PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 44, NO. 2, PP. 121–140 (2000)

TOOLS AND TECHNIQUES IN SIMULATION OF HIGHLY COMPLEX, DYNAMIC SYSTEMS

József DOMBI* and Levente SÁRA**

Department of Applied Informatics
University of Szeged
H–6720 Szeged, Árpád tér 2, Hungary
e-mail: *dombi@inf.u-szeged.hu **lsara@cc.u-szeged.hu

Received: February 1, 2001

Abstract

This paper will tell us some considerations on solution tools for highly complex problems usually unsolvable by classic exact tools. We are all surrounded by such complex problems even in our every-day life. Usually we should not deal with them in the merit, as we can manage them by intuitive steps and reactions. Other models cannot be handled so easily. There is a great need for qualitative simulating tools and techniques. A rural development system, product development processes, a university management or the solid waste system in New York cannot be investigated without sophisticated simulation tools. The subject of this paper is to develop tools for models, which consist of several parallel processes. Tools and techniques, that will be introduced in the following, have the goal to help the analysis of dynamic systems. Researchers need interactivity to see the behaviour of a system. Another issue is reliability. Double run estimations present a solution for this problem. All these features and more others are included in our simulation tool. The simplicity of experimenting has been utilized and new results have been reached in the models mentioned. Further steps have been taken to develop a well-usable interface.

Keywords: simulation, complex system, vizualization.

1. Introduction

'Which way should we choose to reach our destination the fastest?' This is the question when going on holiday. Although this could be a real complex problem, we quite rarely use highly sophisticated route optimization programs to solve this problem. Sometimes, however, alternatives are not so evident for us like roads on a map. Quite a large number of models have been developed all around the world for various purposes [8, 24]. In these models there exists a great number of effects and counter-effects. Several models have been created for studying such complex systems, like insurance management models [1], market forecasting [23], MASHAYEKI's budgeting study [25] or production systems like in [14] by FORD and STERMAN. Change of some values in the system indicates unpredictable side-effects and feedbacks. These effects cannot be managed by classic mathematical tools and methods, because they are highly non-linear and other problems can arise [12]. With the introduction of delayed conditional behaviour the system may become chaotic. While simulations are only inaccurate, mathematics has no tool to deal with chaos.

First we will see the reasons for the use of simulations instead of strict, exact equation systems. We get an overall picture of problems and goals when studying system dynamics in complex models and the most important features of dynamic system analysis will be shown. In the second part we answer these questions through some examples, taken from BONTKES's [4] and GALBRAITH's [17] studies. The particular new results in a certain examined model can be used as starting points for further research.

1.1. Increasing Complexity

Models can be described mathematically and the equations are solvable by well-known techniques. Most common models have a connection-chain with static type of inherencies. In real life one can see considerable difference in the behaviour of a system in two separate moments with the same base values. This is because of the changes in the inside relations of the system. It would lead us to perceive and apply the concept of system dynamics. Modelling a dynamic system means a new level of complexity. It expands the dimensions our modelling tool has to manage. This new problem is introduced as feed-back, which means chains of embedded consequences ending in a circle of chain-reactions. Mathematics still has tools to manage this mechanism. Differential equation systems are extensively investigated in literature. A real complex model, however, could be solved only with great difficulties, especially if relations contain delays. These delays make the most simple model very difficult. However, this addition is not a rare demand at all.

Theoretical models are still good at the question of control and regulation. Feed-back mechanisms in self-controlling, strongly parallel, multilevel complex systems are the real end-points for clear, 'nice' and exact solutions. At this point we are already forced to use noisy data with only approximately good functions and – that is the main point – fuzzy defined concepts, objects or predicates. All these lead us from pure modelling to simulations. Simulations are inaccurate with no clear or mathematically established connection-system. Despite all these simulations have the great advantage that they always produce results.

1.2. Describing and Studying a Model

When using simulations, our first step is to describe the model and the flow of simulation. Usage of description languages is common. There are also time-driven and event-driven simulation methods [18]. Neither definition of the model, nor statements for controlling functions are easy to describe in such a language. Since human thinking uses ill-defined concepts, we always have to translate our intuitions into strict, exact mathematical objects and notions or functions provided by the given simulation description language.

Perception of system dynamics in the life of a simulation leads us to intervene into the events. Simulating tools usually allow us to define various scenarios in order to find out by which way the desired result can be reached in a highly complex dynamic system. It can be stated again that scenario planning as classic method for experiments is far from being a comfortable tool for a researcher.

The most typical feature of highly complex dynamic systems is strong parallelism. Effects indicated by external or internal changes propagate like waves. But these waves seem to run into reflecting objects in the model and come back again. It takes some time from the start till the echos of these waves return. This delay gains absolute importance as the final effect develops from the co-existence of all the altering forces at a given point. It is necessary to see that the actually existing impacts differ from point to point in the model. Simulations use delay to produce this feature.

1.3. Improved Reliability

As it has already been stated simulations do not provide exact, absolutely reliable solutions. That is why a ready-to-use model always has to be tested by comparing simulation data and facts. Despite this fine-tuning of our model we still could not be sure that end-results of different scenarios or interactions are not corrupted by simulation noise [22]. Simulation noise is a consequence of discrete time measurement, even in the case of event-driven simulations. If the advance step of time is greater than the interval of any periodically changing variable in the system, then some effects could be entirely left out from the computations. FORRESTER [15] suggests time delay to be selected as half of this ‘solution interval’. BARTON and TOBIAS [2] introduce a new method by which the error of simulation data can be estimated from two separate runs of simulation where delay times are different. This would take, of course, far more time if we use scenarios.

