ABOUT THE WAY CAI CAN CHANGE A DIFFICULT TRAINING PROCESS INTO A PLEASANT AND USEFUL EXPERIMENT

Rodica Baciu & Dorin Sima
"Lucian Blaga" University of Sibiu, "Hermann Oberth" Faculty of Engineering
Department of Computer Science & Automatic Control
2400, Sibiu, Str. Emil Cioran, nr.4
E-mail: rodica.baciu@ulbsibiu.ro
E-mail: dorin.sima@ulbsibiu.ro

Abstract: It is a reality the fact that a great number of students even from technical faculties are bored of an instructional material containing a lot of mathematical relations. Taking into account this aspect of training we changed a traditional lecture of "Reliability onComputing Systems" course with a computer-aided instruction. In this paper we present the way we transformed a few pages of mathematical relations from curriculum into attractive computer based application.

Keywords: *interactive tutorial, CAI, on-line training*

I. INTRODUCTION

There are a lot of examples of computer's utility in the training process. The history of computer-aided instruction contains a lot of tools, tutorials, and systems. From a pedagogical point of view the most useful achievements are the ones that actively engage students in the learning process. The presentation of training material in an electronic format, with hyperlinks, does not essentially change the traditional presentation of information in paper format. On the other hand some students prefer reading a paper format course rather than an electronic hyperlink course. But changing or completing a lecture with an interactive tutorial involving the student makes use of the full potential of the new technology.

The theory of constructive learning states that knowledge is actively constructed rather than acquired. The student must be an active learner, as opposed to a passive receiver of information, which is then regurgitated back to an assessor [Dick 2000]. Using an interactive tutorial the student actively takes part in the instructional process. An interactive tutorial provides motivation for students to engage themselves in learning. In this paper we will describe our experiences in changing a traditional lecture into a pedagogically successful computer based instruction. In this paper we will concentrate on technical issues involved in creating digital content for "Reliability on Computing Systems" course at "Lucian Blaga" University of Sibiu. The evaluation of the interactive tutorial is addressed in paper [Baciu 2001]. In section II we elaborated an overview of mathematical aspects of our interactive tutorial. In section III we go on with the description of the application.

II. THE MATHEMATICAL BASE OF THE INTERACTIVE TUTORIAL

The goal of this paper is not at all the presentation of structural models or algebraic methods for computing the reliability of the systems. However we consider important to briefly present the mathematical support forming the base for modeling achieved by the interactive tutorial. We state the reason for this short presentation just by the goal of this paper – to show how a lecture involving assimilating some mathematical methods can become interesting for students.

The methods implemented by the tutorial are algebraic methods being part of the logic model for computing the reliability of the systems [Volovici 1997]. A first such method consists of successive transformations of the structure function of the system until it is expressed by a mutual incompatible reunion of logical products. The transformations applied to the structure function of the system are based on the following relation:

$$S = S_1 \cup S_2 = S_1 \cup \overline{S_1} S_2 \qquad (1)$$

The left side of the above relation expresses that the structure function of a system – *S*, is formed by the reunion of two minimum paths, *S1* and *S2*. The right side express that the same structure function is formed by the reunion of two mutual incompatible terms. For *m* minimum paths, (1) is generalized by:

$$S = \bigcup_{i=1}^{m} S_i = S_1 \cup \overline{S_1} \cdot S_2 \cup \overline{S_1}\,\overline{S_2} \cdot S_3 \cup \ldots \cup \overline{S_1}\,\overline{S_2} \ldots \overline{S_{m-1}} \cdot S_m \qquad (2)$$

