissue. Relying on a high quality of image acquisition, we can analyze a succession in time of more images for the same wound and assess changes in wound healing, i.e. the recovery or worse evolution. Our aim is to create and implement in Java an automatic method which can be used as a reference standard for colour and texture wound analysis. The purpose is to create e-learning scenarios for wound image understanding and wound healing simulation, by applying this method to large amounts of wound image data stored in XML based knowledge bases (Figure 23).

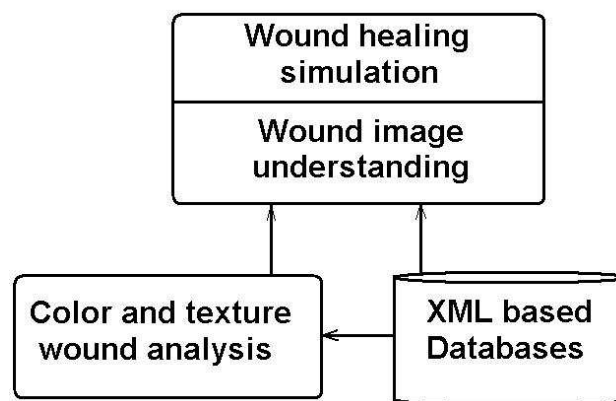


Fig. 23. The general method applied for wound image understanding and wound healing simulation.

Our main objective is to develop appropriate skills in wound management for a learner that traverses such an e-learning scenario. The e-learning scenarios are practice driven and relevant to professional practice, being used by students in medicine, pharmacy and health care, at graduate, postgraduate and residency levels.

For a given wound, we must find out some quantitative and qualitative attributes for assessing the healing state. As quantitative attributes we measure its surface area and its volume (evaluating depth). The original image is processed with the purpose to emphasize the distinction between wound and non-wound area. We use some general methods to enhance the image, because we must exaggerate the distinction between wound and non-wound. As an example, for individuals with fair skin, we lighten the images and then view them using shades of green with the red and blue minimized. This way more clearly exhibit the borders of the wound than in the original image. Removal of the red and blue leaves the wound black and the rest of the image green. For images of individuals with dark skin, both the red and green are accentuated while the blue is minimized. This procedure also leaves the non-wound area green, but colours the wound red. In either case, the wound can easily be distinguished from the non-wound without difficulty.

Our work is based on a continuous collaboration with physicians and wound care experts, because it is necessary to make a rigorous classification for various categories of wounds. We collected large amounts of wound image data and we calculate statistical parameters as mean, median, standard deviation, confidence interval, skewness and kurtosis for them. These historical data are included in XML based databases, to be used as inputs to classification algorithms. The general purpose is to make distinction between infected and non-infected, inflamed and non-inflamed wounds. Based on colour analysis, we build a statistically significant differentiation of mild, moderate and severe wounds. Our system analyses the differences in calibrated hue between injured and non-injured skin, obtaining a repeatable differentiation of wound severity for various time intervals. As an example, burn wounds are characterized according to their depth as:

- Superficial - with bright red colour and the presence of blisters (usually with brown colour);
- Deep - with red-whitish colour and with dark dots;
- Full thickness - with creamy or dark brown colour.

Assuming normality, the first two moments (mean and standard deviation) characterize very well the colour distribution. However, sometimes the first two moments alone are inadequate to discriminate between wound and non-wound skin. Therefore further details of the colour distribution are required. Skewness and kurtosis of the colour data proved to be more useful for this purpose.

We implement e-tools that will enable to assess the current state of the wound and to gain insight into the wound evolution, by comparing the series of wound data collected over time. Based on this knowledge we can design an e-tool for simulating the process of wound healing. The colour image processing is the most acceptable automatic method of objectively and reproducibly analyzing skin wounds and lesions. Also, this method has the main advantage of being completely non-invasive.

5. Conclusions and future work

The main aim of our research is to acquire better results in education based on improving the teaching-learning relation by means of e-learning technologies. We think that an important contribution to a high quality teaching-learning relation have the charm and eloquence of a good teacher. At the same time, it is our belief that e-learning technologies can improve the teaching-learning relations. We may try, but it is very difficult to extend the virtue and charm of a good teacher beyond the face-to-face relation and to include these qualities in e-learning scenarios. The use of Internet as the base line infrastructure for communication is essential, but is not enough for our purpose. It is necessary to transfer the spirit of a good teacher in e-learning scenarios. In our e-learning system, 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 intelligent and practical e-learning tools (Prodan et al., 2006, 2008, 2010).