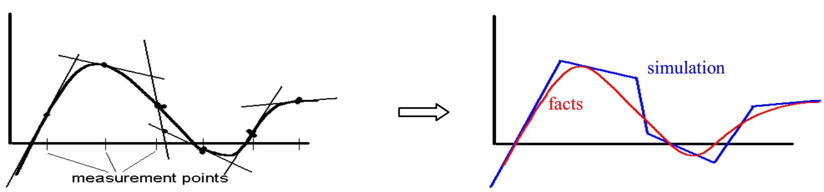


Fig. 1.

2. Our Solutions

A program has been developed to create and develop dynamic models quickly and easily by using visualization tools. During the process of a simulation certain values can be selected to see how they have changed. The program uses various visualization tools to realize the trends or principles which control the actions and 'behaviour' of the whole model. A hyperbolic-like screen view is used to represent the model of a graph [16]. The main advantage is that the user is not forced to use a specific simulation description language to build the model, so developing a new model is extremely fast. Data (the nodes of the graph) and the relations between the nodes can be created by an easy-to-use interface. Our own functions can be defined to describe the nodes using even DOMBI's special fuzzy functions [11] to grab human concepts. These functions use DOMBI-operators, namely $c(x, y)$ and $d(x, y)$ for conjunction and disjunction,

$$c(x, y) = \frac{1}{1 + \left[\left(\frac{1-x}{x} \right)^{\alpha} + \left(\frac{1-y}{y} \right)^{\alpha} \right]^{1/\alpha}},$$

$$d(x, y) = \frac{1}{1 + \left[\left(\frac{1-x}{x} \right)^{-\alpha} + \left(\frac{1-y}{y} \right)^{-\alpha} \right]^{1/-\alpha}},$$

where α is a free parameter. These operators are more efficient than many others like min-max operators.

Unlike min-max operators DOMBI-operators can be continuously differentiable at any point. Special fuzzy functions are more flexible and that is why they are more effective in simulations than min-man functions. E.g. the result of a $\min(x, y)$ function is independent of the value of y, when $y \geq x$. Multi-valued logical operators are extremely efficient [5], while WILLIAM's [32] and CHENG's [6] considerations can be easily taken into account. The program automatically creates and manages the most often used functions such as average, weighted average, sum, product, etc. Weight and speed of the effects are represented by the relation with assigned functions. We deal with the problem of result reliability [22]. The program helps to analyze the simulation results by visualization tools and offers the possibility to increase the accuracy interpreting BARTON and TOBIAS's idea [3] to adjust time parameter. By visualization you can see the speed and direction of processes in diagrams, graphs and colours instead of endless columns of numbers. Some examples will be introduced, including BONTKES's rural development model [4] and GALBRAITH's university management [17], which present the main features of our simulation tool.

2.1. Model Description by Graphs

Graphs are used in our system to describe the structure of a model. Graphs are as close to the human thinking as possible in applied mathematics [16]. The focused objects in a system, that is the human concepts can be interpreted easily as the nodes of the graph. Furthermore, we can derive the basic relations and the associated functions from the known, the observed behaviour of the target system. This makes up our model immediately without using any SDL language. Hyperbolic browser means here an extra possibility to draw structural conclusions without simulation runs. The model can be overviewed easily in this way, while we are not forced to roam in the labyrinth of thousands of objects. This browser stretches our graph display onto a hyperbolic plane. We are standing at the origin point $(0, 0)$ and the hyperboloid is projected on the plane of the screen. Although we can always see all the nodes, the major part of the screen is only occupied by the selected node and its closest neighbours. This browser is a structure-visualization, which shows the user only those points of the graphs, which are in focus. So detection of the relations is much easier as compared to a plane-display.

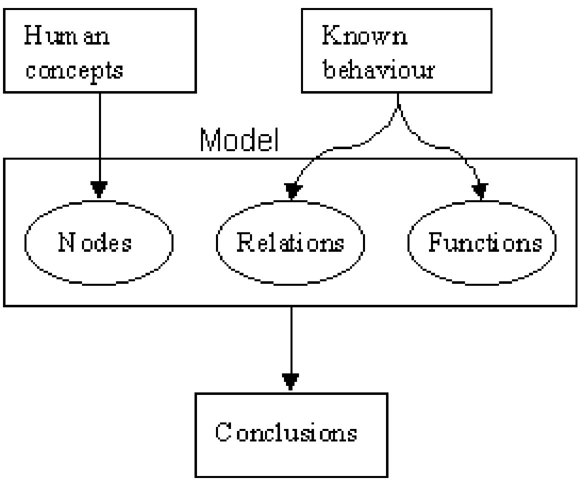


Fig. 2.

Having set up the skeleton of a model by graphs, the sufficient relations between the main objects of the model have already been created without doing any extra effort. Our program can accept human words, intuitions by using the latest advances in multivalued logical research [5, 11].

2.2. Studying the Model

When we run the simulation, we expect some talkative results from which new conclusions can be drawn. Most simulations provide numbers for the user for further processing. A great advantage can be taken of the characteristics of graph display here. We have utilized this, so the flow of the simulation now can be followed by smooth changes of colors still displayed in the graph. This visualization method always catches one's attention and turns it to the most interesting points of the model.

Experiments usually follow the way of scenario-planning, which means multiple simulation runs with many branching points and conditional value assignment. We provide a very simple way for the user to try out his ideas immediately avoiding scenario-planning and without restarting the whole simulation. These interactions can be performed at any discrete point of the simulation time. Simulation tool developed by us provides an included time-slicing in order to automate this error-correction. This method can also be used to verify the stability of our model.

Real models need delays. Interpreting models by graphs results in occurring delays as parameters of edges. Our simulation tool offers the user to specify the weight and delay for all the effects that could occur in a simulation. You can set the length of the edges as well as the strength of those in your model. Through this point of view concepts in the graph theory can also be interpreted. E.g. capacity of a graph can be translated as the limits of possible configurations and speed required to reach a desired state. Here opens an unexplored field to connect theory and practice and to make easier for mathematics to approach human thinking.