The form of the terms {S$_i$ | i=1, 2, ., m} is the following:

$$S_i = x_{i1} \cdot x_{i2} \cdot \ldots \cdot x_{in} \qquad (3)$$

If the minim number of paths does not contain common variables (x_{ij}), the reliability of the system can be directly computed by the next relation:

$$R_s = P(S=1) = \sum_i \prod_{k=1}^{j_i} P(x_{i_k} = 1) \qquad (4)$$

Else, if the minim paths contain common variables, the structure function must be transformed for eliminating common variables. In this way the independence of the terms expressed by logical products is assured. If De Morgan rules are used to eliminate common variables, the mutual incompatible feature of the terms is lost. For common variables eliminating from minim paths only the next equivalents are used:

$$\begin{aligned}
&\overline{xy} \cdot \overline{xz} = (\bar{x} \cup x\bar{y}) \cdot (\bar{x} \cup x\bar{z}) = \bar{x} \cup x\bar{y}\bar{z} \\
&\overline{xy} \cdot xz = (\bar{x} \cup \bar{y})xz = x\bar{y}z \\
&\bar{x} \cdot \overline{xy} = \bar{x} \cdot (\bar{x} \cup x\bar{y}) = \bar{x} \qquad (5) \\
&x \cdot \overline{xy} = x\bar{y}
\end{aligned}$$

The second algebraic method presented by the tutorial is based on transformation of the structure function of the system in an algebraic function by equalizing the logic operations through algebraic operations relied on the next table:

Table 1

	Logic operation	Algebraic operation
AND	xy	xy
OR	x∪y	x+y-xy
NOT	x	1-x
XOR	x⊗y	x+y-2xy

The two algebraic methods are very similar but one is using logical transformations, and the other algebraic operations. Another method presented by the interactive tutorial is the experimental method for roughly evaluating the reliability function of a system. The method consists of generating possible states of all component elements depending on the individual reliability functions and evaluating the performance of the system for every change. In case that the structure function of the system is known, the effective experimentation is not necessary, because the possible states of the elements can be simulated, according to the probabilities associated with binary variables x$_i$, probabilities which are reliability functions of components.

III. THE INTERACTIVE TUTORIAL FOR TRAINING IN COMPUTING THE RELIABILITY OF SYSTEMS

The goal of the interactive tutorial is the understanding and the experimentation of algebraic methods and experimental method for the determination of the reliability of a system on condition that the reliability functions of the component elements are known.

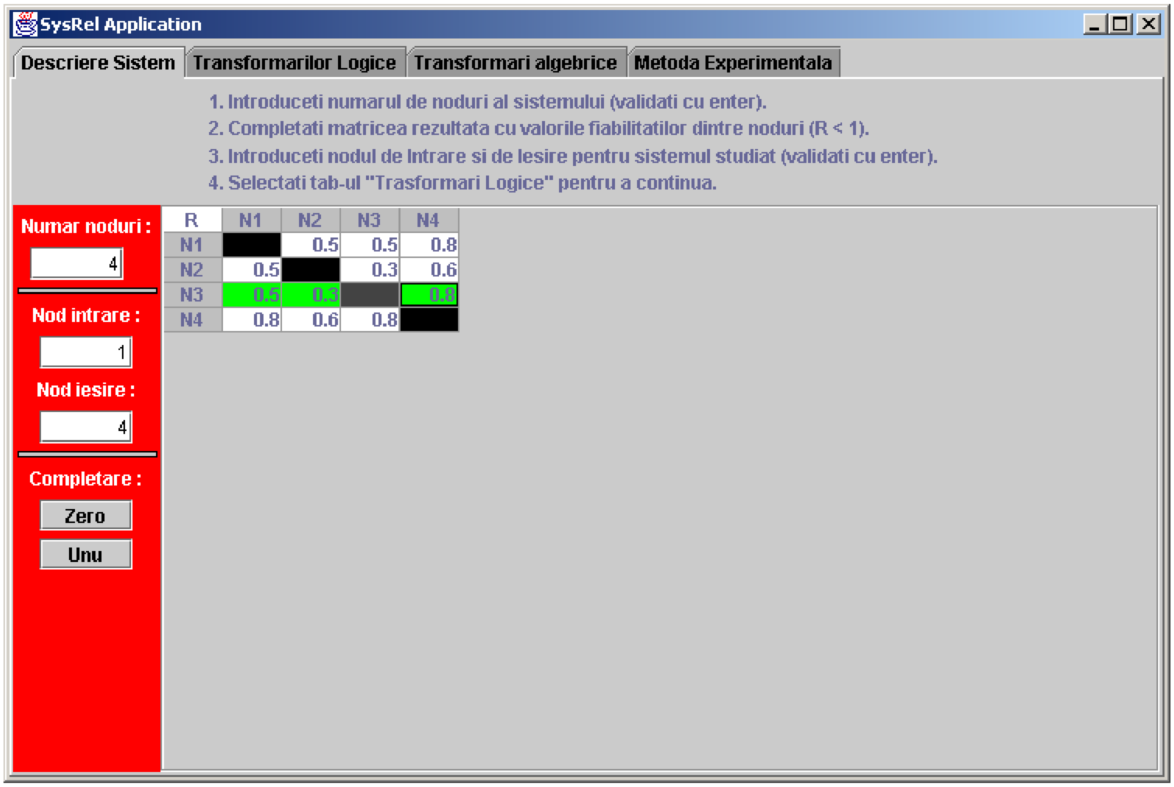
The first step in tutorial is the configuration of the studied system and the setting of the reliability values for component elements of the system. The system configuration is stored in an adjacency matrix. The dimension of the system is established by specifying the number of rows and columns of an adjacency matrix. The user can specify the system configuration and the reliability of the component elements by directly filling in adjacency matrix.