As future work, we have to create new e-learning scenarios and to find new methods for improving the teaching-learning relation by means of e-learning tools. The best way to follow for improving the teaching-learning relation is by using electronic portfolios and by

incorporating the e-learning tools in a Moodle based e-learning system. Both the teachers and the learners agreed that electronic portfolios improve very much the teaching-learning relations and provide a better assessment among students than traditional methods (Rusu & Prodan, 2006). Also, our Moodle based learning management system provides dynamic and flexible student-centred authentic e-learning experiences, improving this way the teaching-learning relations.

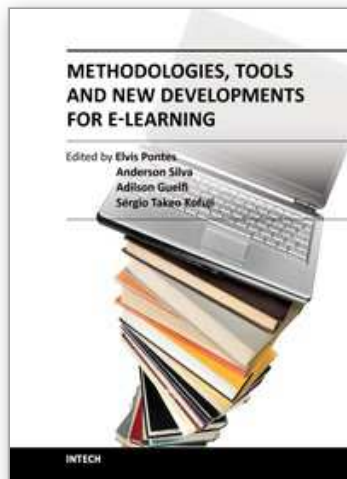
6. Acknowledgment

This chapter was supported by UEFISCU PNII Partnership grant 11018/2007-2010 (MoDef).

7. References

- Barrett, H. C. (2001). Electronic Portfolios. A chapter in *Educational Technology, ABC CLIO*. Available from: <http://electronicportfolios.com/portfolios/encyclopediaentry.htm>
- Farr, R. (1990). Setting directions for language arts portfolios. *Educational Leadership*, Vol.48, No.3, pp. 103
- Francis, R. & Raftery, J. (2005). Blended Learning Landscapes. *Brookes eJournal of Learning and Teaching*, Vol.3, No.1, ISSN 1744-7747, Available from http://bejlt.brookes.ac.uk/vol1/volume1issue3/perspective/francis_raftery.html
- Garrison, D. R. & Kanuka, H. (2004). Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education*, Vol.7, pp. 95-105
- 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*, Vol. 9, No. 1, pp. 185-195
- Kanuka, H. (2008). *The Theory and Practice of Online Learning (Second Edition)*, edited by Terry Anderson, AU Press, ISBN 978-1-897425-08-4. Available from <http://www.scribd.com/doc/51302878/4/Heather-Kanuka>
- Leake, D. B. (1996). *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, AAAI Press/Mit Press, ISBN 0-262-62110-X, Menlo Park, USA
- Martinez, M. & Jagannathan, S. (2008). Moodle: A Low-Cost Solution for Successful e-Learning, *DevLearn, Learning Solutions Magazine*, November 10. Available from <http://www.learningsolutionsmag.com/articles/71/moodle-a-low-cost-solution-for-successful-e-learning>
- Nwana, H. S. (1996). Software Agents: An Overview. *Knowledge Engineering Review*, Vol. 11, No. 3, 1996, pp. 205-244
- Prodan, A. (1996). Quelques barrières à franchir dans la relation enseignement-apprentissage, In: *Flash Informatique*, EPFL, spécial été, 3 septembre 1996, pp. 23-26, Available from: <http://ditwww.epfl.ch/SIC/SA/publications/FI96/fi-sp-96/sp-96-page23.html>
- Prodan, A. & Prodan, M. (1997). *Java Environment for Internet*. ProMedia Plus, Cluj-Napoca, ISBN 973-9275-07-9
- Prodan, A. (1998). Towards a new type of open learning. *Clujul Medical*, Vol. LXXI, No. 3, pp. 424-431, ISSN 1224-550X