2.3. Summary of Features of our Simulation Tool

Let us summarize what is expected from a simulation tool:

- First we want to create models easily. This is presented by our program as we can use our natural concepts as nodes and edges of the graph.
- Second the model should be clear and easily seen. Using the program's hyperbolic browser it becomes a cake-walk.
- Third we want is to follow the process of simulation during run. Visualization techniques help us to do it easily.
- As fourth flexibility is an evident requirement. Modification by graph management is involved in our program.
- The fifth thing usually demanded is to verify the results. We want either to use exact computations or just to ensure that discreteness of time would not corrupt the data. The program supports the latter by functions and time slicing methods.

- Six we want is to model strongly parallel systems with great importance of different delays in effects. For that purpose we can use again the advantages of graph representation.
- The last thing needed is interactivity. As already mentioned, the program enables users to change any parameter in the simulation at any moment even during a given run.

Table 1.

Requirement	Our solution
1. Easy creation	1. Use of nodes and edges
2. Overviewable display	2. Hyperbolic browser
3. Follow the simulation	3. Visualization techniques
4. Flexibility	4. Graph management
5. Result verification	5. Time-slicing
6. Importance of delays	6. Parametric edges
7. Experiments	7. Interactive access points

3. Examples

3.1. Rabbits and Foxes

In the second part of this paper we will introduce some examples without striving for completeness. We take these examples in order to show how our simulation tool works. At first we will see how models are created. The models are restricted in some respects so that we could express the main features of our simulation tool. First we take a very simple model of population dynamics. It is very popular in teaching differential equations, called 'Predator-prey model'. In this model we have a number of rabbits and some foxes. The foxes are hunting for rabbits. Hunters kill both rabbits and foxes. Rabbits multiply as far as they do not reach the limits of the area. Foxes need also rabbits for multiplication. Results of the simulations is very sensitive for starting configuration and parameters described below. Although it is usually referred as a fairly difficult model to solve, we introduce it now as our simplest example. Since differential equations provide an exact solution set, this model would show us how to check and verify the data presented by simulation.

Firstly let us see how we can build up our model. The program presents the graphical graph-board mentioned in the foregoing. With simple clicks we add new objects as nodes to the model. Meanwhile relations are automatically created. We can set the name, the shape and the start value of a node and delay, weight and parameter of a relation easily.

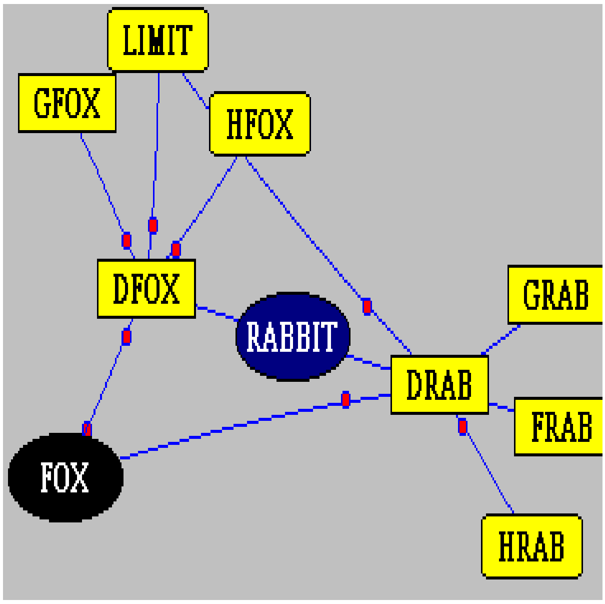


Fig. 3.

Further relations can be added by another click. The program is capable to create the most known functions (such as mean, weighted mean, DOMBI-operators etc.) using the given structure of the model and we can specify our own function types. At last nodes are assigned to their function prototypes and parameters are stated. We also want the model to be overviewed easily.

The second feature we can show by this example is the advantage of hyperbolic browser. In the center of the final model of rabbits and foxes we can see the most important objects in the model, while other parts are smaller. Hyperbolic browser provides the feature that we always see only the parts we are interested in.

Now let us start the simulation! From exact mathematical solutions we know that this model should converge to a fix point at the given parameters we used. As model structure is well- known, here we only state the differential equations and their solutions.

$$x' = a \cdot x(M - x) - x(c + b \cdot y),$$
$$y' = e \cdot x \cdot y(M - y) - d \cdot y,$$

where x, y is the number of rabbits and foxes. Parameters 'a' and 'e' indicate the growth of the populations, 'c' and 'd' are the hunting for them, 'b' is the interrelation between the two populations and 'M' is a territorial limit.

We used the following values:

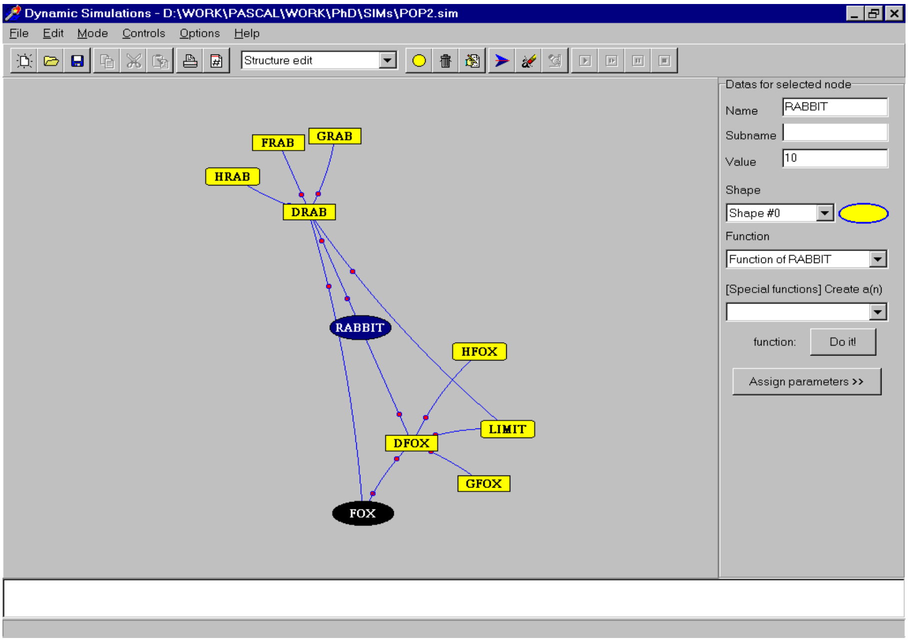


Fig. 4. The structure of the model can be created by some clicks; relations are already shaping...

growth of rabbits	GRAB	$a = 0.5‰$
growth of foxes	GFOX	$e = 0.02‰$
hunting for rabbits	HRAB	$c = 3\%$
hunting for foxes	HFOX	$d = 4\%$
killings by foxes on rabbits	FRAB	$b = 1‰$
area limit	LIMIT	$M = 200$ units

DRAB and DFOX are two variables that store the changes in the number of rabbits and foxes. These variables collect the above parameters and incoming values (RABBIT/FOX). DRAB and DFOX represent x' and y'.

If we take extreme number of rabbits or foxes, we have two trivial fix points. When there are no foxes it means that rabbits were extinct earlier or they reach a stable level at $x = 140$.

Equation systems tell us that the number of rabbits is basically influenced by two factors. A positive force is their own number until it exceeds a limit. On the other hand, foxes and hunting are reducing the number of rabbits. The same two kinds of effects have impact on foxes. In a close inherence they are multiplied by their own number and rabbits' number, while being killed by hunters. The solution of such an equation system shows that there is a non-trivial fix point at the values of parameters we used. Namely number of rabbits tends to about 14,57 meaning

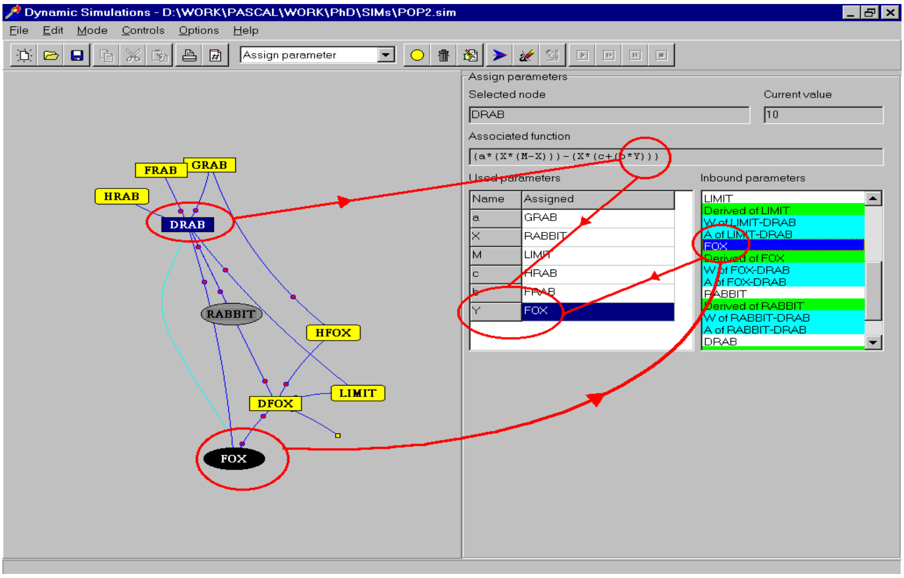


Fig. 5. Parameter assignment: the program presents possible inbound parameters for the user to set the values of the parameters in symbolic functions, that associated with the nodes of the graph

1457 rabbits and foxes to approximately 63 pieces.

$$y = \frac{M - \frac{c}{a} - x}{\frac{b}{a}} = \frac{M\left(x - \frac{d}{eM}\right)}{x},$$

$$\Downarrow$$

$$x = \frac{1}{2}\left[M\left(1 - \frac{b}{a}\right) - \frac{c}{a} + \sqrt{\left(M\left(1 - \frac{b}{a}\right) - \frac{c}{a}\right)^2 + 4\frac{bd}{ae}}\right],$$

$$\Downarrow$$

$$x \approx 14.5683 \text{ and } y = 62.7158.$$

And now let us see the results provided in the first simulation run.

Despite of this fact our simulation fails to prove that. Rabbits become extinct in period 11, and as a consequence the same happens to foxes just 2 periods later. Why? In a real complex system we cannot compare the results with exact solutions.

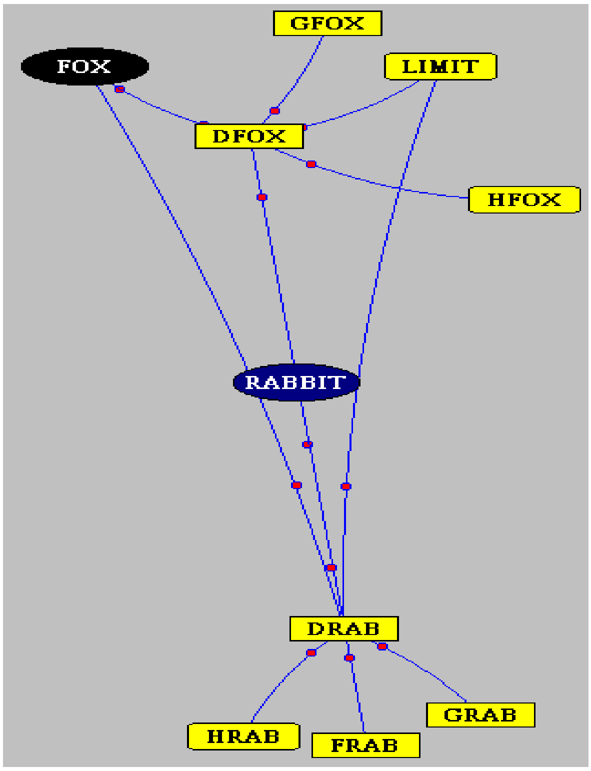


Fig. 6. Hyperbolic browser presents outside objects smaller

So now we should find another way to verify our results. The reason of the failure is the discrete time measurement. In period 8 there are so many rabbits that the population of foxes suddenly starts to grow at a very big rate. Consequence: many foxes kill a large number of rabbits. If this situation lasted for a relatively long time, then rabbits would become extinct. If...! But it is not the case. In reality this tendency would slow down somewhere between period 9 and 11 and then stop. So our time steps are too long at present. Divide these intervals by four and we can grab the moment when these 'extinction' is slowing down. Here we can use the involved time-slicing to refine our results.

Let us try to reduce the length of intervals, refine discrete time steps. This can be carried out with the time-slicing method included in the program. Our second trial is successful, variables converge to their fix points. This way we can reach the right results and BARTON and TOBIAS's new procedure [2] can be applied to estimate more exact values.

Selection of slicing factor is a really difficult question. It always depends on the derivation features of our model functions.

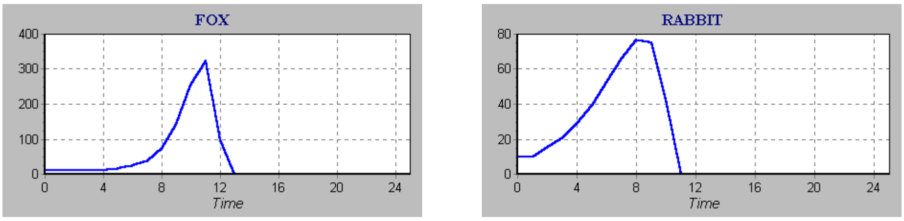


Fig. 7. Simulation results without time-slicing

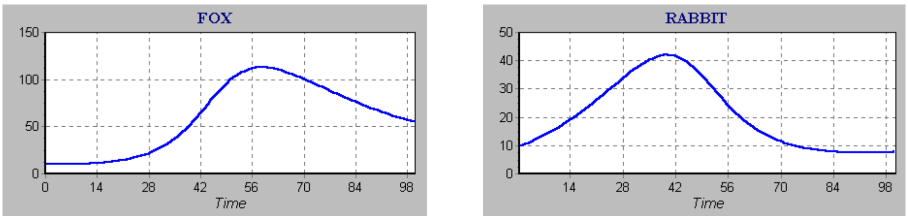


Fig. 8. Simulation result with refined intervals. Slicing factor is about 4

3.2. Funding System of High Education

Let us see a bit more complex mode to show how we can follow the changes in a simulation. This example describes the behaviour of the funding system of educations. This model was introduced first by Peter L. GALBRAITH [17]. He lives in Australia and described the situation in high education systems of Australia, England and other western countries. Universities are forced to sell themselves, plan their budget and operate like typical profit-oriented companies. This could be reached by allocating funds by the number of students and the effectiveness of research produced by faculties. In this system, unfortunately, the weight of research is determined at 15%. This situation has led to the elimination of such research which is fund-depending and could not produce enough publications, rewards and degrees per $, even if it employed more students than others. Furthermore as willingness to apply for admission to universities follows the demands only with a great delay, this has caused a dangerous fluctuation in the number of students.

We – following GALBRAITH's suggestions – tried to simulate this using relative loads in respect of student load and research products. Here is the main function used for simulating the fund allocation and an example of relative loading in the research subsystem. The last function is a simplified version derived from the mathematically parameterized one. We can do these simplifications as the indexes and parameters are coming from the structure of the graph. We used similar simplifications all over the model taking advantage of the graph representation.

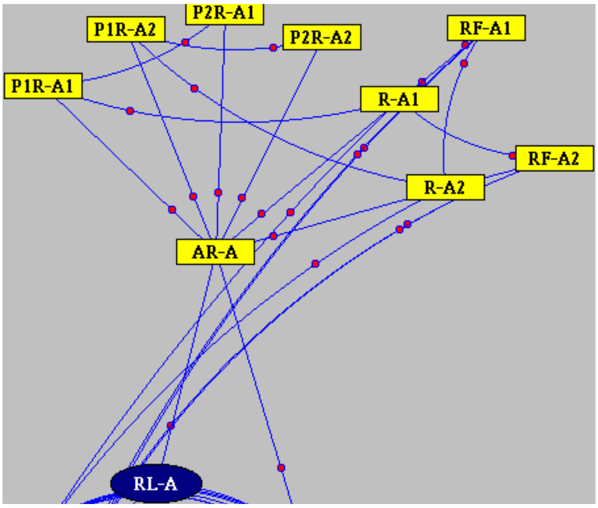


Fig. 9. Part of the research subsystem in model

$$\mathrm{FUND} = [(1 - \mathrm{Res})SL + \mathrm{Res} \cdot RL]G,$$

$$RL_x = \frac{w_{AR_x,RL_x} AR_x(t - d_{RS_x,RL_x})}{TR_x(t - d_{TR_x,RL_x})},$$

$$\Downarrow$$

$$RL = \frac{w \cdot AR}{TR}.$$

Relativity means targeting values around a base-level of 1, e.g. the performance of a university in research compared to the three-year average of all universities' researches. The same can be stated about student loads. This way we have translated these equations into nodes and edges with assigned functions as shown. We have defined the suitable function prototypes by the function editor of the program and assigned them to the appropriate nodes. The required parameters are determined by the program automatically using the known structure of the graph.

A great number of equation systems could be drawn from such a model. Allocated funds for University '*A*' depend on the relative student and research load, the weight of researches and of course the gross budget. This function relies on student and research subsystems. In research subsystem e.g. average research is compared to total researches, while research on a given faculty depends on the gained funds, the main features of the research and the students' number at the university. Here we found a feed-back, connecting at the same time to two different subsystems: the fund allocating and the student recruiting systems. Through these

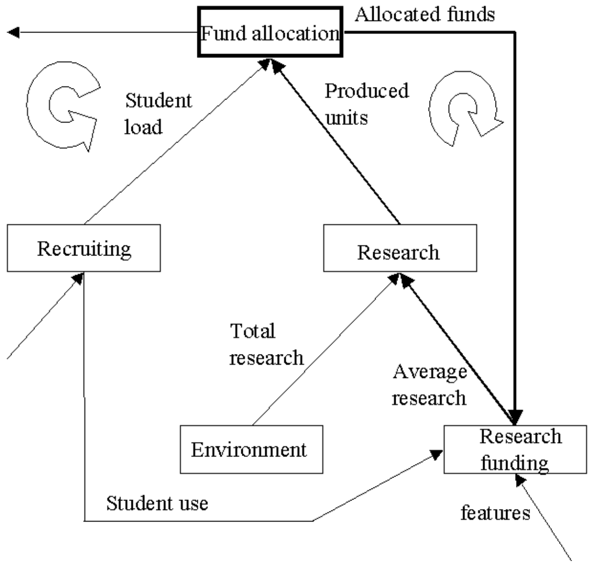


Fig. 10.

feed-backs we return again to our starting point, although we have inspected only one, single route in the structure.

Several subsystems have non-trivial fix points. The entire model, however, contains fluctuating or unsteady systems which yield either trivial extinction of some part of the structure or challenges of critical points in time, when the whole system can collapse. The former means a learning process in which the system tries to eliminate the given subsystem's fatal impacts, while the latter means malfunctioning of an essential subsystem which cannot be tolerated. It can be shown e.g. that such a critical system is student recruiting as it has quite a great effect on all underlying subsystems in the model. Analysis of circles in the graph has proved that life of the main circle depends on stability of the recruiting system.

We simplified GALBRAITH's model [17] significantly for our purposes. In this model we can study other two features of the program. First let us see how visualization can ease monitoring of the simulation run. We have three main objects. In our model there are two universities with four research projects. We will examine unversity '*A*'. Key properties are the student load (signed as SL-A), research load (RL-A) and the funding gap (GAP-A).

When an object has the desired value, then its color is white. The worse is the situation, the darker is the colour. If the value is better than the desired one, the color turns into blue. We can see the waves of the universities up and downs. While running the simulation we always have a picture of the way it reaches its final state. The great power of this feature is evinced in learning this way quite a lot about the life of the model in contrast with the pure end-results. Second we will

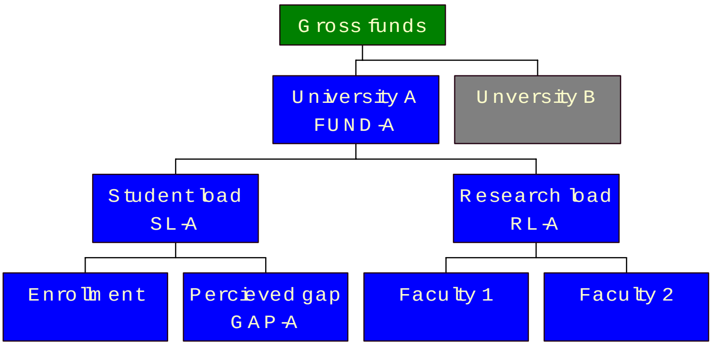


Fig. 11.

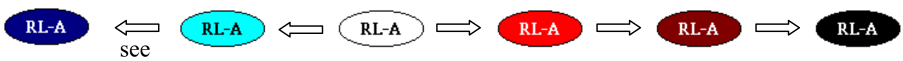


Fig. 12.

see how our simulation tool maintains strict parallelity. The first run has proved that fluctuation is getting worse and worse, meanwhile research is crushed.

Now raise the weight of research in funding up to 50% and see what happens. After a fluctuating period values begin to converge to a fix point. Finally the ebb and flow of students seem to be eliminated. At last we can see that researches, especially the student related ones grow more than other.

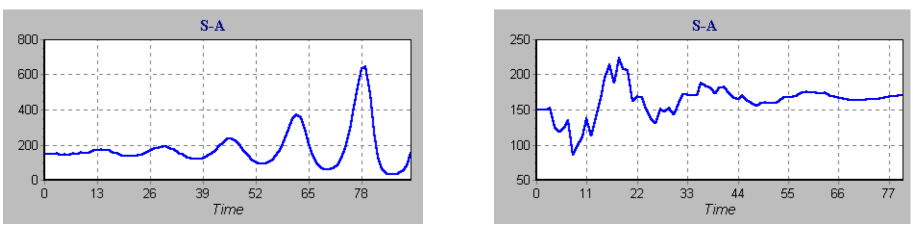


Fig. 13. Number of students at University '*A*', when research weight is at 15% and 50%

This is caused by real parallel feed-backs of research in respect of funding. If research goes up, $ per research unit goes down, so research goes down again. This stabilizes the fluctuation, so funding gaps disappear. Consequently, fatal impacts generated by the instability of the recruiting system are suppressed.

3.3. Socio-Economics

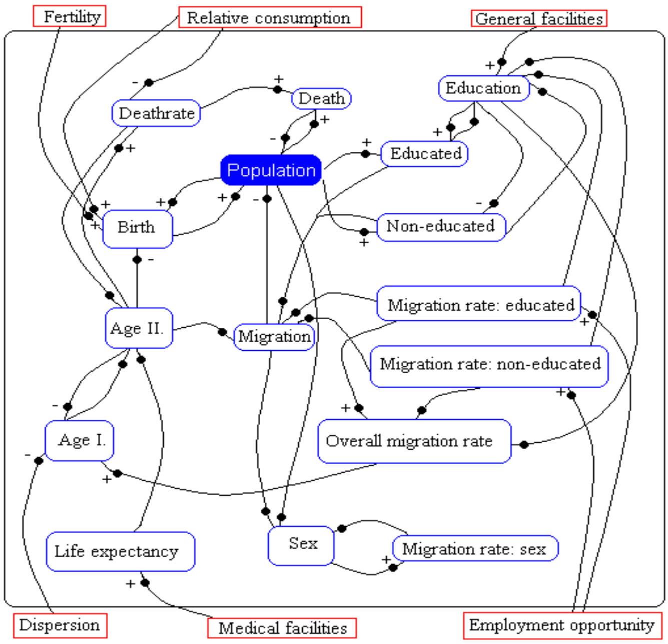


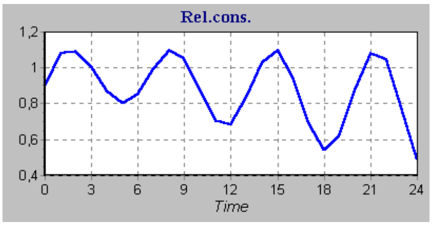
Fig. 14. Main system's outlines

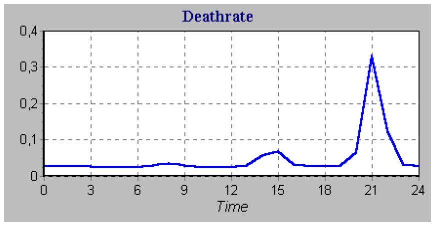
Our last example shows us immediate modifications and user interactions through our simulation tool. This model describes an economic situation in an agricultural country. The first study was issued by Evans-Pritchard, then later dynamical system structure was developed by Struif BONTKES [4]. The structure outlines a really difficult, complex and merely dynamic net of parallel systems. Our main goal is to stop migration of rural population which historically has lead to the depopulation of the area. Main problems are in the fields of death rate [19], balance of sexes and migration rates.

The basic run simulates the historical events [13, 21, 29, 31, 34]. High death rates are caused by alternation of drought and good years [10, 24]. See relative consumptions and death rate. Men have more willingness to migrate [18, 30], this causes problems in the balance of sexes. Finally population has gone.

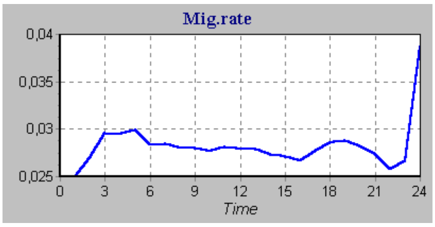
Our first step was an interaction in consumption waves. By several interactions and the introduction of cropping we managed to eliminate the fluctuation of relative consumption [4].

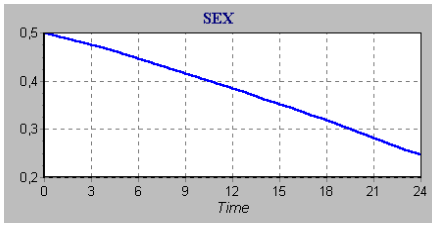
Results begin to be better. Population tries to grow but the sexual imbalance destroys dreams. We performed our interaction at period 1. One can see that there was a positive effect via low death rates, but finally sex balance yields greater





Alternation of relative consumption... ...causes serious epidemics.





People (men!) migrate because of the troubles... ... so women are left alone.

Fig. 15.

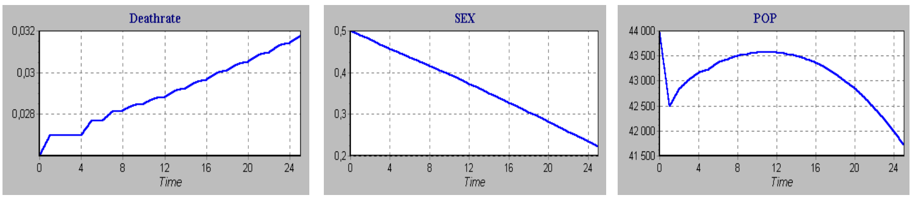


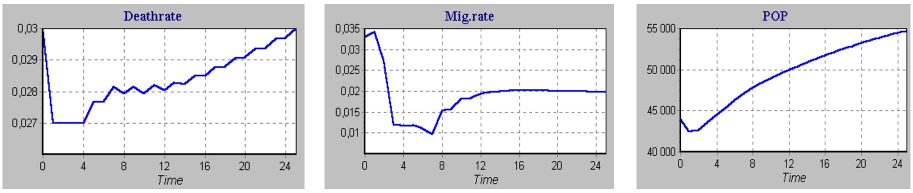
Fig. 16. Death rates cause migration and sexual imbalance. Consequence: population is in danger

negative force on population.

We had six different major interaction types during the experimental phase [14, 27, 28, 33]. Last action performed by us was presenting new opportunities to work in the area. As the main force behind migration is the temptation of the work opportunities in the city, we tried providing workplaces in the rural area. The effect of that was immediate. Population grows again, migration stops, death rate is acceptable.

4. Summary

Dynamic simulation models can be applied in almost every field of life: from the inside processes of an enterprise or modeling of the market grow, through demo-



Death rate stays below 3% Migration is managable Population begins to grow

Fig. 17.

graphic and social changes affecting a whole country until studying the cycle of atmosphere or researching the interactions of ecological situations. E.g. in 1993 BONTKES published his article about the agricultural investments and their effects in South Sudan. The author introduced a dynamic model of the rural life of Sudan [4]. We have chosen it as our last example. Another article talks about the solid waste processing system of New York State [25], where the budget allocation is about to reach a critical point. There are other studies on the inherences of the CO-concentration in the air and the tropical forest-fires using possible dynamic models [26]. There is a great need of easy-to-use simulation tools that provide talkative results and overviewable display. We have developed such a computer tool that fulfils these requirements and by which new conclusions can be drawn. To help the user we use graph implementation [16], so the user is not forced to learn a simulation language. Furthermore various visualization techniques provide new ways to learn more about the life of a model. We introduced three short examples to show how to use these advantages of our simulation tool [17]. The program is capable to manage the problem of discrete time measurement using a time-slicing method. In addition user can freely interact at any moment during the simulation. At last we presented some new conclusions in the examples that can be starting points of new studies.

5. Future Plans

The current function editor of our program is planned to be unified on the base of pliant functions [11]. Multivalued logic will be developed in order to interpret human words [6, 32]. In addition to the present features of our program we are going to develop a learning algorithm, that could help the user to find the best interaction in order to reach the desired goals. This mechanism can be based on neuron nets, global optimization or genetic algorithms. With such additions system refinement can be speeded up and we could find the best solution with greater chance.

References

[1] BARLAS, Y. – CIRAK, K. – DUMAN, E., Dynamic Simulation for Strategic Insurance Management, *System Dynamics Review*, **16**, Spring 2000.
[2] BARTON, P. M. – TOBIAS, A. M., Accurate Estimation of Performance Measures for System Dynamics, *System Dynamics Review*, **14**, Spring 1998.
[3] BARTON, P. M. – TOBIAS, A. M., Use of Adjusted Time Parameters in System Dynamics Models, *System Dynamics Review*, **16**, Spring 2000.
[4] BONTKES, T. E. S., Dynamics of Rural Development in Southern Sudan, *System Dynamics Review*, **9**, Winter 1993.
[5] BUSTINCE, H. – CALDERÓN, M. – MOHEDANO, V., Some Considerations about a Least Squares Model for Fuzzy Rules of Interference, *Fuzzy Sets and Systems*, **97**, 1998.
[6] CHENG, CH.-H., A New Approach for Ranking Fuzzy Numbers by Distance Method, *Fuzzy Sets and Systems*, **95**, 1998.
[7] CORBETT, J., Famine and Household Coping Strategies, *World Development*, **16**, 1988.
[8] COYLE, G., The Nature and Value of Futures Studies or Do Futures Have a Future? *Futures*, **29**, February 1997.
[9] *Delphi 3 – Programmer's Manual*, Borland International, Inc. 1997.
[10] DIRKS, R., Social Responses During Severe Food Shortages and Famine, *Current Anthropology*, **21**, 1980.
[11] DOMBI, J., Fuzzy halmazok operátorainak szerkezete a többtényezős döntések szempontjából, Candidate Thesis, Szeged, 1993.
[12] DORDAN, O., Mathematical Problems Arising in Qualitative Simulation of a Differential Equation, *Artificial Intelligence*, **55**, 1992.
[13] FAO (Food and Agriculture Organization): Population and the Labour Force in Rural Economics, In: *FAO Economic and Rural Development Paper*, Rome, 1986, 1987.
[14] FORD, D. N. – STERMAN, J. D., Dynamic Modelling of Product Development Processes, *System Dynamics Review*, **14**, Spring 1998.
[15] FORRESTER, J. W., *Industrial Dynamics*, MIT Press, Cambridge, MA; Republished by Productivity Press, Portland, OR, 1961.
[16] NORTH, S. C., Applications of Graph Visualization.
[17] GALBRAITH, P. L., System Dynamics and University Management, *System Dynamics Review*, **14**, Spring 1998.
[18] GODFREY, E. M., Economic Variables and Rural-Urban Migration, Some Thoughts on the Todaro Hypothesis, *Journal of Development Studies*, 1973.
[19] GROSSE, E. M., Interrelation between Health and Population, Observations Derived from Field Experiences, *Social Science and Medicine*, 1980.
[20] HEIN, A., *Conjoint Simulation – a Modeling Framework for Combined Performance and Dependability Analysis of Computer Systems*, Erlangen, 1997.
[21] ILACO (International Land Development Consultants): Feasibility Study Pengko Plain In: Interim Report, Pengko Pilot Project, 1980.
[22] KUIPERS, B.J. – CHIU, Ch. – MOLLE, D. T. D. – THROOP, D. R., Higher-order Derivative Constraints in Qualitative Simulation, *Artificial Intelligence*, **51**, 1991.
[23] LYNEIS, J. M., System Dynamics for Market Forecasting and Structural Analysis, *System Dynamics Review*, **16**, Spring 2000.
[24] MACCORMACK, C. P., Primary Healthcare in Sierra Leone, *Social Science and Medicine*, **19**, 1984.
[25] MASHAYEKI, A. N., Transition in the New York Solid Waste System: A Dynamic Analysis, *System Dynamics Review*, **9**, Winter 1993.
[26] NEWELL, R. E. – REICHLE, H. G. Jr. – SEILER, W., A légköri szén-monoxid és a földi tüzek égése, *Tudomány*, **V**, December 1989.
[27] SÁRA, L., Dinamikus szimulációk, Diplomawork, Szeged, 1998.
[28] SPURR, G. B., Physical Activity, Nutritions Status and Physical Work Capacity in Relation to Agricultural Productivity, *Energy Intake and Activity*, New York: Alan R. Liss, 1984.

[29] STOCK, R., The Social Impact of Differential Access to New Health Programs in Northern Nigeria, *Social Impact Analysis and Development Planning in the Third World*, **12**, 1985.
[30] TODARO, M. P., *Economics for a Developing World*, London, Longman, 1977.
[31] UN (United Nations): Age Structure of Mortality in Developing Countries, New York, 1986.
[32] VOXMAN, W., Some Remarks On Distances Between Fuzzy Numbers, *Fuzzy Sets and Systems*, **100**, 1998.
[33] WALSH, J. A. – WARREN, K. S., Selective Primary Healthcare, An Interim Strategy for Disease Control in Developing Countries, *Social Science and Medicine*, **14**, 1980.
[34] WFS (World Fertility Survey): Report of a Joint FAO/WHO/ONU Expert Consultation, *Technical Report Series*, **724**, 1984.
[35] *Windows API Guide*, Borland International, Inc. (1992).