Automatically the program completes with 0 free cells in the matrix, reports for values grater than 1 for the reliability of components (*R$_{ij}$ > 1*) and automatically completes symmetry cells of the adjacency matrix (R$_{ij}$ = R$_{ji}$). Then the user specifies the input and output node of the system (Figure 1). The system configuration ends by displaying the structure function of the system, configured by the user.

Next, the user can choose from one of the three methods for determining the system reliability:

- The algebraic method based on logical transformation;
- The algebraic method based on algebraic transformations;
- The experimental method.

For either of the first two methods, the tutorial will successively display the transformations of the structure function till the final form and the relations associated with every transformation. So the students can see step by step the transformations of the structure function of the system until it is in the form of a reunion of mutual incompatible logical products. After eliminating common variables from the structure function, the reliability function can directly be obtained by replacing variables with suited reliability functions. The program calculates the value of the system's reliability for the set of values of the reliability of the components chosen by the user. For a specific system configuration the user can experiment different values of the reliability of the component elements. Also the user can experiment different configurations for the same set of component elements (the same set of reliability values)(Figure 2 and Figure 3).



1. Introduceti numarul de noduri al sistemului (validati cu enter).
2. Completati matricea rezultata cu valorile fiabilitatilor dintre noduri (R < 1).
3. Introduceti nodul de Intrare si de Iesire pentru sistemul studiat (validati cu enter).
4. Selectati tab-ul "Trasformari Logice" pentru a continua.

Numar noduri : 4

Nod intrare : 1

Nod iesire : 4

Completare :

R	N1	N2	N3	N4
N1		0.5	0.5	0.8
N2	0.5		0.3	0.6
N3	0.5	0.3		0.8
N4	0.8	0.6	0.8	

Figure 1

In the case of the experimental method, the user will specify the number of the simulations. For every component element of the system, the user is asking for the value of the λ parameter. With that end in view, an adjacency matrix for λ parameter setting is displayed by the tutorial. Only cells corresponding with component elements are unlocked. Because the structure function of the system is known the effective experimentation is not necessary – the possible states of the components are simulated. Therefore the program successively generates random sets of values for fault probability of every element of the system. For each simulation the program determines the time after which the system breaks down. Using the default time values the tutorial determines punctual estimation of the reliability function of the system and displays the diagram of the reliability function (Figure 4).

Next we present the algorithm of the experimental method.

1. The values $x_1, x_2, ..., x_k \in [0, 1]$ for fault probabilities of component elements of the system 1, 2,..., N are randomly generated.

2. For every k component element the moment of the fault is computed based on:

$$t_k = \frac{1}{\lambda_k} \cdot \ln \frac{1}{1 - x_k} \qquad (6)$$

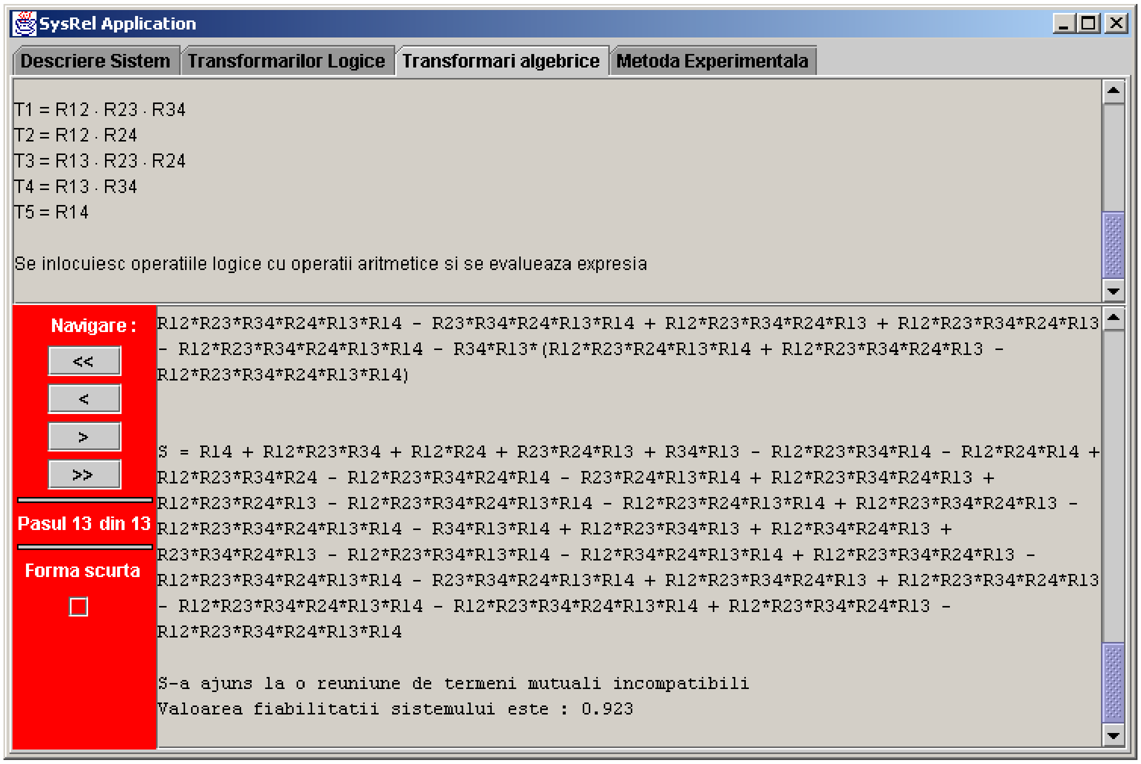
3. The fault moments are ascending sorted: $t_1, t_2, ..., t_n$.

4. The minim t_k is determined and the corresponding k element is setting the first fault element.

5. The k element is extracted from the system configuration by filling with 0 in the adjacency matrix, suitable for its reliability value. Then a path from input to output of the system is looked for.
   If "there is no path from input to output the system"
      Then t_k is also the system fault time in this simulation and go to step 6.
      Else the next time from step 3 list is chosen and go to step 5.

6. The 1 to 5 steps are repeated for s times where s is the number of simulations, stated by the user. A great value for s is recommended.

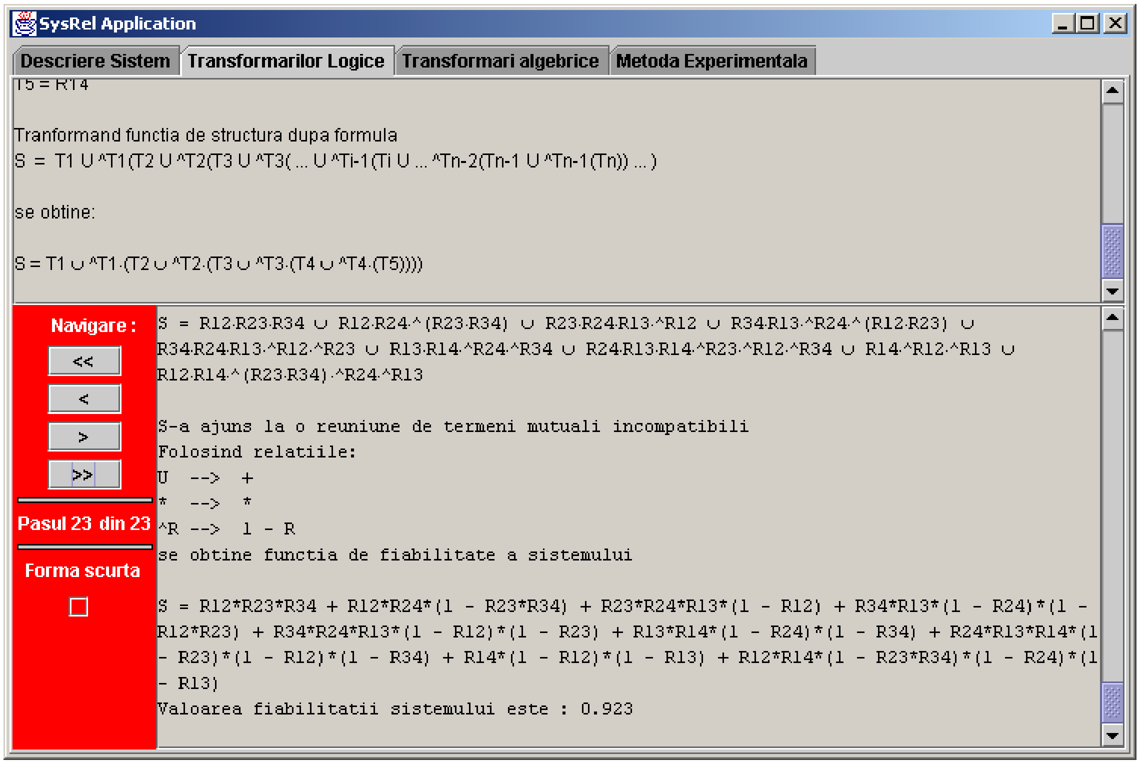
7. The fault moments of the system for each simulation are used to determine the reliability function of the system R_s.



```
T1 = R12 · R23 · R34
T2 = R12 · R24
T3 = R13 · R23 · R24
T4 = R13 · R34
T5 = R14

Se inlocuiesc operatiile logice cu operatii aritmetice si se evalueaza expresia
```

```
R12*R23*R34*R24*R13*R14 - R23*R34*R24*R13*R14 + R12*R23*R34*R24*R13 + R12*R23*R34*R24*R13
- R12*R23*R34*R24*R13*R14 - R34*R13*(R12*R23*R24*R13*R14 + R12*R23*R34*R24*R13 -
R12*R23*R34*R24*R13*R14)


S = R14 + R12*R23*R34 + R12*R24 + R23*R24*R13 + R34*R13 - R12*R23*R34*R14 - R12*R24*R14 +
R12*R23*R34*R24 - R12*R23*R34*R24*R14 - R23*R24*R13*R14 + R12*R23*R34*R24*R13 +
R12*R23*R24*R13 - R12*R23*R34*R24*R13*R14 - R12*R23*R24*R13*R14 + R12*R23*R34*R24*R13 -
R12*R23*R34*R24*R13*R14 - R34*R13*R14 + R12*R23*R34*R13 + R12*R34*R24*R13 +
R23*R34*R24*R13 - R12*R23*R34*R13*R14 - R12*R34*R24*R13*R14 + R12*R23*R34*R24*R13 -
R12*R23*R34*R24*R13*R14 - R23*R34*R24*R13*R14 + R12*R23*R34*R24*R13 + R12*R23*R34*R24*R13
- R12*R23*R34*R24*R13*R14 - R12*R23*R34*R24*R13*R14 + R12*R23*R34*R24*R13 -
R12*R23*R34*R24*R13*R14

S-a ajuns la o reuniune de termeni mutuali incompatibili
Valoarea fiabilitatii sistemului este : 0.923
```

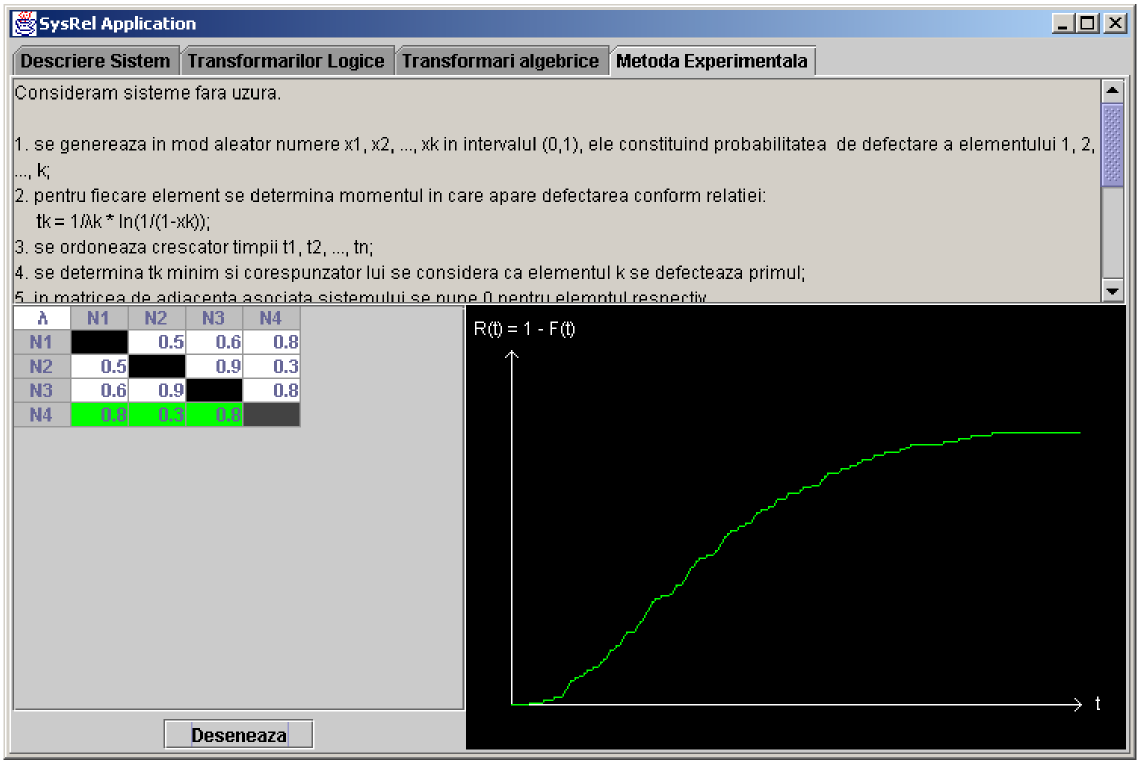
Figure 2



```
T5 = R14

Tranformand functia de structura dupa formula
S  =  T1 U ^T1(T2 U ^T2(T3 U ^T3( ... U ^Ti-1(Ti U ... ^Tn-2(Tn-1 U ^Tn-1(Tn)) ... )

se obtine:

S = T1 ∪ ^T1·(T2 ∪ ^T2·(T3 ∪ ^T3·(T4 ∪ ^T4·(T5))))
```

```
S = R12·R23·R34 ∪ R12·R24·^(R23·R34) ∪ R23·R24·R13·^R12 ∪ R34·R13·^R24·^(R12·R23) ∪
R34·R24·R13·^R12·^R23 ∪ R13·R14·^R24·^R34 ∪ R24·R13·R14·^R23·^R12·^R34 ∪ R14·^R12·^R13 ∪
R12·R14·^(R23·R34)·^R24·^R13

S-a ajuns la o reuniune de termeni mutuali incompatibili
Folosind relatiile:
U  -->  +
*  -->  *
^R -->  1 - R
se obtine functia de fiabilitate a sistemului

S = R12*R23*R34 + R12*R24*(1 - R23*R34) + R23*R24*R13*(1 - R12) + R34*R13*(1 - R24)*(1 -
R12*R23) + R34*R24*R13*(1 - R12)*(1 - R23) + R13*R14*(1 - R24)*(1 - R34) + R24*R13*R14*(1
- R23)*(1 - R12)*(1 - R34) + R14*(1 - R12)*(1 - R13) + R12*R14*(1 - R23*R34)*(1 - R24)*(1
- R13)
Valoarea fiabilitatii sistemului este : 0.923
```

Figure 3

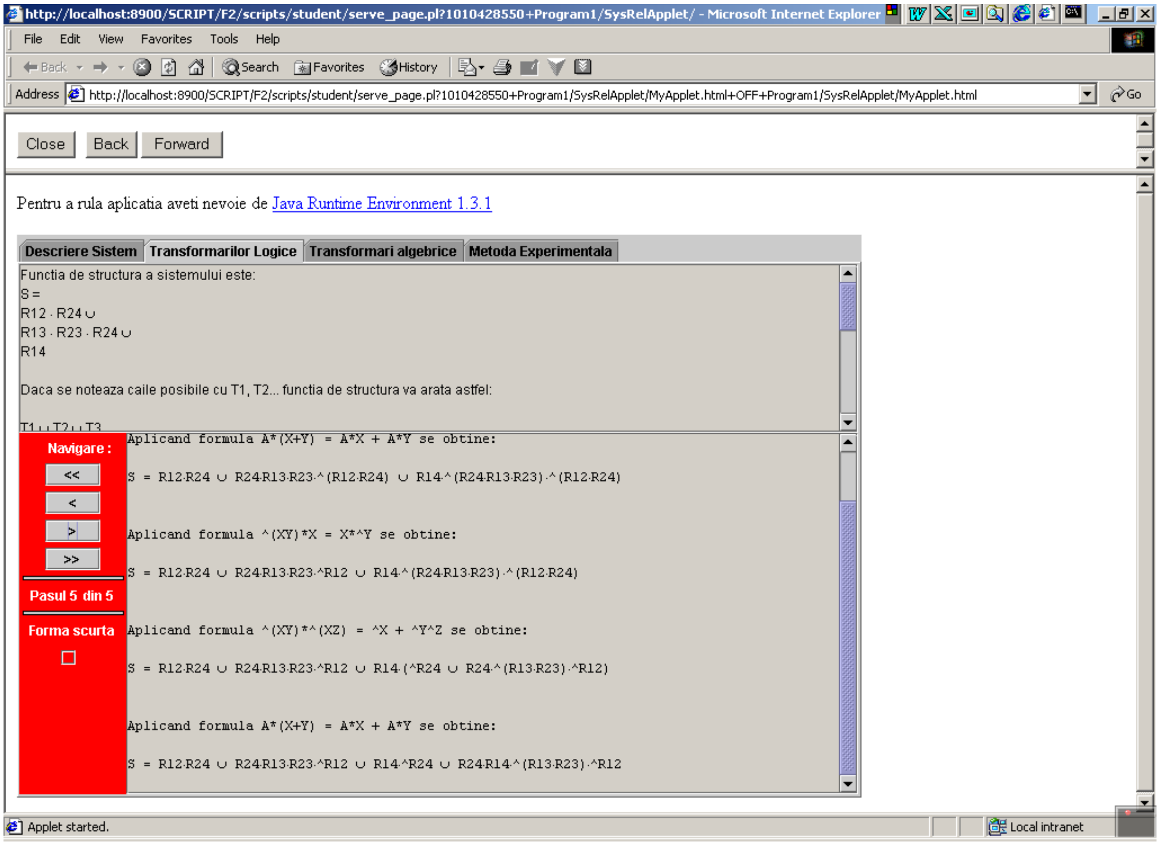


Consideram sisteme fara uzura.

1. se genereaza in mod aleator numere x1, x2, ..., xk in intervalul (0,1), ele constituind probabilitatea de defectare a elementului 1, 2, ..., k;
2. pentru fiecare element se determina momentul in care apare defectarea conform relatiei:
   tk = 1/λk * ln(1/(1-xk));
3. se ordoneaza crescator timpii t1, t2, ..., tn;
4. se determina tk minim si corespunzator lui se considera ca elementul k se defecteaza primul;
5. in matricea de adiacenta asociata sistemului se pune 0 pentru elementul respectiv

λ	N1	N2	N3	N4
N1		0.5	0.6	0.8
N2	0.5		0.9	0.3
N3	0.6	0.9		0.8
N4	0.8	0.3	0.8	

Figure 4


http://localhost:8900/SCRIPT/F2/scripts/student/serve_page.pl?1010428550+Program1/SysRelApplet/ - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
History
Address
http://localhost:8900/SCRIPT/F2/scripts/student/serve_page.pl?1010428550+Program1/SysRelApplet/MyApplet.html+OFF+Program1/SysRelApplet/MyApplet.html
Go
Close
Back
Forward
Descriere Sistem
Transformarilor Logice
Transformari algebrice
Metoda Experimentala
Navigare :
<<
<
>
>>
Pasul 5 din 5
Forma scurta
□
Applet started.
Local intranet

Pentru a rula aplicatia aveti nevoie de Java Runtime Environment 1.3.1

Functia de structura a sistemului este:
S =
R12 · R24 ∪
R13 · R23 · R24 ∪
R14

Daca se noteaza caile posibile cu T1, T2... functia de structura va arata astfel:

T1 ∪ T2 ∪ T3

```
Aplicand formula A*(X+Y) = A*X + A*Y se obtine:

S = R12·R24 ∪ R24·R13·R23·^(R12·R24) ∪ R14·^(R24·R13·R23)·^(R12·R24)


Aplicand formula ^(XY)*X = X*^Y se obtine:

S = R12·R24 ∪ R24·R13·R23·^R12 ∪ R14·^(R24·R13·R23)·^(R12·R24)


Aplicand formula ^(XY)*^(XZ) = ^X + ^Y^Z se obtine:

S = R12·R24 ∪ R24·R13·R23·^R12 ∪ R14·(^R24 ∪ R24·^(R13·R23)·^R12)


Aplicand formula A*(X+Y) = A*X + A*Y se obtine:

S = R12·R24 ∪ R24·R13·R23·^R12 ∪ R14·^R24 ∪ R24·R14·^(R13·R23)·^R12
```

Figure 5

We elected to use Java as our implementation language. The tutorial is available in two versions:

- A version running off-line after downloading from on-line course.
- A version running on-line. The condition is that the client computer runs "Java Runtime Environment 1.3.1" software. This version has the advantage that it can run in any Web browser and therefore can be delivered via web browser.

The figure 5 is a screen capture of the application running from Internet Explorer browser.

A first achievement of the tutorial is the demonstration that methods presented in the lecture are algorithmic methods. The tutorial is useful to the students both for training in computing the reliability of a system and for effective computing of the reliability function of a system with known configuration and known reliability values for component elements of the system. To understand the three methods, the program displays the mathematical relation from the base of the transformation for every transformation in part. The students can experiment computing methods for the reliability of a system, can compare implemented methods. The implementation of the application in Java and running from any browser is a support for including it in an on-line course.

IV. Conclusions and future work

Programming techniques necessary in implementation of this tutorial are not very easy (the Polish form of mathematical expression, graphs). In this way the implementation is time eater. Taking into account this aspect it can be said that such an approach is not justified only for an iteration of a lecture. For a use of a tutorial year after year, the effort due to programming is justified.

The interactive tutorial was well received by the students, their appreciation was positive. We consider the tutorial an exciting experimentation for the enrichment of teaching with more such applications. "The Reliability of Computing Systems" subject is teaching senior students, when they have advanced techniques of programming. Taking this aspect into consideration, programming such tutorials can constitute subjects for annual assignments. The best projects of the students could be integrated in the next years as teaching tools.

V. Getting the application

Readers interested in the tutorial should contact the author by e-mail.

VI. References

[1] Baciu R., Ştețiu G., 2001, Interactive program of training in the "Reliability of computing systems" discipline, New Horizons in Industry and Education, Milos Island, Greece, pages 101-106.

[2] Dick, M., Postema, M., Miller, J., 2000, Teaching Tools for Software Engineering Education, In Proceedings of ITiCSE 2000 (July 2000), Helsinki, Finland, pages 49-52.

[3] Volovici, D., 1997, Fiabilitatea hardware şi software, Sibiu, Editura Universității "Lucian Blaga".