- Prodan, A., Gorunescu, F., and Prodan, R. (1999). Simulating and Modelling in Java, *Workshop POOSC'99 (Parallel/High-Performance Object-Oriented Scientific Computing)*, Lisbon, TR FZJ-ZAM-IB-9906, Forschungszentrum Julich GmbH, pp. 55-64
- Prodan, A., Gorunescu, F., Prodan, R. and Câmpean, R. (2000). A Java Framework for Stochastic Modelling, *Proceedings 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. & Prodan, Rodica (2001). Stochastic Simulation and Modelling, *Proceedings ETK-NTTS'2001 (Exchange of Technology and Know-how - New Techniques and Technologies for Statistics)*, Crete, pp. 461-466
- Prodan, A. & Prodan, R. (2002). A collection of Java class libraries for stochastic modelling and simulation, *Proceedings ICCS'2002 (International Conference on Computational Science)*, Amsterdam, Springer, Sloot et al.(Eds.), Part. I, LNCS 2329, pp. 1040-1048
- Prodan, A.; Câmpean, R. & Rusu, M. (2004). Stochastic Modelling and Simulation in Medicine and Pharmacy, *Proceedings ECMI 2004 (European Conference on Mathematics for Industry)*, pp. 242-247 Eindhoven, Netherlands, 21-25 June 2004
- Prodan, A. & Câmpean, R. (2005). Bootstrapping methods applied for simulating laboratory works, *Campus-Wide Information Systems*, Vol. 22, No. 3, pp. 168-175, Emerald, ISSN 1065-0741, ISBN 1-84544-399-X
- Prodan, A.; Rusu, M.; Câmpean, R. & Prodan, R. (2006). A Java framework for analyzing and processing wound images for medical education, *Proceedings of the ECMS 2006 (European Conference on Modelling and Simulation)*, pp. 421-426, ISBN 0-9553018-0-7, 28-31 May 2006, Bonn, Germany
- Prodan, A.; Rusu M.; Câmpean,R. & Prodan, R. (2008). Artificial intelligence for wound image understanding, *Proceedings of the ICEIS 2008 (International Conference on Enterprise Information Systems)*, pp. 213-218, ISBN 978-989-8111-48-7, 12-16 June 2008, Barcelona, Spain
- Prodan, A.; Rusu, M.; Revnic, C.; Câmpean, R.; Mitrea, P. & Prodan, R. (2010). Intelligent e-Tools for Wound Image Understanding and Evaluation, *Proceedings of the IEEE/ICDS 2010 International Conference on Digital Society (ICDS 2010)*, pp. 8-13, IEEE Catalog Number: CFP10738-CDR, ISBN: 978-0-7695-3953-9, 10-16 February 2010, St. Maarten, Netherlands Antilles. IEEE Computer Society Digital Library, Available from <http://www.computer.org/portal/web/csdl/doi/10.1109/ICDS.2010.11>
- Rusu, M. & Prodan, A. (2006). Electronic portfolios as means of improving the teaching-learning relation, *Proceedings of the ICICTE 2006 (International Conference on Information Communications Technologies in Education)*, pp. 17-21, ISBN: 1-895802-26-8, ISSN: 1109-2084, 6-8 July 2006, Rhodes Island
- Ross, S. M. (1990). *A Course in Simulation*. Macmillan Publishing Company, New York, USA, 2nd edition



Methodologies, Tools and New Developments for E-Learning

Edited by Dr. Elvis Pontes

ISBN 978-953-51-0029-4

Hard cover, 332 pages

Publisher InTech

Published online 03, February, 2012

Published in print edition February, 2012

With the resources provided by communication technologies, E-learning has been employed in multiple universities, as well as in wide range of training centers and schools. This book presents a structured collection of chapters, dealing with the subject and stressing the importance of E-learning. It shows the evolution of E-learning, with discussion about tools, methodologies, improvements and new possibilities for long-distance learning. The book is divided into three sections and their respective chapters refer to three macro areas. The first section of the book covers methodologies and tools applied for E-learning, considering collaborative methodologies and specific environments. The second section is about E-learning assessment, highlighting studies about E-learning features and evaluations for different methodologies. The last section deals with the new developments in E-learning, emphasizing subjects like knowledge building in virtual environments, new proposals for architectures in tutoring systems, and case studies.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Augustin Prodan, Paulina Mitrea, Mădălina Rusu, Cornelia Revnic and Remus Câmpean (2012). E-Learning Tools as Means for Improving the Teaching-Learning Relation, Methodologies, Tools and New Developments for E-Learning, Dr. Elvis Pontes (Ed.), ISBN: 978-953-51-0029-4, InTech, Available from: <http://www.intechopen.com/books/methodologies-tools-and-new-developments-for-e-learning/e-learning-tools-as-means-for-improving-the-teaching-learning-relation>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen