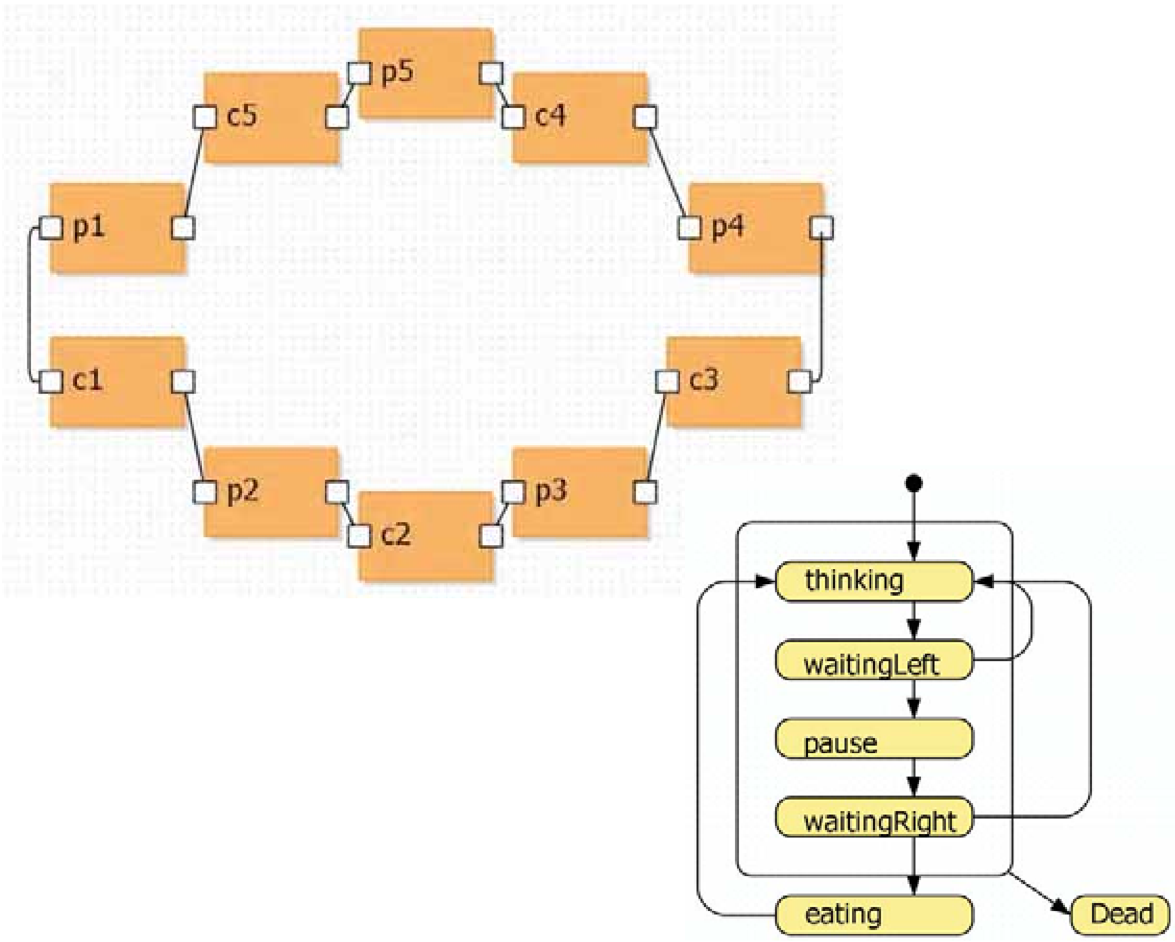
SNE SIMULATION NEWS EUROPE



Volume 18 Number 1

April 2008, ISSN 0929-2268

Journal on Developments and Trends in Modelling and Simulation

Membership Journal for Simulation Societies in EUROSIM

ARGESIM

Dear Readers,

This first SNE issue in the year 2008 introduces two novelties. Up to now, SNE has published in the Notes Section scientific contributions of type Technical Note, Short Note, Software Note and Benchmark Solution. This issue introduces a new type, Educational Notes. Educational Notes should address any kind of modelling and simulation for education, in education, or with education, from teaching and learning impacts via applications in high school and university to e-learning and blended learning, and for/by/with modelling and simulation. This issue starts with a note on discrete event modelling and simulation at high school level (by E. Rybin). We will try to publish in each regular SNE issue one contribution of this type.

The second novelty is a change in the News Section. In the last three years we were faced with the problem that some societies were not active, and did not send in reports. As compromise, we have repeated 'neutral' reports from the foregoing issues. As a better solution, we have developed a section 'Data & Quick Info', which summarizes the relevant basic information of all EUROSIM societies – to appear as first part in the News Section of each regular SNE issue. This Data & Quick Info section is to be followed by specific news from EUROSIM and from EUROSIM societies. As this issue is delayed, it contains only EUROSIM news – reports on EUROSIM Congress 2007, and on the EUROSIM Board Meeting at EUROSIM 2007, with the changes in EUROSIM. News from societies will be published in SNE 18/3-4.

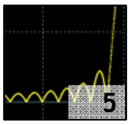
We are proud to publish an interesting contribution by G. Korn, a doyen among the simulationists, aged beyond eighty, an still active in programming and working with Desire (in this issue concentrating on vector features). A Technical Note on computer network simulation, a Short Note on decision strategies in supply chain management, and six benchmark solutions complete the Notes Section. The new layout and the double space for benchmark solutions have proven successful, increasing also the quality of solutions.

We hope, readers enjoy the novelties and the content, and we thank all contributors, members of the editorial boards, and people of our ARGESIM staff for co-operation in producing this SNE issue. In summer 2008, we will publish the special issue SNE 18/2 'Object-oriented and Structural-dynamic Modelling and Simulation II', followed by the regular issue SNE 18/3-4 end of 2008 – being then again in time with SNE issues.

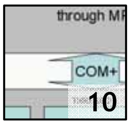
Felix Breitenecker, editor-in-chief, Felix.Breitenecker@tuwien.ac.at

SNE 18/1 in Three Minutes

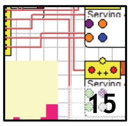
Impressum, Table of Contents: page 4



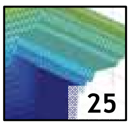
Fast Simulation of Digital and Analog Filters conveniently reduces simulation code by using Desire's vector notation.



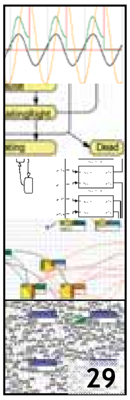
Selected Aspects of the Simulation and Visualization of Mathematical Models in the Computer Network deals with parallel simulation in a distributed computer system.



Discrete Simulation in High School education familiarizes students with aspects of economics and logistics simulation.



Modelling Decision Strategies in Supply Chain Management – A comparison and fusion of modelling approaches uses agent-based models for communication effects.



ARGESIM Benchmarks section:

EUROSIM short info gives a short overview over EUROSIM and its national societies
Report from the EUROSIM 2007 congress
Christian Berweger in simulationists section

Table of Contents

SNE Contact

SNE - Editors /ARGESIM
c/o Inst. f. Analysis and Scientific Computation
Vienna University of Technology
Wiedner Hauptstrasse 8-10, 1040 Vienna, AUSTRIA
Tel + 43 - 1- 58801-10115 or 11455, Fax – 42098
sne@argesim.org; www.argesim.org

SNE Editorial Boards

Simulation News Europe (SNE) is advised by two Editorial Boards. The *SNE Editorial Board*, a scientific editorial board, is taking care on peer reviewing and handling of Technical Notes, Short Notes, Software Notes, Book and Journal Reviews, and of Comparison and Benchmark Notes (Notes Section). The *SNE News Editorial Board* (see News Section) is responsible for reports from EUROSIM, EUROSIM societies, International Societies, and Industry News.

SNE Editorial board

Editorial Info – Impressum

SNE Simulation News Europe ISSN 1015-8685 (0929-2268).
Scope: Technical Notes and Short Notes on developments in modelling and simulation in various areas /application and theory) and on bechmarks for modelling and simulation, membership information for EUROSIM and Simulation Societies.
Editor-in-Chief: Felix Breitenecker, Inst. f. Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria; Felix.Breitenecker@tuwien.ac.at
Layout: Markus Wallerberger, ARGESIM TU Vienna; markus.wallerberger@gmx.at
Printed by: Grafisches Zentrum, TU Vienna, Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria
Publisher: ARGESIM/ASIM; ARGESIM, c/o Inst. for Scientific Computation, TU Vienna, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria, and ASIM (German Simulation Society), c/o Wohlfartstr. 21b, 80939 Munchen

TECHNICAL NOTES

Fast Simulation of Digital and Analog Filters: Using Vectorized State Equations

Granino A. Korn, University of Arizona, USA, *gatmkorn@aol.com*

We present new compact programs for computer simulation of digital and analog filters defined in ordinary transfer-function form. Such filter models used to require a substantial amount of code for inner products and multiple delay or integration operations. The open-source program Open Desire formulates the filter operations as short, easily readable vector expressions, which automatically compile into multiple assignments. In particular, Desire's index-shift operation models *n*-element delay lines, cascaded filters such as gamma delay lines, and repeated integration with a single program line. We show example programs and results for digital and analog filters.

Introduction

We shall describe very compact computer programs for modeling digital and analog filters. Sections 1 and 2, based on textbook references [3,4], quickly review dynamic-system modeling with differential and/or difference equations in scalar and vector form. Our vectorizing compiler translates human-readable vector relations into fast machine code and permits vector-index shifting for modeling delay lines or cascaded integrators. Sections 3 to 4 apply these techniques to digital filters represented in standard transfer-function form; Section 5 deals with filter combinations; cascaded small filters let you experiment with different pole/zero combinations. Finally, Section 6 applies our vectorization technique to the integrator chains needed for modeling analog filters.

1 A simulation language for interactive dynamic system modeling

Desire simulation programs define dynamic-system models with assignments and/or differential equations like

```
u = alpha * sin(w * t + beta) + c
d/dt x = xdot
d/dt xdot = -a * x - b * xdot
```

which are screen-edited into a *DYNAMIC program segment* (Figure 1). Simulation studies are controlled by typed interactive commands and/or by an *experiment-protocol script*. Experiment-control commands set and changes parameters and initial conditions and then call a *simulation runs* to exercise the dynamic-system model, as in

```
t0 = 0          |  t = t0
a = - 5.00      |  x = 17.1
drun
```

`|` is a statement delimiter. `t0` is the initial value of the *simulation time* `t` and usually defaults to 0. When the experiment protocol encounters the `drun` statement the `DYNAMIC` segment is compiled with a fast runtime compiler and runs immediately to produce time-history displays (Figure 1). More elaborate experiment protocols can call multiple simulation runs with modified parameters and different `DYNAMIC` segments [2, 3].

2 Vector operations and delay-line models

Desire experiment-control scripts can declare *vectors* like `x ≡ (x[1], x[2], …, x[n])` and *matrices* like `W ≡ (W[1,1], W[1,2], …, W[n, m] )` with single or multiple `ARRAY` statements such as

```
ARRAY x[n], a[m], b[n], c[n], y[m], W[m,n],
      u[m], v[m], ...
```

`DYNAMIC` program segments can then use the vectors and matrices in *vector assignments*, and *vector differential equations*, say

```
Vector x = a + alpha * b * c
Vector y = tanh(W * x)
Vectr d/dt x = beta * cos (t + c)
```

which automatically compile into multiple scalar operations

$$\begin{aligned}&\texttt{x[i] = a[i] + alpha * b[i] + c[i]} && \texttt{i=1,…,n}\\ &\texttt{y[i] = tanh(}\textstyle\sum_{k=1}^{m}\texttt{ W[I,k] * x[i])} && \texttt{i=1,…,n}\\ &\texttt{d/dt x[i] = alpha * cos(t + c[i])} && \texttt{i=1,…,n}\end{aligned}$$

SNE 18/1, April 2008

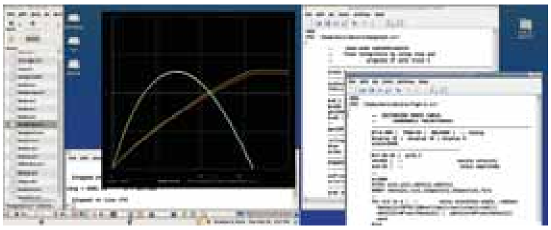


Figure 1. Unix dual-screen display showing Open Desire running with two screen-editor windows. Programs in either editor window can be run to compare models.

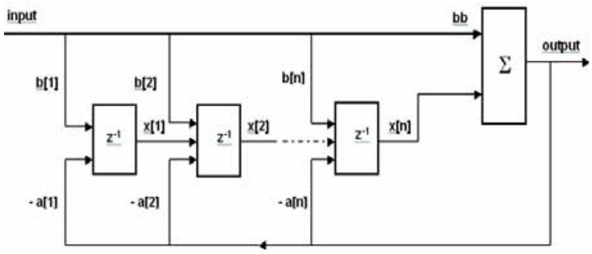


Figure 2. Block diagram of a digital filter with the z transfer function

There is no vector-loop overhead. We can also compute inner products

$$p = \sum_{k=1}^{m} u[k]\ v[k]$$

with *inner-product assignments* `DOT p=u*v`, again without program-loop overhead.

Given a vector `x ≡ (x[1], x[2], …, x[n])` and an integer `k`, the *index-shifted* vector `x{k}` is the vector `(x[1+k], x[2+k], …, x[n+k])` where components with indices less than 1 and greater than `n` are simply set to 0. In particular, repeated execution of the assignments

```
Vector x = x{-1} | x[1] = u
```

neatly models shifting successive samples of a function `u(t)` into a simple *tapped delay line* with tap outputs `x[1], x[2], …, x[n]`. Note that the assignment `x[1] = u` overwrites the `Vector` operation's assignment to `x[1]`.

Desire vector operations have been used extensively for model replication (parameter-influence studies, Monte Carlo simulation) and in simulations of neural networks, fuzzy-logic controllers, and systems involving partial differential equations [3]. We shall now apply them to create efficient models of digital and analog filters.

3 Modeling digital filters

An *n*th-order linear digital filter [1,5] with the classical transfer function *z*

$$H(z) = \frac{bb\, z^n + b[n]\, z^{n-1} + b[n-1]\, z^{n-2} + \ldots + b[1]}{z^n + a[n]\, z^{n-1} + a[n-1]\, z^{n-2} + \ldots + a[1]} \qquad (1)$$

can be represented by the block diagram in Fig. 2. The time `t` is read at the sampling points `t0, t0 + COMINT, t0 + 2 COMINT, …`. Normally the initial time `t0` defaults to zero. Practical filters are often realized as cascaded and/or parallel combinations of simpler filters [1].

Our digital filter is modeled with the difference-equation system

1 input = (*given function of* t)
2 output = x[n] + bb * input
...
3 x[1] = z^{-1} (b[1] input - a[1] output)
4 x[2] = z^{-1} (x[1] + b[2] input - a[2] output)
...
5 x[n] = z^{-1} (x[n-1] + b[n] input - a[n] output)

Computer-simulation software solves such difference equations by successive substitutions, starting with given initial conditions for each state variable `x[i]` on the right-hand side. A Desire program would represent the filter with the readable assignments

```
1 input = (given function of t)
2 output = x[n] + bb * input
     Place display or store commands here
3 x[1] = b[1] input - a[1] output
4 x[2] = x[1] + b[2] input - a[2] output
  ...
5 x[n] = x[n-1] + b[n] input - a[n] output
```

Listing 1. Desire program modeling difference equations representing an n-th order linear digital filter.

Each of these assignments relates directly to the block diagram in Figure 2. The program repeatedly executes these assignments in order, with `t` successively set to `t = t0, t0+COMINT, t0+2 COMINT, …` on the right-hand side of each assignment. This updates the assignment targets on the left for the next sampling time. For `t = t0`, the right-hand expressions are initialized with the given initial values of `x[1], x[2], …, x[n]`, which usually default to 0.

We remark that the order of difference-equation assignments must be carefully observed (some simulation languages order scalar differential-equation systems automatically, but difference equations must be ordered by the programmer). Placing display, print, or store commands (such as `dispt v, x[1]`) just ahead of the first state-variable updating assignment (`x[1] = input - output`) will ensure that state variables `x[i]` and defined variables like `input` are both sampled at the same sampling time `t`.

4 Vectorization makes the program much simpler

Simulating, say, a 50th-order digital filter would require programming `n` + 2 = 52 assignments (listing 1, lines 3–5), but there is a better way. The Desire simulation language can declare the state variables `x[i]`, and the filter coefficients `a[i]` and `b[i]` as `n`-dimensional *vectors* `x = (x[1], x[2], …, x[n])`, `a = (a[1], a[2], …, a[n])`, `b = (b[1], b[2], …, b[n])` with

```
ARRAY x[n], a[n], b[n]
```

We then invoke the index-shift operation defined in Sec. 2 to replace our `n` + 2 assignments with *only 3 program lines*

```
input = (given function of t)
output = x[n] + bb * input

Vector x = x{-1} + b * input - a * output
```

Listing 2. Same filter as in listing 1, using Desires vector notation introduced in section 2.

Our runtime-compiled vector operations cause no assignment-loop overhead.

Figures 3a and 3b show a complete program simulating a 20th-order digital filter. We obtain the filter response to a unit impulse at `t = t0 = 0`, we programmed `input = swtch(1 - t)`. The impulse response equals 1 for `t = 0, t = COMINT, …, t = (n - 1) COMINT` and then goes to 0. More elaborate Desire experiment-protocol scripts can call a fast Fourier transform to produce the frequency response of the filter (Fig. 3c).

5 Combining simple filters

As special cases of the program in listing 2, we can model a *simple recursive filter* with

```
1 H(z) = 1/{z^n + a[n]z^(n-1) + a[n-1]z^(n-2) + … + a[1]}
2 ARRAY x[n], a[n]
3 input = (given function of t)
4 output = x[n]
5 Vector x = x{-1} - a[1] * output |
      x[1] = input - a[1] * output
```

Listing 3. Simple recursive filter. Note that in line 5, the final assignment to `x[1]` overwrites the `x[1]` component of the `Vector` assignment.

Box A: Digital Filter

```
DYNAMIC
--------------------------------------------------
input = swtch(1-t) |
    -- for impulse response; substitute a desired input signal
output = x[n] + bb * input
Vector x=x{-1} + b * input - a * output
dispt output
```

Figure 3a. A complete DYNAMIC program segment modeling any desired filter with the z transfer function

$$H(z)=\frac{bb\,z^{n}+b[n]z^{n-1}+b[n-1]z^{n-2}+\ldots+b[1]}{z^{n}+a[n]z^{n-1}+a[n-1]z^{n-2}+\ldots+a[1]}$$

`input = swtch(1 - t)` is a unit impulse at `t = t0` = 0 and produces the filter inpulse response.

```
--                                         DIGITAL FILTER
--------------------------------------------------
display N1 | display C8 | display R | -- display colors
NN=4096 |      -- number of samples
--
--   parameters for  H = (z^n + 1)/[z^n - z^(n-1)])
--
n=20 | ARRAY x[n], a[n], b[n]
--
a[n] = -1
b[1] = 1   |    -- all other a[i], b[i] default to 0
bb = 1
--------------------------------------------------
t = 0   |  -- initial value of t
drun   |  -- make a simulation run
```

Figure 3b. This experiment-protocol script declares vector arrays and sets the parameters `n`, `a[i]`, `b[i]`, and `bb` for the 20th-order filter with he z transfer function `H = (z^n - 1)/(z^n - z^(n-1))`.

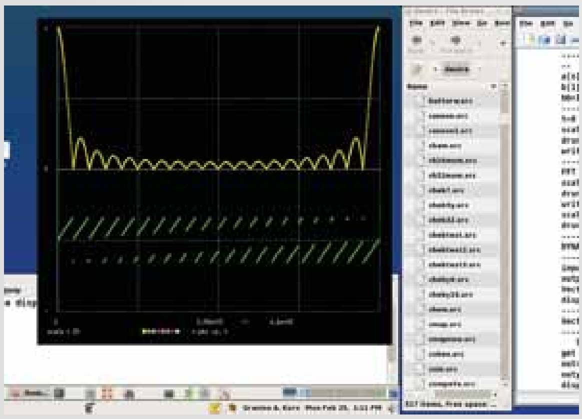


Figure 3c. Linux screen showing graphs of the amplitude and phase response for a filter with the `z` transfer function `H(z) = (z^n - 1)/(z^n - z^(n-1))`. The frequency response was obtained as the fast Fourier transform of its impulse-response function. The order `n` of the filter is 20.

A FIR (finite-impulse-response) filter is modeled with

```
H(z) = {bb z^n+b[n] z^(n-1)+b[n-1] z^(n-2)+ … +b[1]}/z^n
ARRAY x[n], b[n]
input = (given function of t)
output = x[n] + bb * input
Vectr x = x{-1} + bb * input
```

Listing 4. A finite impulse response (FIR) filter.

Compact filter models like (listings 2, 3, and 4) can be cascaded, and filter outputs can be added to represent parallel combinations of filters. Cascading several small filters is useful for designing filters with different pole/zero combinations [1].

6 Vectorized programs for analog filters

We next consider similar compact models of linear analog filters, which are represented by differential equations rather than difference equations. The time `t` is now a continuous variable starting at `t = t0`. The block diagram in Figure 4 implements the transfer function

$$H(z) = \frac{bb\,s^n + b[n]s^{n-1} + b[n-1]s^{n-2} + \ldots + b[1]}{s^n + a[n]s^{n-1} + a[n-1]s^{n-2} + \ldots + a[1]} \quad (2)$$

This transfer function represents a differential-equation system, which is modeled by the easily readable Desire program

```
input = (given function of t)
output = x[n] + bb * input

d/dt x[1] = b[1] * input - a[1] * output
d/dt x[2] = x[1] + b[2] * input
                        - a[2] * output
...
d/dt x[n] = x[n-1] + b[n] * input
                        - a[n] * output
```

The initial values `t0` and `x[i]` normally default to zero. Since there is no difference-equation updating, display, print, or store commands can be placed at the end of the program.

Instead of programming `n` scalar differential equations, we declare `n`-dimensional vectors `x`, `a`, and `b`,

```
STATE x[n] | ARRAY a[n], b[n]
```

noting that differential-equation state vectors like `x` must be declared as separate `STATE` arrays. We can then again program the complete filter model in only 3 lines:

```
1 input = (given function of t)
2 output = x[n] + bb * input
3 Vectr d/dt x = x{-1} + b * input - a * output
```

Listing 5. Desire program modeling differential equations for analog filters using the vector notation. Line 3 represents a chain of *cascaded integrators*.

We obtain the impulse response by setting `input = 0` and the initial value of `x[1]` to 1. The amplitude/phase frequency response is then obtained with Desire's built-in FFT routine. Figure 5 shows results and a complete program.

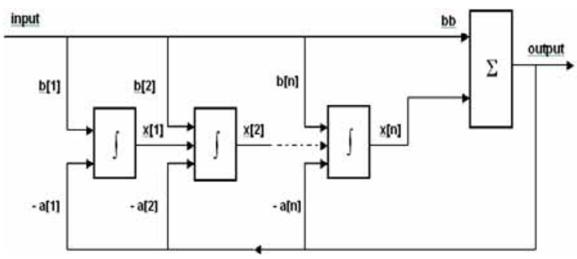


Figure 4. Block diagram of an analog filter with the z transfer function

7 Concluding remarks

Our new programming technique is not meant to replace specialized programs for filter design [1] as such, but for convenient interactive simulation of control or communication systems that include digital or analog filters. Each filter is defined simply by its transfer-function coefficients `a[i]`, `b[i]` and requires only three program lines, *no matter how the high the order* `n` *of the filter is*.

Interestingly, this technique is not restricted to linear and time-invariant filters. The programs (7) and (9) work just as well when the filter coefficients `a[i]`, `b[i]` are simulation-program variables. This is an interesting topic for future research.

References

[1] J. Smith. *Introduction to Digital Filters*, complete course text on the Web at http://ccrma.stanford.edu /~jos/filters/

[2] G. A. Korn. *Interactive Dynamic-system Simulation under Windows*, Gordon and Breach, London, 1998.

[3] *Advanced Dynamic-system Simulation: Model-replication Techniques and Monte Carlo Simulation*, Wiley, Hoboken, N.J., 2007.

[4] *Neural Networks and Fuzzy-logic Control on Personal Computers and Workstations*, MIT Press, Cambridge, MA, 1995.

[5] Papoulis: *Signal Analysis*, McGraw-Hill, New York, 1977.

[6] J. Principe, et al.: *Neural and Adaptive Systems*, Wiley, Hoboken, N.J., 2001.

Corresponding author: Granino A. Korn
ECE Department, University of Arizona
7750 South Lakeshore Road, Chelan, WA 98816
gatmkorn@aol.com

Received: December 10, 2007
Revised: February 16, 2008
Accepted: February 20, 2008

Box B: Butterworth Low-pass filter

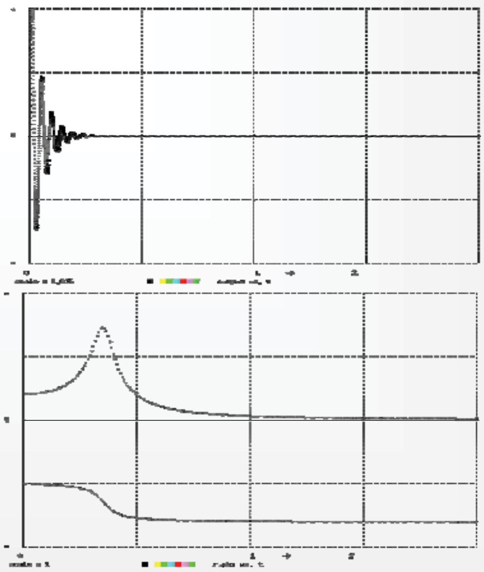


Figure 5a. Desire stripchart-type displays showing the Impulse response and amplitude/phase frequency response for a simple analog bandpass filter with the tranfer function
$H(s) = 1/(s^2 + 40s + 20000)$

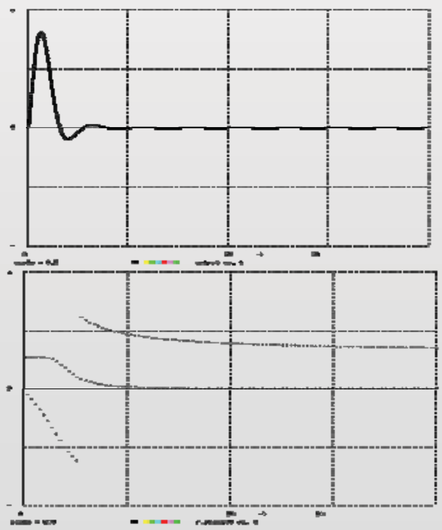


Figure 5b. Impulse response and amplitude/phase frequency response for a 3rd-order Butterworth lowpass filter with the transfer function
$H(s) = 1/(s^3 + 2s^2 + 2^s + 1)$

```
-- SIMPLE BUTTERWORTH LOW-PASS FILTER
-- H = 1/(s^3 + a[1]s^2 + a[2]s + a[3])
------------------------------------------------
display N1 | display C7 | display Q |  -- display
NN=8192 |  DT=0.001 |  TMAX=150
--
n=3 |  STATE x[n] |  ARRAY a[n],b[n]
--
--  array OUTPUT gets output samples for FFT
ARRAY OUTPUT[NN],OUTPUTy[NN]
------------------------------------------------
--  specify the filter parameters
a[1]=1 | a[2]=2 | a[3]=2
b[1]=1           | -- other a[i], b[i] default to 0
bb=0             | -- feedforward coefficient
------------------------------------------------
t=0              | -- (default initial t would be t=1)
x[1]=1           | -- to get impulse response
scale=0.5        | -- display scale
drunr            | -- drunr resets t=0
write 'type go for FFT' | STOP
------------------------------------------------
FFT F,NN,OUTPUT,OUTPUTy
display R        | -- use thick dots
scale=100 | NN=101
drun FFT
------------------------------------------------
DYNAMIC
------------------------------------------------
input=0          | -- for impulse response
output=x[n]
Vectr d/dt x=x{-1}+b*input-a*output
dispt output
---------------------
store OUTPUT=output                | -- fill FFT array
--
--     amplitude/phase display
label FFT
get xx=OUTPUT | get yy=OUTPUTy    | -- FFT arrays
r=sqrt(xx*xx+yy*yy)
phix10=10*atan2(yy,xx)
dispt r,phix10
```

Figure 5c. Complete program for the simple Butterworth filter. The experiment protocol script sets up arrays, parameters and initial conditions and calls a simulation run. The filter-output time history is displayed and also stored in the FFT array `OUTPUT`. If prompted by the user, the experiment-protocol script then computes the FFT and calls a second DYNAMIC program segment labeled `FFT` to display the frequency response.

Selected Aspects of the Simulation and Visualization of Mathematical Models in the Computer Network

Peter Kvasnica, Tomáš Páleník, Dep. of Informatics, Dubček Univ. of Trenčin, Slovakia
Michal Čižmár, ICZ Slovakia, (*kvasnica@tnuni.sk, palenik@tnuni.sk , michal.cizmar@gmail.com*)

The article deals with the design of parallel simulation in information systems of a simulator. The concept arises from computers that create a distributed computer system of a flight simulator. This information system is created by computers and program applications of mathematic models. An important part of this article describes network computing of an aircraft mathematic models and simulation. It also explains the basic ideas of message passing system that was used in the application for data exchange. Mathematic modeling is the art of transformation of a problem from an original application into a theoretic area to mathematical formulations for a numerical analysis.

A significant part of this article describes simulation of aircraft modeling by cluster computing implemented in single-processor architecture. Modeling processes of a flying simulator on cluster's computers also support visualization of a parallel system. Visualization is based on a modern COM+ technology and own created graph components in program of the flight simulator.

SNE 18/1, April 2008

Introduction

It is possible to solve some dynamic tasks in kinematics as well connected with objects motion on the computer. The tasks from kinematics mass point are focused mainly on compound motions. The essence is based on the equations modeling of some individual components on the computer. For actual numerical methods and computers graphics it is possible to solve such tasks easily.

The tasks are successfully solved also from the field of dynamic motion of the aircraft thanks to mathematical modeling and computer simulation. These tasks are possible to reduce to the mentioned problem.

The mathematical model of mentioned tasks is defined by the system of ordinary differential equations. Nowadays it is possible to compute them by two different methods. The first is based on the analytical solving of the mathematical model (system of differential equations) by methods of numerical analysis. In the second case, the solution is realized by solving the mathematical model using the computer technology [2].

The method of an indefinite coefficients, sequential approximation or method of Taylor series expansion is mainly applied. The latter one can be also employed in the numerical solution of differential equations.

1 Continuous mathematical model of the simulation object

A simulation means experimenting with a model represented by the computer program informing about the system through exploration.

The simulation of the simulator systems usually needs creating the arrangement all computer by a sequential process. The application simulation is possible to be divided into next steps [4]:

- Articulating the problem, that shall be resolved by a system simulation
- Collecting and processing information on the system, which is studied through simulation
- Defining the mathematical model of the system
- Creating the computer program to simulate experiments with a defined model
- The project on simulation experiments.

In the process of creating the computer program simulation languages are employed. They are defined and characterized as problem-oriented languages of the automatic programming. They have been created with the aim of making the processes of building-up and debugging/fine-tuning simulation programs much easier.

Simulation languages have other advantages in contrary to classic programming languages. The advantages are as it follows: they accelerate/dispatch model distortion in the program for the computer. Except for this they create available ordinary structure for viewing of the simulating models and enable to carry out changes in the model of the system to the program speed. According to this fact mathematical models are created of the simulator and a linking computer program is built-up. The principle is represented by the project of simulation experiments defining simulation of the described model in the computer network.

Problem formulating is based on the conception of the simulation characteristics determined by mathematical models. The aircraft mathematical models in simulator [1], [5] are modified according to the form needed and desired. The increase in speed is defined by the equation (1), depending on the addition of the change in the fuel supply (2), or addition of the change of the elevator (3):

$$\Delta V(s) = -W_V^{\delta_T}(s)\Delta\delta_T(s) - W_V^{\delta_B}(s)\Delta\delta_B(s), \quad (1)$$

Fuel supply:

$$W_V^{\delta_T}(s) = 5\frac{s^3+1{,}12s^2+62{,}78s+25{,}32}{s^4+1{,}13s^3+62{,}80s^2+28{,}66s+4{,}09}\Delta\delta_T(s). \quad (2)$$

For the elevator:

$$W_V^{\delta_B}(s) = \frac{-0{,}11\cdot(9{,}81s+620{,}97)-0.42\cdot(-9{,}81s-10{,}01)}{s^4+1{,}13s^3+62{,}80s^2+28{,}66s+4{,}09}\Delta\delta_B(s). \quad (3)$$

For the addition of the angle of attack the equation (4) is valid and it defines the addition due to the change of the fuel supply (5) or increase in the change of the elevator (6):

$$\Delta\alpha(s) = -W_\alpha^{\delta_T}(s)\Delta\delta_T(s) - W_\alpha^{\delta_B}(s)\Delta\delta_B(s), \quad (4)$$

Fuel supply:

$$W_\alpha^{\delta_T}(s) = 5\frac{0{,}002s^2-0{,}25s-0{,}1}{s^4+1{,}13s^3+62{,}80s^2+28{,}66s+4{,}09}\Delta\delta_T(s). \quad (5)$$

For elevator:

$$W_\alpha^{\delta_B}(p) = \left(\frac{-0{,}11\cdot\left(-s^3+0{,}89s^2+0{,}012s-2{,}45\right)}{s^4+1{,}13s^3+62{,}80s^2+28{,}66s+4{,}09} - \frac{0{,}42\cdot\left(-s^2-0{,}41s-0{,}025\right)}{s^4+1{,}13s^3+62{,}80s^2+28{,}66s+4{,}09}\right)\Delta\delta_B(s) \quad (6)$$

In equation (2) or (5) the change in the fuel supply represents one step, in Laplace transformation $\Delta\delta_T(s) = 1/s$. In equation (3) or (6) the change in the position of the elevator represents one step, in Laplace transformation $\Delta\delta_B(s) = 1/s$.

2 Mathematical models simulation in the network

The process of simulation experiments needs finding solutions to the separate tasks defined at the beginning of the previous chapter.

A simulation is used so that dynamic characteristics of flight simulators can be studied. In the simulation program a time factor in the form of explicit simulation time is expressed and in that one the model state is changed. There are two opposite methods of work with simulation time that are implemented into the programming technique. It is necessary to choose a constant interval of real time that is considered to be a simulation time unit in the method of constant time step. All simulators' mathematical models need to process a simulating run during this time unit—the simulating step (figure 1).

A flight simulator is a system that simulates the flight of an aircraft as closely and precisely as in real conditions. Hence the parallel computing e.g. on clusters of workstations (personal computers) has very high potential also for this type of simulation. This fact significantly affects the data acquisition and processing order of simulation. Running a computer program in parallel on multiple processors allows running programs faster and solving larger and more complex problems. These parallel architectures are made of multiple processors and multiple memory modules connected together via some interconnection network. They can be divided into two categories:

1. shared memory
2. message passing.

The message passing system, that has been used, combines the local memory and the processor at each node of the interconnection network. There is no global memory so it is necessary to move data from one local memory to another one. In our condition the simulation data must be transferred between computers in the computer network. This is done by sending/receiving pairs of commands which must be written into the application software by a programmer. The data exchanged among processors cannot be shared but they can be copied only [9].

It is useful to apply MPI (Message Passing Interface) which is accessible for the developers and users through message passing libraries for this message passing system. The goal of the MPI is to establish a portable, efficient and flexible standard for message passing that will be widely used for writing these programs. The main function of this interface is to communicate data from one process to another one. Other mechanisms, such as TCP/IP and CORBA, do essentially the same thing, but MPI provides a level of abstraction appropriate for communication of data in scientific computing, whereas TCP/IP is connected

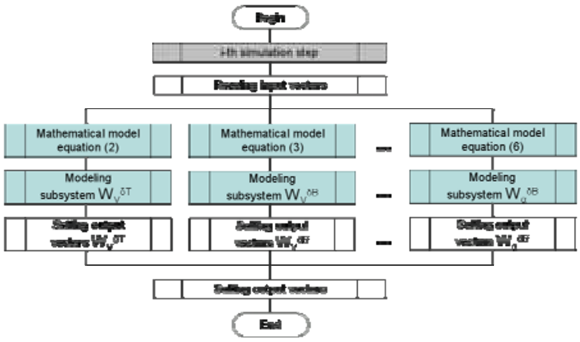


Figure 1. Parallel simulation algorithm of simulators' models.

with low-level network transport and CORBA with client–server interactions. General MPI program structure consists of three steps: initializing of MPI environment, scientific work with message passing calls and terminating MPI environment. This property affects the simulators' mathematical models constructions.

MPI does not specify how the processes are created, it specifies only how they establish communication. All parallelism is explicit. It means that the programmer is responsible for correctly identifying parallelism and implementing parallel algorithms using MPI constructions. Moreover, an MPI application is static: it means that no processes can be added to or deleted from an application after it has been started.

3 Program application in MPI

In this section we will overview technical architecture applied in the flight simulator that is being done. This architecture has an influence upon the message passing program code and it has been finalized in the form of connection of five nodes. Each node consists of a processor P and a local memory M. We used processor *Celeron 2,4 GHz* with *512 MB RAM* connected through *100 Mb/s Fast Ethernet*. These nodes cooperate to perform a task by independently computing with their local data and communicating data with central process by exchanging messages (see Figure 2).

One node is designed as central computer and the others are compute nodes and each of them processes just one mathematical model. The actual simulation data are sent in periodical time to central node. It collects data in real-time and display it in the monitor. We used point-to-point type of operations for this collecting. MPI point-to-point operations typically involve message passing between two (and only two) different MPI processes. One process is performing a send operation and the other process is performing a matching receive operation.

The system consists of an MPI program in SPMD (Single Program Multiple Data) style, running on a cluster of five processing elements. We considered using C++ language as a language for implementing MPI because it is appropriate for numerical computations. The architecture is developed in the framework of the MPICH software architecture - a well-known MPI implementation that is used worldwide. It is known for its portable software architecture that allows achieving reasonably good message passing performance without extensive porting efforts and platform-specific tuning.

The message passing is realized by library functions calls. These sending/receiving commands for message exchange are implemented in source code of application assigned to execution in selected nodes. Nodes are also able to store messages in buffers and perform send/receive operations at the same time as processing. The creation of such program is essential for its correct operating. Two fundamental functions for sending and receiving messages that we used are [11]:

- `MPI_Send (parameters)`
- `MPI_Recv (parameters)`

where `MPI_Send()` is called at the side of the message sender. The corresponding `MPI_Recv()` is placed at the destination process. Both functions are blocking that is important for the mathematical models simulation synchronization of flight simulator.

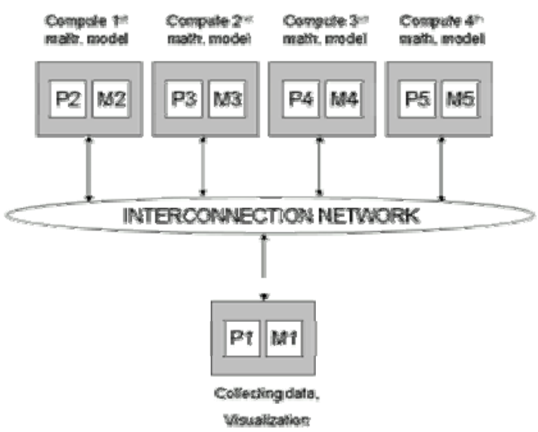


Figure 2. Design of interconnection architecture of simulator

Used MPICH system includes also the basic tools for parallel running process coordination - barrier synchronization (see `MPI_Barrier` call in Listing 1). We used this method for processes time synchronization after MPI initialization call. Barrier means that each process stops at this barrier until all processes in job reach it.

```
MPI_Init()
...
MPI_Barrier()
if (processor == 0){          // central node
   // collecting data and visualization
} else {
   switch (processor)         // other compute nodes
   {
      case 1: ModellingEquation1(); break;
      case 2: ModellingEquation2(); break;
      case 3: ModellingEquation3(); break;
      case 4: ModellingEquation4(); break;
   }
}
MPI_Finalize()
```

Listing 1. Pseudocode of MPI application

4 Graphic interpretation of model results

Data acquisition and processing is supported by visualization of data obtained from MPI program in our solution. There are three main practical solutions to visualize data in MPI programs by implementing:

1. auxiliary graphical libraries (e.g. MPE, but graphic capabilities are very simple),
2. MPI directly integrated in higher languages with graphic support,
3. technology integration (e.g. by COM or DCOM technology).

We implemented visualization as an autonomous module to view the results as they are generated in simulation using COM+ technology. COM (Component Object Model) defines an application-programming interface (API) to allow the creation of components. They are used for integrating custom applications or to allow diverse components to interact.

In order to interact, components must adhere to a binary structure specified by Microsoft. As long as components adhere to this binary structure, components written in different languages can interoperate. COM+ is an enhanced version of COM that provides better security and improved performance. The experience gained in this field leads us to use C# language based on .NET platform for its good 2D visualization support in the simulation of flight simulator models. Thus the application on central node consists of two layers (see Figure 3):

- application layer coded in C++ that uses also all MPI functions,
- presentation layer that is coded in C# for data visualization.

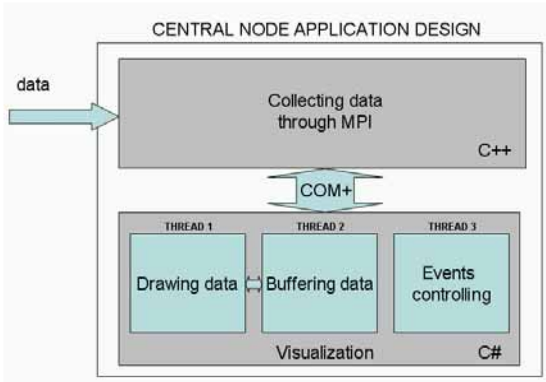


Figure 3. Block diagram application connected by COM+

The layers are connected through COM+ technology. The visualization of simulation results is realized using three threads:

- thread that process data from application layer using buffer,
- thread that asynchronously uses data in mentioned buffer and draws the graphs using GDI+ technology,
- thread for events controlling of main graphic window and its components.

All computing nodes are able to store their data in the buffer on central node. The buffer presents the reserved memory region as the storage for data that waits for their visualization.

Visualization data are buffered in a separate thread, to obtain better data collecting speed. The requirement is done to make data collecting process as fast as without visualization. Buffer was implemented as generics class with initial size 2000 points. Due to the fact that graphical context cannot be shared by more than one thread, the mutex was used in these sections: point addition, scale updating, graph refreshing and getting status of drawing buffer. This was a reason to use NET in version 2.0, because the former version does not support generics classes and mutexes only partially. Used version supports two types of mutex: local and named system mutexes, we used the first one. The dynamic graphs are created as own coded component that supports all basic functions for drawing graphs. It is possible to create multiple instances of this component for each observing simulation variable. COM+ object should be digitally signed and described by Global Unique Identifier (GUID), to be accepted by .NET Platform.

The simulation time of mathematical models of flight simulator is delimited to 25 seconds for sample and integration step is 5 milliseconds. The simulation inter-results are sent in periodical time to the central node. We obtain Figure 4 as the graphical results from simulated data of mathematical models in realized message passing system.

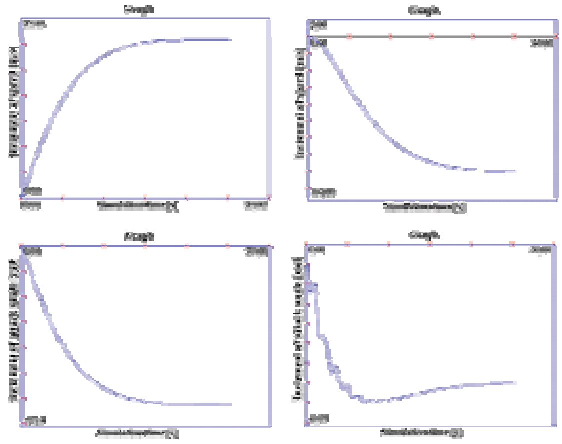


Figure 4. Simulation results of equations (2) – top left, (3) – top right, (5) – bottom left, (6) – bottom right

5 Conclusion

The simulation of the mathematical models with flight simulator cluster technology was based on mathematical formulas expressed by equations (2) and (3) or (5) and (6). It was used single-processor architecture and MPI libraries with programming support in the Windows operating system environment. It was used for faster processing of mathematical models and as well as visualization. We introduce also the connection architecture and program model of the simulator, where we split the visualization and processing part of the application. For this purpose we use COM+ technology which is non-standard way of visualization handled data in MPI applications. The opportunities for a future application development include:

- Encapsulation of existing COM object in to DCOM technology that gives us possibility to show graphical data also on other node in network as central one
- Using OpenGL (CsGL or TAO framework) instead of GDI+ to visualize the graphs.

This method can be also used in other models of the configuration of the flight simulator under Rolfe [5]. Thus it is necessary to improve and modify the mathematical formulas in a suitable way for the stimulation on the cluster's nodes of the flight simulator.

References

[1] Blakelock, John H.: *Automatic control of aircraft and Missiles*. New York, John Wiley & Sons. Inc. 1991. ISBN 0-471-50651-6.

[2] Bartsch, Hans J., *Matematické vzorce*. Praha, Mladá fronta 1996. s.720, ISBN 80-204-0607-7.

[3] Feitelson, D., G., Rudolf, L. *Parallel Job Scheduling: Issues and Approaches* Lecture Notes. In: Computer Science, Vol. 949, 1995.

[4] Kolektív autorů, *Aplikovaná matematika*, Oborová encyklopedie, SNTL 1978, s. 1159-1161, L28-S2-V-51/11589.

[5] Krasovskij, A., A. *Sistemy avtomaticeskogo upravlenija poletom i ich analiticeskoje konstruirovanie*, Moskva, 1973.

[6] Pota, S., Sipos, G., Juhasz, Z., Kacsuk, P., Distributed Parallel Systems. *Parallel Program Execution Support in the JGRID System*, Springer Science, New York 2005, ISBN 0-387-23094-7, p. 13-19.

[7] Rolfe J.M., Staples K.J.: *Flight Simulation*, Cambridge University Press: Cambridge, 1986, s. 42, ISBN 0-521-35751-9.

[8] Sun Microsystems, *Jini Technology Core Platform Specification*, http://www.sun.com/jini/specs, 2005.

[9] El-Rewini, H., Abd-El-Barr, M.: *Advanced Computer Architecture and Parallel Processing*, John Wiley & Sons, Inc.: New York, 2005, ISBN 0-471-46740-5

[10] Gropp, W., et al.: MPICH: *A high-performance, portable implementation for the MPI message-passing interface*, Parallel Computing Vol. 22 (1996), ISSN 0167-8191, p. 789-828.

[11] MPICH2. online: www.mcs.anl.gov/mpi/mpich2

Corresponding author: Peter Kvasnica
Dept. of Informatics, Faculty of Mechatronics,
Alexander Dubček University of Trenčín,
Študentská 2, 911 50 Trenčín, Slovak Republic
kvasnica@tnuni.sk

Received: January 22, 2008
Revised: March 3, 2008
Accepted: March 10, 2008

EDUCATION NOTES

Discrete Simulation in High School Education

Erwin Rybin, HTL Donaustadt, Austria, *erwin@rybin.eu*

This paper deals with different aspects in the area of economic and logistics simulation in High School education. Starting with a general overview there follows a brief theoretical discussion with concern of queueing theory within the daily life background of the students. The illustration of typical problems in the production and supply chain area with the help of animated computer models forms the main part of the work. Thus the problems can be represented descriptive and the students can point out, which mathematical, technical and economical considerations would be necessary, in order to master these and still more complex problems.

1 Introduction

Discrete Simulation is not used very often in High School education, since the modeler software is expensive and the models are often very difficult to understand. Nevertheless there exist student versions of all main modeling software, which allow modeling and saving at least simple models, which could be used to open the mind for more complex problems in logistics and economy.

In this paper I will describe the use of Enterprise Dynamics (student version) in High School projects in the fields of economics and/or logistics, so the main audiences are schools with technical and economic focus. It is partly based on my MBA thesis [1]. As an easy start, I will begin with a very simple model which deals with the daily experience of most students, the queues at the school canteen.

1.1 Canteen Queues

After an introduction to the ED modeling system, the students try to construct the proper situation in their school in the simulation tool (see Figure 1).

Starting with a primitive 1 Source/1 Queue/1 Server (=Counter) – Sink Model, where first analyses of the Queue behavior in connection with the queueing theory could be checked, the students go on to a little more complex simulations to describe better the reality. One interesting question is: "Should we pool separate queues into a single queue or not?"

So a model could have two ore more servers, more queues ore more sophisticated action of the persons in the queues (e.g. one large queue, 2 ore more servers, 2 ore more queues with the logic to enter the shortest queue etc.)

1.2 Canteen Places

A second typical model in discrete simulation is the occupation of gives places in a restaurant (Figure 2). This model can easily transformed for the students own school canteen (or any fast-food nearby).

Although it seems to be a simple model, there can be explored a lot of interesting questions, e.g. regarding group behaviour (will a given group separate on different tables? How long will it take? etc.)

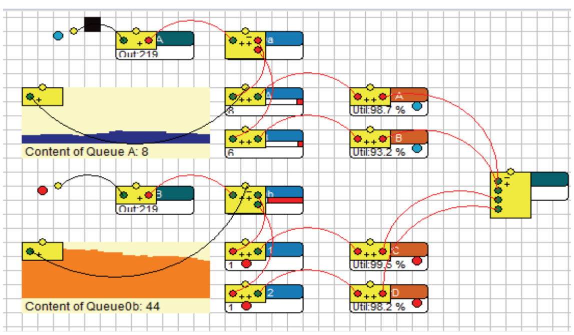


Figure 1. Simple Canteen model with two separate queues or one main queue

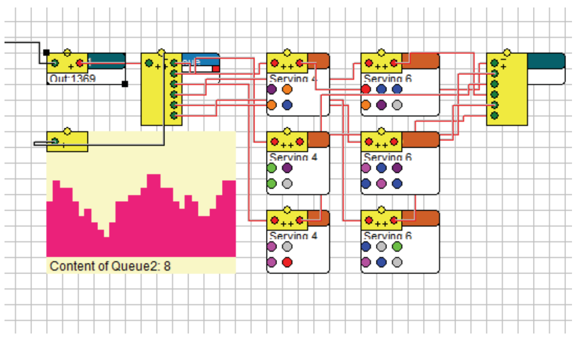


Figure 2. School canteen with 3 tables with 4 seats and 3 tables with 6 seats

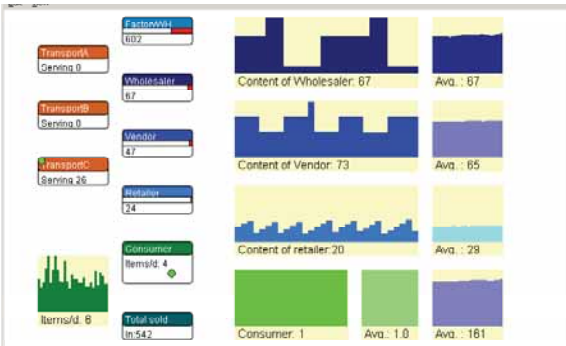


Figure 3. Animation of a simplified Supply Chain

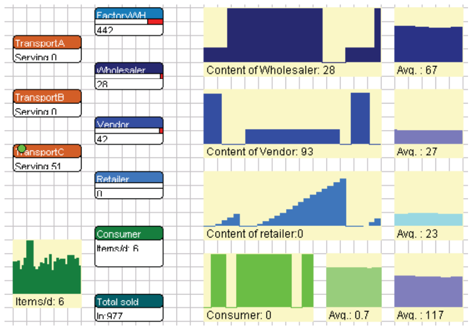


Figure 4. Average Stock: 117, Satisfaction of customer demand: 70%

2 Problems in order and sell processes

In traditional ordering and delivery processes often articles are out of stock at the reseller, which leads to unsatisfied of the consumers. In order to escape this problem, the dealers must hold an often very large stock, which naturally leads to appropriate costs.

2.1 Example: Distribution chain

The following model, developed by the author basing on a precious work in the Austrian Research Center Seibersdorf together with the Technical University Vienna [2], particularly demonstrates the value of sharing information across the various supply chain components. The implementation was done in Taylor ED [3], like all other models in this paper.

Consider a simplified supply chain, consisting of a single retailer, a single vendor which supplies the retailer, a wholesaler which supplies the vendor, and a single factory supplied with unlimited raw materials which makes the product and supplies the wholesaler (Figure 3). Each component in the supply chain has limited storage capacity, and there is a supply lead time and order delay time between each component. Each component in the supply chain tries to meet the demand of the downstream component. Any orders which cannot be met are recorded as backorders, and met as soon as possible.

The transport devices have a transport time of 24 hours. The members follow the strategy to ordering a demand to refill the storage to a certain maximum as soon as a certain minimum content is reached. The order time is 24h for the retailer, 48h for the vendor and 72 h for the wholesaler. The delivery delays are 48h for the vendor and 24h for the wholesaler.

In the first scenario, shown in Figure 4, the retailer sets the minimum stock to 8 pieces, with a target quantity of 30, this is the average consumption of one day plus 33% reserve. With this he could secure the ability to deliver during the transportation time of 24h. But since there are order delays at the retailer and supply delays at the vendor (24h respectively 48h), there are bottlenecks and the customer demand (approx. 6 pieces a day) can not fully be covered. Analog, but even worse is the situation between vendor (20/50) and wholesaler (30/80).

Since all three, retailer, vendor and wholesaler have to fight against delivery bottlenecks but don't know why these are occurring (no flow of information in the moment), they increase their stock capacity in the following scenario: retailer (40/60), vendor (50/100), wholesaler (60/120). So the customer demand can be covered by 100%, but the average stock increases by nearly 40% (see Figure 5).

To evade this dilemma, they are using the most trivial form of supply chain management: They share the information, how long the transport-, order-, and supply delays are. So they can optimize their stocks without neglecting the satisfaction of the customers, displayed in Figure 6.

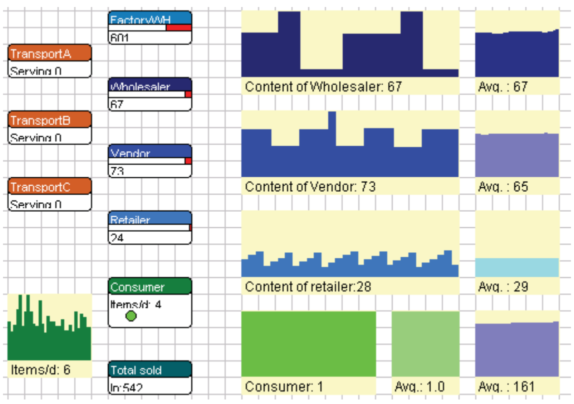


Figure 5. Average Stock: 161, Satisfaction of customer demand: 100%

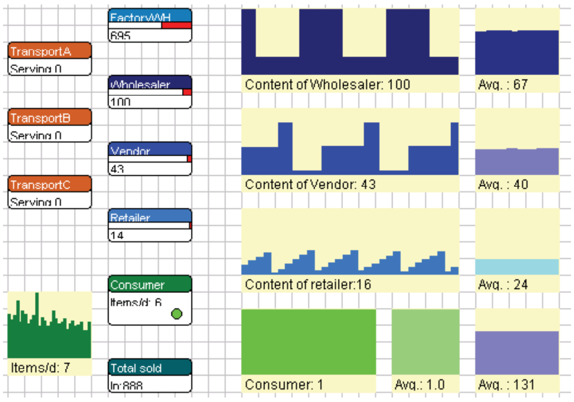


Figure 6. Average Stock: 131, Satisfaction of customer demand: 100%

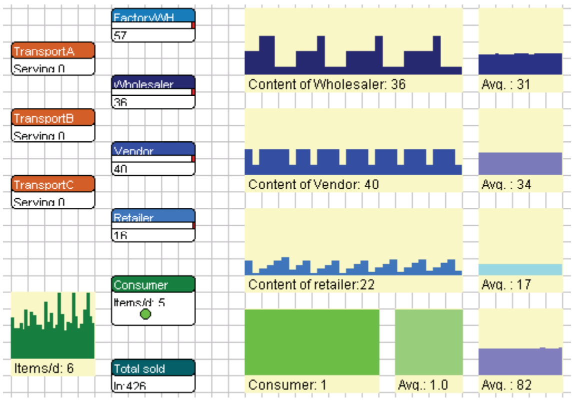


Figure 7. Average Stock: 82, Satisfaction of customer demand: 100%

We can only expect an improvement of the situation, when we shorten the transportation time or the delays in the order and delivery processes. The first can be done my means of modern transportation software, the latter by using e-commerce solutions.

If we reduce the order delay to 1h and the supply delay to 2h, Figure 7 shows an enormous increase of stock if we still use the same warehousing strategies.

After an optimization of the strategies of the retailer (12/36), vendor (20/40) and wholesaler (30/60), we see an overall improved situation (see Figure 8): the average stock is 30% less than without the use of e-commerce applications, but still the customers are satisfied to 100%, and there is even some extra stock planned for unforeseen events.

One could see the basic correlations in the supply chain in this simplified model, as well as some practical solutions for the problems occurring there.

Beside the distribution problems the next examples also deal with the supply of raw material and production, first under a strategic, then under a tactical and in the end under an operational point of view. For every case I will also present and discuss a suited model.

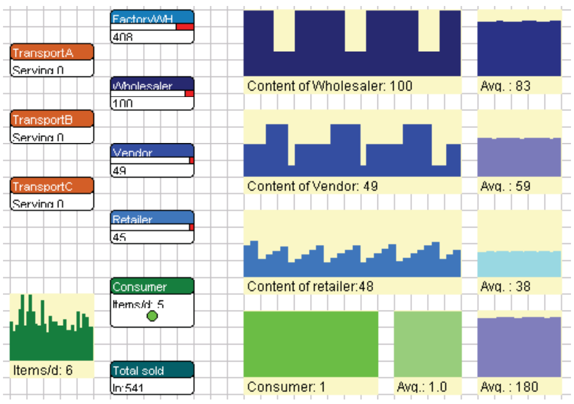


Figure 8. Average Stock: 180, Satisfaction of customer demand: 100%

3 Problems with production and distribution sites

Strategic planners of companies face an extremely complex task in the age of the globalization. By the possibilities of world-wide production, distribution and markets there are a multiplicity of options, so it is nearly impossible to find even a temporary optimum without the right tools.

3.1 Example: Transport optimization and market expansion

The following example starts in the initial scenario from a domestic production and sales structure. The same article is produced at three production locations, supplied to a central distribution centre and delivered to three customers (and/or vendors) from there.

The factories need approx. 10h (normally distributed) per product, with production costs of 5.00 per hour. The transporters (trucks) have a certain, variable capacity (essentially 1 or 2 articles per travel), and costs of 3.00 per piece per hour. The transportation times are approx. 3-6h, depending upon the length. If products are not fetched in time from the factories, there are storage costs of 0.50 per hour per piece.

To be able to easily compare between the scenarios all costs of domestic production and distribution are summed up. One substantial restriction in this model is the infinite customer demand, i.e. the more becomes produced (and delivered), the larger the gain of the company (100.-/piece) is.

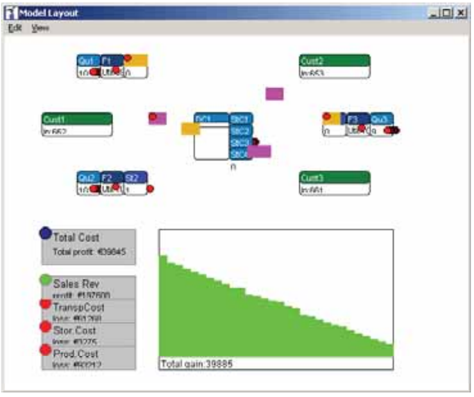


Figure 9. Animation of distributed production sites and customers

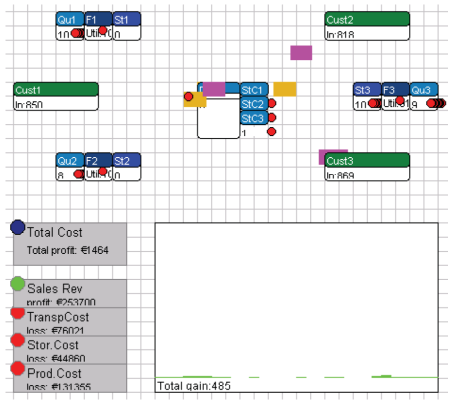


Figure 10. Even product distribution with delivery problems

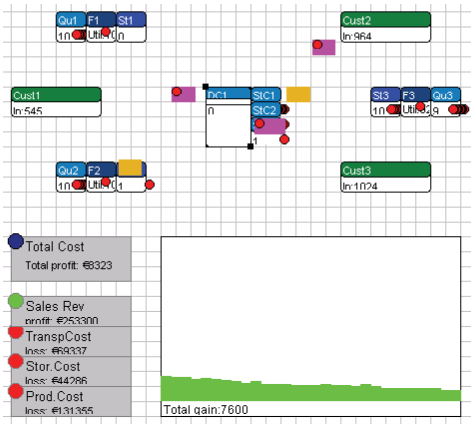


Figure 11. Uneven distribution of products due to transport times

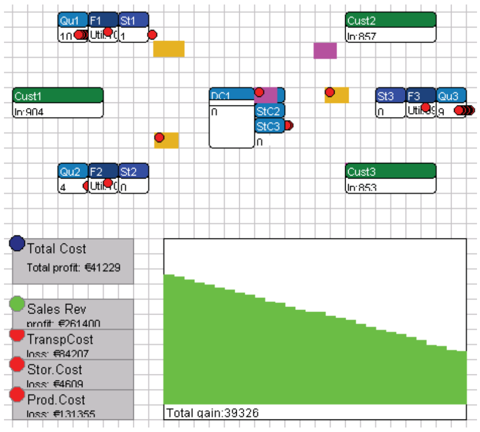


Figure 12. Well balanced production and distribution of products resulting in high profit

Optimising of Transportation- & Warehouse capacities

As we see in Figure 9, *Customer1* as well as *Factory3* have a longer distance to the distribution center than the others (Figure 10): while *Customer2* and *Customer3* buy nearly 2000 products together, *Customer1* buys just approx. 550 products, but for the only reason because less goods are delivered to him than to the others.

For an optimal supply of *Customer1* we double the capacity of the corresponding transport device (which means that also the transportation costs are doubled to 3,-/piece/h). Figure 11 shows that the overall sales increase just a little bit, because the produced articles are just spread over the customers. So the total profit sinks.

On the other hand we realize that the warehouse at *Factory3* is always full, because the transport lenght to the distribution center is too long for the capacity of the transporter.

If we double again the capacity (Figure 12), of course the transportation costs increase, but the costs for storage decreases dramatically (1/10) and more products are available for sales, so the total profit clearly increases.

3.2 Expansion to a foreign country

In the following example, modelled in Figure 13, the activities of the company are extended to the neighbour country. In a first step only products to customer residing there are sold, however the price is lower than in the homeland (80, - instead of 100, -). Further there is an option for an acquisition of a factory with similar products and lower production costs.

Figure 13. Expansion to a foreign country

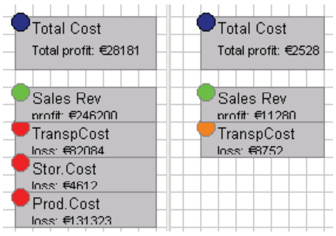


Figure 14. Expansion leads to small increasing profits

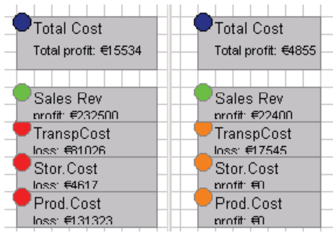


Figure 15. Further expansion negatively influences the profit in the home market due to delivery problems

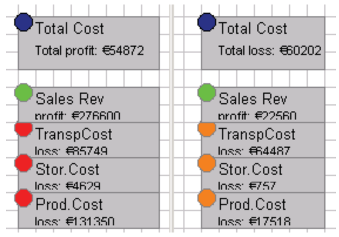


Figure 16. High losses due to enormous transportation costs

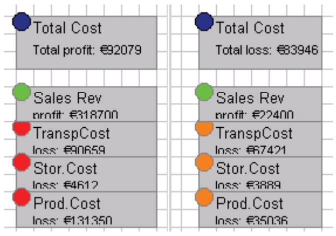


Figure 17. Still losses, even after productivity enhancement

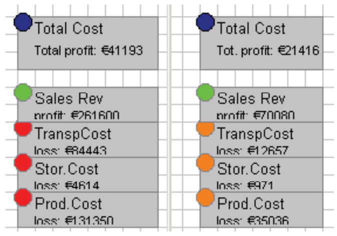


Figure 18. With strict separation, the earnings increase, but without profiting from synergy effects

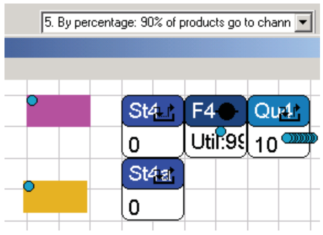


Figure 19. Intelligent distribution of products, produced in separated markets

In the first phase of the expansion all products are supplied to the new customers by the central distribution center. Although the transport costs per hour are lower, substantial costs result due to the large distance. Nevertheless a clear profit of approx. 2500, - results in the new market, as can be seen in Figure 14.

In the case of duplication of the transportation capacity also the profit rises nearly about the double (4800,-).

However in the homeland market there are now fewer products for the sales available, so the turnover and the profit decreases, due to the lower price abroad as well as the high transport costs (Figure 15).

In the second phase a factory is acquired. Production costs are very small (2.- instead of 5.- /h/piece), however the productivity is just 50%. The produced products are also supplied to the central storage depot and distributed from there, also to the new customers. Despite low production costs and although the turnover in the home market increases, a total loss as a result of the transport costs arises (Figure 16).

If it is possible to increase the productivity to the same level as in the homeland (by doubling the production costs, but still 20% smaller), we can increase the sales in the home country and so the loss due the expensive transportation is not that big (see Figure 17). Nevertheless the total gain is beyond the situation before the expansion.

If the products of the foreign factory are only sold on the foreign market, a large profit in the foreign market arises due the enormous decreased transport costs.

The home market drops naturally back again to the level before the expansion. However both markets are now strictly separate, so there are no synergies in the distribution (Figure 18).

So we now have to establish the possibility to deliver both markets from the acquired factory (Figure 19).

A split-up (e.g. 90% remain in the foreign market) does not lead to an advantage due to the assumption of infinite demand in the system. The sales would just be shifted and despite the higher profits in the home market the profit decrease due the higher transport costs (Figure 20).

The spilt-up only make sense, if there is limited demand in one or both of the markets.

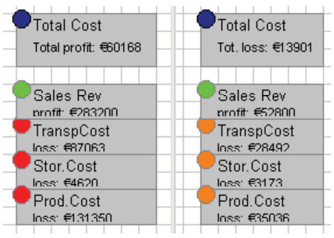


Figure 20. Still losses, but perhaps depending on the model restrictions

The above examples illustrated some points of the problem during the decision processes for the ideal sites (production, distribution, new markets) on the basis very simple basic conditions. In real life, the conditions are much more complicate e.g. the following points have to be considered:

- Regional different consumer behaviour
- Seasonal different consumer behaviour
- Possible market penetration
- Opposition of the personnel against acquisition
- Political border conditions
- Different taxes
- Customs
- National promotions
- Just in Time Production
- Internal or external fleet
- Choosing the right transportation device
- Routing optimisation
- Queue time at borders

Many of these problems can only be solved with experience and instincts in dealing with foreign partners and markets, on the other hand, for the technical and financial aspects today already powerful software tools are available.

4 Problems with Planning

Today many industrial companies are no more on concentrated one location but have several manufacturing plants and/or distribution centres. Medium-term planning over a multiplicity of spatially distributed sites can become therefore very complex.

4.1 Example: Utilization planning of different manufacturang plants

In the example 3 different products are manufactured in 3 factories, whereby the blue product can be produced in factory A, B and C, the green one in factory B and C and red one only into factory C (Figure 21). The products all have different production times and different profit margins and come into the system with different lot sizes, shown in Table 1.

The factories on the other hand have each a capacity of 10 simultaneously machinable products as well as strongly different hourly rates, which can be attributed e.g. to different wage levels or machinery costs, summarized in Table 2.

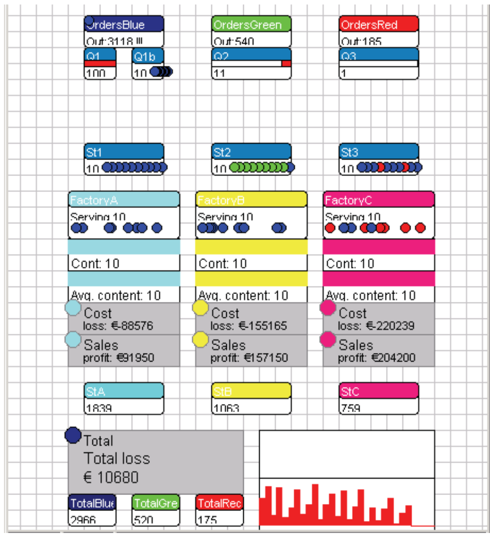


Figure 1. Animation of a planning with different sites

If Factory B produces just the green product and Factory C just the red product, we receive a small profit after 6 month, which is below 1% of the turnover, a randomly dependent, because of the normal distributed production times.

Factory A operates at full capacity, Factories B and C have still 30% free capacity. The losses at start-up (down right in the illustration) are due to less utilization at the start of the production (see Figure 22).

To better use the free capacities of the other factories, in a first phase we also produce blue products at Factory B (Figure 23).

So we can produce 2300 blue products, compared to 1800 before.

The total profit doubles, but is still just around 1% of the turnover. We also recognise the losses at the start-up of the production.

	Blue	Green	Red
Prod. time	approx. 24h	approx. 60h	approx. 168h
Profit	50,-	250,-	1000,-
Number	Max. 20/d	20/d but just Mon, Tue	1/d

Table 1. Specifications for the three products

	Factory A (cyan)	Factory B (yellow)	Factory C (pink)
Costs/ h	5,-	10,-	20,-
Costs/piece/h	1,-	2,-	3,-

Table 2. Specifications for the factories

SNE 18/1, April 2008

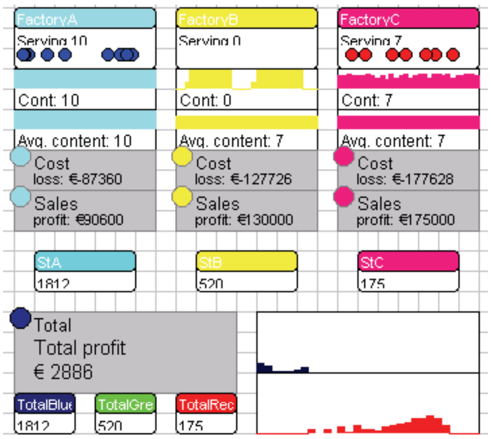


Figure 22. Small profit but large capacities unused

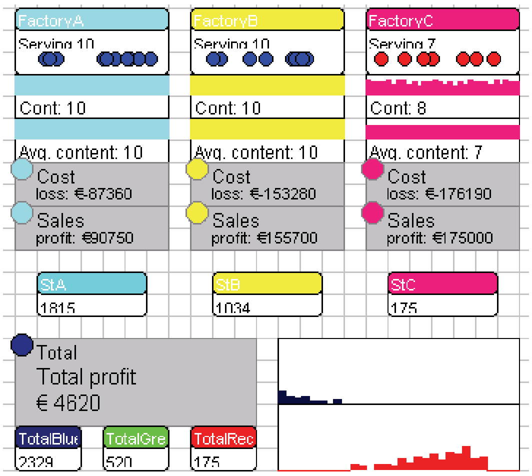


Figure 23. Increasing profit, but still small compared to turnover

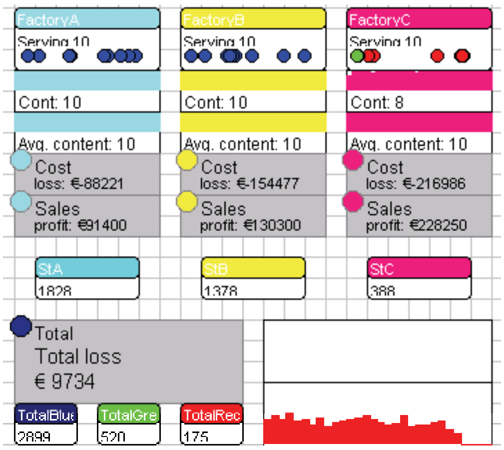


Figure 24. High losses in factory C due to increased production costs

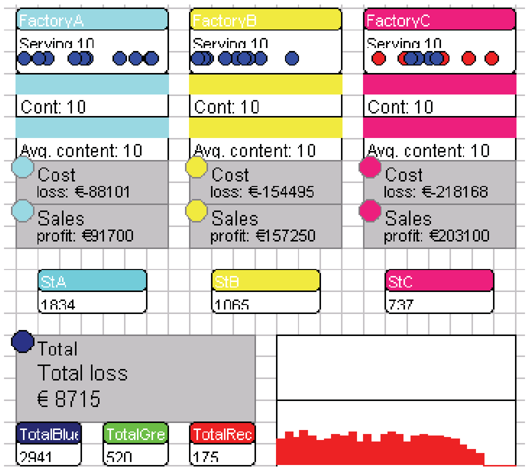


Figure 25. Still losses due to production mix in the different plants

Another possibility is the partly production of the green products in factory C, so factory B has more free capacity for the blue products.

In this scenario, the turnover of the blue products increases to nearly 2900 pieces, but on the other hand there is a total loss, because the production costs increase for both, the blue and the green products.

Due to this rearrangement Factory C has now a large loss (ca. 25000,-), although the utilization is the same as before (Figure 24).

Since the profit for the green products shifts only to Factory C, an split-up of the production is not meaningful.

To reach the demanded output of 2900 blue products also Factory C has to produce blue products. Due the high hourly rates this results in a total loss, however not as large as in the previous scenario (approx. 8700, opposite 9700).

Depending upon the operational defaults the planner in the above example must decide for a certain scenario. There is no optimum possible in this case that connects maximum production and maximum profit, but only a combination of both. The example illustrates the complexity of planning on the basis of very simplified boundary conditions over several production plants (Figure 25).

In practice substantial elements have further to be added:

- an uneven distribution of the orders, e.g.
- seasonal fluctuations
- custom-mode products
- substantial more products

- possibility to increase the capacity (to appropriate extra costs)
- different production costs of the products
- storage costs
- setup-time and reequipping costs
- transportation times and transport costs

Planning and development of scenarios are therefore no longer feasible manually under normal conditions and need the support of suitable tools.

5 Problems on the Shop Floor

Although the most production plants today are equipped with ERP-systems and MSP-Systems, there is often no (or just a poorly) connection between this systems. Simulation can help also in this case.

5.1 Example: Rush order

The following example demonstrates a typical problem which production managers have to deal nowadays: In addition to the running orders an important client needs a rush-order done. The responsible now have to determine very fast, if they can guaranty this order to their client at with what consequences for the already running orders. For the example, modelled in Figure 26, we use simplified conditions:

- 3 different products with the same production time
- 3 production lines with identical machines
- Raw material enters approx. every 10 hours
- For one product a machine needs 8 hours; the capacity can be increased, increasing also the hourly costs
- Set-up time for every machine is 1 hour
- For every order delivered to late, a penalty has to be paid

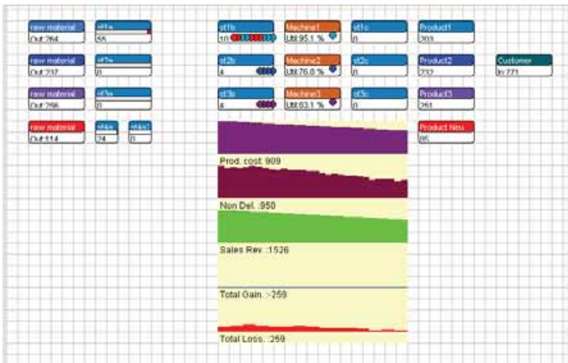


Figure 26. Computersimulation shop-floor production lines

Normally, the machines work with 80% utilization (0.3 products/hour, capacity 0.375 products/hour). Since the raw material arrive with negative-exponential distribution, short queues and short delays are possible. In a simulation of 10000 hours 6 products were delayed, which means 60.- penalty costs. The earning with each product is 2.- and the machines cost 1.- per production unit and hour (see Figure 27).

Now the rush order is also dispatched. If the load is equally distributed to all machines (set-up time 1 hour) and the capacity is not increased we see the following result: because of the high penalty costs, results a high loss, although volume and turnover increases by 1200. To fulfil this rush-order in a profitable way, we have to change system parameters (Figure 28).

One possibility would be to produce all extraordinary products on just one machine, so just one client (and the rush order client) gets the products to late. But as we see after the simulation, the situation is even worse: The penalty increases nearly by 100%, as shown in Figure 29.

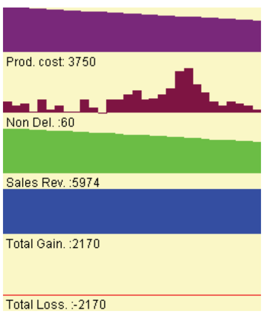


Figure 27. Financial situation before the rush-order

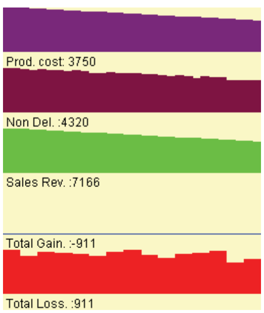


Figure 28. High loss caused by the rush-order

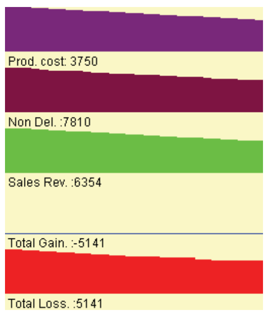


Figure 29. Rush order produced on just one machine

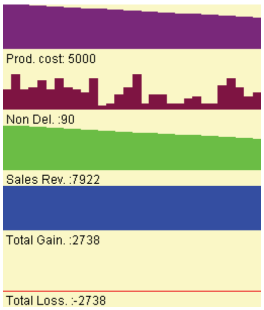


Figure 30. Financial situation with rush-order and more machine capacity

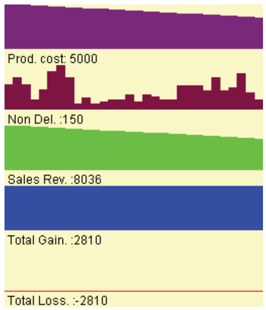


Figure 31. Increasing the capacity of just one machine, thus reducing the set-up times

Another possibility is an equal increasing of the machine capacity, e.g. with more persons at a machine, so the production time would decrease to 6 hours per product. Of course this way also the production costs increases, but since we earn more money with more sold products and pay nearly no penalties, the situation gets much better (figure 30).

Another possibility is to double just the capacity of the first machine, so the production costs rest the same (1x100% instead of 3x33%), but thanks to lower set-up times more products can be produced, resulting in a slightly bigger total gain (figure 31).

As a result of the simulation we see a lot of potential for optimizing shop-floor planning with help of these advanced technologies. Even more useful is simulation in real world problems, like complex production steps, maintenance time, different capacities, specialised workers, necessary tools for each production step.

References

[1] E. Rybin. 2002. *Mikroökonomische Vorteile von B2B & Supply Chain Lösungen*, Abschlussarbeit zum MBA, ausgeführt an der Univ. Krems

[2] S.-M. Rahmi, E. Rybin,. 2000. *Supply Chain Management - The Paper Game*, Eurosim - Simulation News Europe, `www.eurosim.org`

[3] Enterprise Dynamics, Student Version `www.enterprisedynamics.com`

Corresponding author: Erwin Rybin,
HTBLA Donaustadt
Donaustadtstraße 45, 1220 Vienna, Austria
erwin@rybin.eu

Received: February 23, 2008
Revised: March 10, 2008
Revised: March 20, 2008
Accepted: March 25, 2008

SNE 18/1, April 2008

SHORT NOTES

Modelling Decision Strategies in Supply Chain Management – A Comparison and Fusion of Modelling Approaches

Jürgen Wöckl, Lisa Gimpl-Heersink, Alfred Taudes, Vienna University of Economics, Austria
{Juergen.Woeckl, Lisa.Gimpl-Heersink, Alfred.Taudes}@wu-wien.ac.at

This study deals with a simulation approach of decision processes in supply chain management under consideration of related marketing effects on an indivuidual consumer level, modelled by an agent-based approach. Such integrated demand models are often formulated using aggregated variables and don't describe individual effects of the consumers. In this study a typical integrated demand model has been enriched by a specific marketing model, to enrich the aggregated simulation results of classical demand models with characteristics based on individual consumer preferences. So the combined model describes the influence of the human decision process to optimal decisions of integrated demand models.

In detail, communication effects - especially positive/negative satisfaction and local word of mouth effects - which formulate the basic dynamics of the disaggregated consumer model on an individual basis have been implemented using an agent-based approach. Further the resulting demand-influencing characteristics of each consumer have been aggregated to form a model-based demand function. The demand is an important input of many - if not all - supply chain models and so demand is often assumed as a stochastic variable with specific characteristics. As mentioned, here the aggregated individual characteristics are used as a more realistic input into an approved integrated supply-chain model.

Introduction

In economics the choice of a proper modelling approach is a fundamental issue to describe the focused effects. In many economic disciplines there exists no apriori rule to choose and validate a modell. So the mechanisms and dynamic effects explained by the model frequently are not or merely heavy accessible. On this basis economic models usually are formulated based on a qualitative or fuzzy knowledge of the underlying economic process. But, this results not by the lack of knowledge about the process—which could be gained by proper research, but on the the fundamental property of the dynamics based on human behavior.

Especially in the case of marketing the focus lies on the interaction between companies, brands and consumers. Here a proper formulation of the model is essential but the properties of the process are difficult to measure. So the results of the model just represent so-called stylized facts, which are based on the fuzziness of the human behavior and the human decision process. The model approach has to be chosen properly and the results have to be interpreted with care. The implication of the results to solve and optimize real-world problems cannot be used straight forward, but with care and always adapted to the specific question.

On the other hand there are also economic disciplines, as logistics and supply chain management, where the effects are well known and where fundamental properties of the model can be derived by comprehensible assumptions. So a stochastic analysis with proper assumptions regarding the variables, the distributions and the concavity of the used functions, provides reasonable results, which can be calibrated and evaluated using real-world data. Here the constraints are the assumption of the structural properties concerning the variables, which are mostly too strong to represent realistic effects. Based on this findings this study wants to provide a passably way to fuse both approaches to an integrated model.

In the following the basic integrated demand model will be presented and finally the extension with the disaggregated market model will be presented.

1 Integrated demand model

Traditionally in economic and operational models consumers are assumed as rational agents who make decisions based on current prices, income and market conditions. In a market with repeated interactions, customers' purchase decisions are also determined by past observed prices. Here an integrated demand model is focused, including the dynamic of the reference price into the classical demand model approach, where demand is just a function of the price. So, here the demand is a function of the actual price and all past prices. In the following two typical approaches to formulate the influence of the reference price on the demand are presented. The evolvement of the reference price over time in both cases is modeled using first order exponential smoothing. The first approach is just regarding the difference between the price and reference price, but the second equation additionally includes an asymmetry of the influence of a price higher than the reference price, and a price lower. The following equations show both assumed dynamics of the reference price:

$$rp_t = \alpha(rp_{t-1}) + (1-\alpha)(p_{t-1}) \tag{1}$$

$$\begin{aligned} rp_t = rp_{t-1} + \mu \max\{0, p_{t-1} - rp_{t-1}\} + \\ + \eta \min\{0, p_{t-1} - rp_{t-1}\}, \quad \mu \le \eta \end{aligned} \tag{2}$$

Which approach should be chosen depends on the complexity of the research question, in general the second one provides a more reasonable dynamics, as consumers are adopting much faster to positive facts - like lower prices, as to negative ones. It is further assumed that the demand does not depend on an absolute price level, but rather on its deviation from some reference level, which results from the pricing history. As mentioned above, in this model the demand depends on the price and the reference price. The formulation of the final demand model is shown in the following equation:

$$\begin{aligned} D_t(p_t, rp_t, \varepsilon_t) = \beta_0 + \beta_1 p_t + \beta_2 \max\{p_t - rp_t, 0\} + \\ + \beta_3 \min\{p_t - rp_t, 0\} + \varepsilon_t \end{aligned} \tag{3}$$

A study using this approach has been figured out separately (see Gimp-Heersink et al. (2006)). The aim of the model is to derivate optimal pricing strategies to maximize total expected discounted profit over the entire planning horizon. The objective function is given by the following equation:

$$\max_{p_t \in [\underline{p}, \bar{p}]} E\left[\sum_{t=1}^{T} \gamma^t (p_t - c_t) D_t(p_t, rp_t, \varepsilon_t)\right] \tag{4}$$

with $c_t \le p_t$ denoting the per unit ordering costs in periode t, $\gamma \in [0,1]$ the discount factor and T the total number of periods of the finite planning horizon. The optimal solution can be found using dynamic programming. So the Bellman equation of the dynamic program is given here:

$$\begin{aligned} v_t(rp_t) = \max_{p_t \in [\underline{p}, \bar{p}]} \{E[(p_t - c_t) D_t(p_t, rp_t, \varepsilon_t)] + \\ + \gamma E[v_{t+1}(rp_{t+1}(rp_t, p))]\} \end{aligned} \tag{5}$$

$$v_{T+1}(rp_t) = 0$$

Here optimal pricing strategies are derived just using an aggregated approach focusing on some pricing aspects like the reference price. The reference price is assumed to be equal for all consumers, due to the parameters μ and ν in equation (2) are assumed equal for all costumer. As can be seen the demand function has a stochastic characteristic, including not mentioned effects by the ε, which is assumed normal distributed with mean zero and variance σ^2.

2 Extension to a disaggregated marketing model

In this study some other disaggregated factors - like communication effects, especially positive/negative satisfaction and local word of mouth effects - have been included into the model. So, the aim has been to combine the optimization problem described above with a disaggregated model formulating the interactions of the consumers - mostly after the purchase process - to include more relevant and also local effects into the model. To keep the optimization problem simple and solveable, the structure of the optimization problem has be kepted, just the demand function has been extended by the consumers' satisfaction level, which depends on the difference between the real products' features and their actual perceptions (knowledge about the products' attributes). In this disaggregated model, information about the products' attributes can be exchanged between the customers at each time-step and so the perception of each consumer can be adapted depending to his neighbors. Knowledge about the products can only be obtained by an agent if he purchases the product – so only if the price is lower than the reference price. The knowledge about the products and the appropriate satisfaction level is spreading out from the "early adopters" with a higher reference price and therefore buying first.

At the beginning of the simulation each consumer has a randomly assigned reference price and his perceptions are zero. Both will evolve over time, the reference price as described in equation (2) and the perceptions by communicating with the neighbors (described below). Each product has an unique price $p_i(t)$, which has to be optimized using dynamic programming—and a remaining feature vector f_k. Here i refers to the consumer, j to theproduct and k to the attribute, att to the attitude and f to the real features of the product, which are initially not transparent to the consumer. At each time step the perceptions are exchanged between the neighbors as given by the following equation:

$$\Delta att_{ijk} = att_{ijk}^{t+1} - att_{ijk}^{t} = \frac{f_k - att_{ijk}^{t}}{2}$$
$$\Rightarrow \quad att_{ijk}^{t+1} = \frac{f_k + att_{ijk}^{t}}{2} \tag{6}$$

The interaction process has been kept as simple as possible, as the aim of the study was to demonstrate the interaction of the two modelling approaches. At each information exchange the agent communicates his value of perception to his neighbors. To communicate the perceptions to the neighbors following design of the agents' neighborhood has been defined (see figure 1). The agents environment equals the design of a cellular automata environment. The similarity of the concepts is used to implement the interactions of the agents using classical update-rules for cellular automata. The state space of each cell (or each agent) is defined by two variables. The third variable "Satisfaction" is a function of the other two (Reference price and perception level) and influences the choice process and therefore the demand. The implementation has been done object-oriented in Python and the optimization step using a dynamic programming algorithm has been done in Matlab. The lattice has a dimension of 40x50 cells and the neighborhood of the cells has been defined using the neighborhood definition of "Moore", where each cell has 8 neighbors (see figure 1).

3 Results

The problem cannot be solved for each consumer excluding the interaction process between the consumers. So the optimization and the information exchange process have to be calculated parallelly for each periode. Here the demand function at each timestep depends on the calculated evolvement of the reference price for each consumer (see equation. 2) and the considered interaction between adjacent agents. The simulation has been performed applying at each time step the update process with an information exchange between the agents and using the result

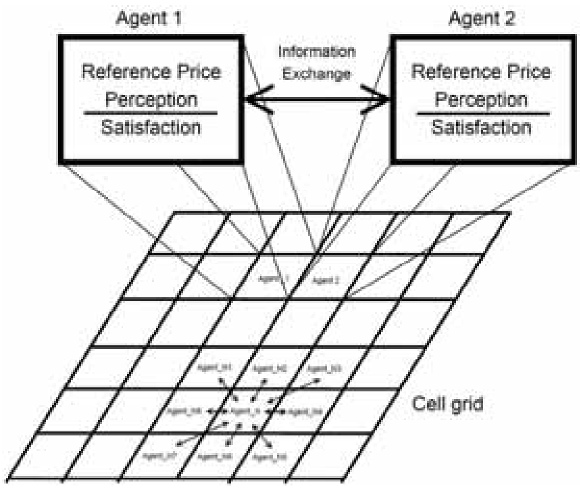


Figure 1. Neighborhood function and the interaction process between two agents.

as the demand function for the specific cell. To avoid too long calculation times, the optimization using the dynamic program has been done for an aggregation of 25 cells (5×5 squares). This does not restrict the concept of disaggregation, as this restriction could be dismissed immediately accepting longer calculation time. The results of this study have to be interpreted with care and differentiated – depending on the interpretation of the model assumption and the aggregation of the results. In figure (2) the heterogeneous states of the perception are shown at a certain timestep. The different perceptions levels are effecting different choice propabilities and therefore locally different demand. As mentioned above, the chosen modelling approach allows different interpretations, especially depending on how to the cell geometry and the resulting concept of the neighborhood are interpreted. If it is interpreted as a geographical configuration, the optimal prices will vary over certain regions and so also the price strategy should adapted to the different regions. If the neighborhood function is assumed as the probability of information exchange, the result can be interpreted as an existence of different focus groups with certain socio-demographic structures which should be addressed by different strategies. Such strategies can also be found in real businesses, but to provide realistic and validated recommendations, further model extensions and calibrations will be necessary.

Figure 3 shows the optimal price and inventory decision of a single cell. Here a proper aggregation is necessary, depending on the interpretation mentioned above and the willingness of a company to drive

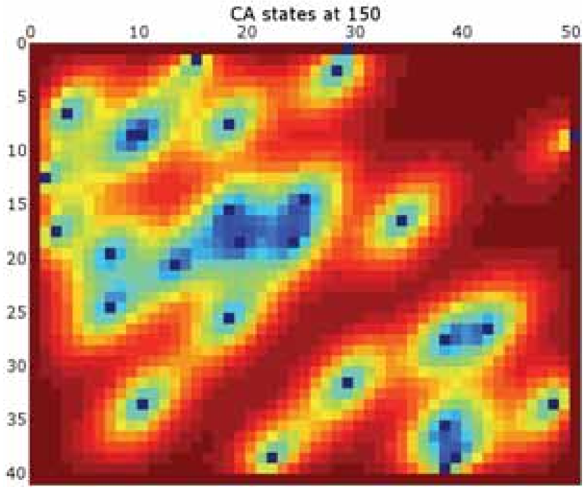


Figure 2. Disaggregated marketing effects (perception exchange, “word-of-mouth” effect)

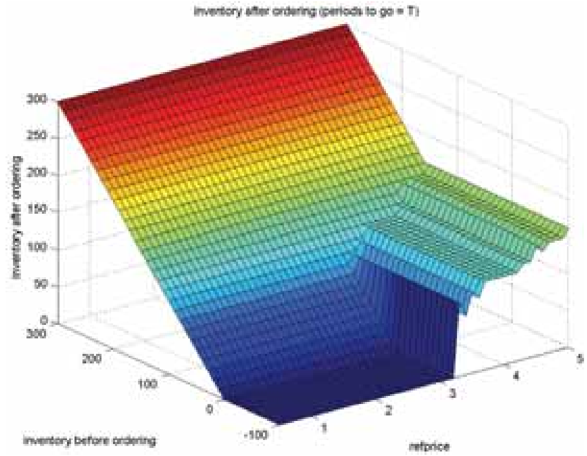


Figure 3. Optimization of the price and inventory decision inside a single cell

differentiated pricing strategies for each geographic region or socio-demographic group. But nevertheless, the study provides interesting effects and certain stylized facts.

References

[1] Bertsekas, Dimitri P., Steven E. Shreve. 1996. *Stochastic Optimal Control*: The Discrete-Time Case, vol. 2. Athena Scientific, Belmont, Massachusetts.

[2] Buchta, Christian, Josef Mazanec. 2001. *SIMSEG/ACM A Simulation Environment for Artificial Consumer Markets*. Working Paper 79, SFB - Adaptive Modelling: www.wu-wien.ac.at/am.

[3] Gimp-Heersink, L., J. Wöckl, A. Taudes. 2006. *Modelling reference price effects - a discrete dynamic programming approach*. Informs International. Hong Kong.

[4] Hauser, John R., Shugan Steven S. 1983. *Defensive Marketing Strategies*. Marketing Science 2(4) 319–360. Fall.

[5] Hruschka, Harald. 1996. *Marketing-Entscheidungen*. Vahlen.

[6] Kumar, K. Ravi, D. Sudharshan. 1988. *Defensive Marketing Strategies: an Equlibrium Analysis based on Decoupled Response Function Models*. Management Science 34(7) 805–815.

[7] Schuster U., J. Wöckl. 2005. *Optimal Defensive Strategies under Varying Consumer Distributional Patterns and Market Maturity*. Journal of Economics and Management 1(2) 187–206.

[8] Smith, James E., Kevin M. McCardle. 2002. *Structural properties of stochastic dynamic programs*. Operations Research 50(5) 796–809.

[9] Winer, Russel S. 1986. *A reference price model of brand choice for frequently purchased products*. Journal of Consumer Research 13 250–256.

[10] Wöckl, J. 2005. *Agent-based artificial consumer market and optimization of defensive strategies*. P. Baillie-de Byl, Rimane R., eds., Proceedings of the 6th Workshop on Agent-Based Simulation. 34–37. Erlangen.

Corresponding author: Jürgen Wöckl,
Dept. of Production Management, Vienna Univ. of Economics and Business Administration
Augasse 2-6, 1090 Vienna, Austria
Juergen.Woeckl@wu-wien.ac.at

Accepted OSIM 2006, June 2006
Received: October 10, 2006
Revised: June 20, 2007
Accepted: June 25, 2007

ARGESIM BENCHMARKS

A Graphical Modelling Approach to ARGESIM Benchmark C3 'Generalized Class-E Amplifier' with Simplorer

Matthias Auchmann, Moritz Gschwandtner, Vienna Univ. of Technology, Austria

{e0125454, e0125439}@student.tuwien.ac.at

Simulator: Simplorer, developed by ANSOFT, offers a way of modelling and simulation which is quite different from the common known block oriented modelling. The Simplorer simulation software adheres to the VHDL – AMS standard, therefore one can use the predefined modelling libraries and draw schematics in the graphical interface, it is not necessary to implement the component's behaviour manually. Furthermore optimization and statistical analysis are offered.

It is also possible to enter initial conditions for the circuit, which is used in the second part of the exercise. The questions defined in ARGESIM Benchmark 3 are all solved using Simplorer SV, the student version of the software. Thus we cannot assure that features absent in this version are not included in the professional version. In Figure 1 the schematic of the problem defined in benchmark 3 as implemented in Simplorer is shown:

The circuit below is equivalent to the following equations:

$$dx1/dt = (-x2 + VDC)/L1$$

$$dx2/dt = (x1 - x2/R(t) - x3)/C2$$

$$dx3/dt = (x2 - RL * x3 - x4)/L3$$

$$dx4/dt = x3/C4$$

Model: Even if drawing the schematic in Simplorer is quite easy, the ease of this way of model implementation comes at the prize of less flexibility. Anyway, we succeed in doing the modelling correctly, even if we have to come to a compromise in some parts of the model.

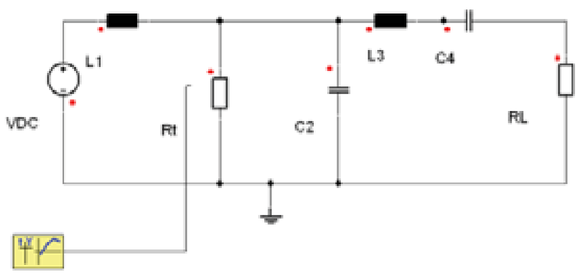


Figure 1. The Simplorer model of the amplifier with standard components

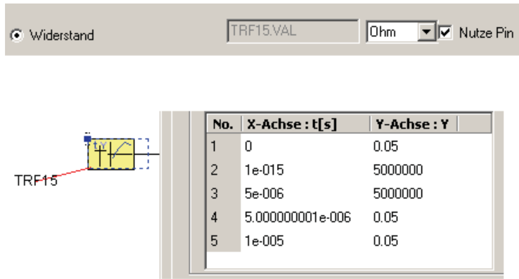


Figure 2. Modelling of the tome dependent resistor $R(t)$ using a 2D table

First we implement the switching resistor $R(t)$, a key part of the model. The variable resistors value can be defined by its input pin, which we connect to a 2D value table. This table specifies the periodic switching function. The following screenshot (Figure 2) shows these setups. In addition, the 2D value table has to be switched to periodic, which is not shown in the picture.

The other components of the model are configurated by a right-click on the icon; choose *Properties* from the context menu and then enter the components parameters, like voltage, resistance, capacitance or inductance.

We have to make sure that the polarity of the components is chosen correctly. One has to keep in mind that the way of modelling used in this approach is physical component oriented modelling and simulation and therefore for example reservoirs like a capacitor have a defined polarity.

The context menu of each component has a *Rotate* and *Mirror* entry which can be used for this purpose.

SNE 18/1, April 2008

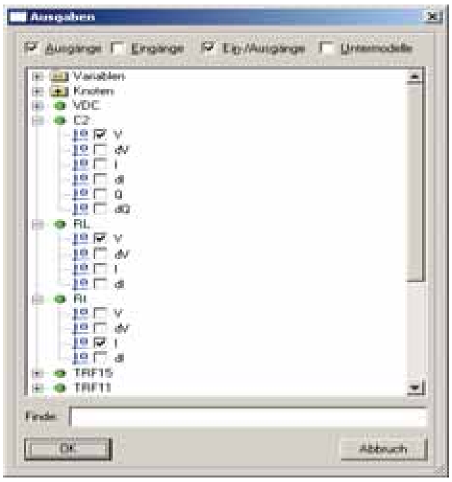


Figure 3. Interface of value selection for the plot.The selected values can then be plotted together in one plot with different colors.

One of the aforementioned compromises is that we do not succeed in getting one plot with all the rise/fall times. For that reason we use four 2D value tables, one for each *TRF* and connect them one after the other.

Now we want to describe briefly how we obtained the plots. Simplorer has a widget called 2DGraphSel. One can easily add different values to be plotted, see Figure 3.

A-Task: This task cannot be realized with the used version of Simplorer.

B-Task: In this task simulation of the system over the time interval $[0,100\mu]$ sec with the zero – solution as initial state was calculated. We are interested in the time dependent values of $IR(t) = x2 / R(t)$ and $VL = x3 * RL$. The graphical interface offers easy handling and plotting of calculated data after simulation. The results are shown in Figure 4, whereby the red curve depicts $IR(t)$ and the black one shows the voltage at RL in Volt.

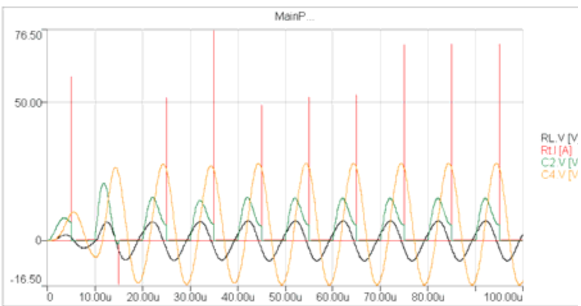


Figure 4. Solution of task B

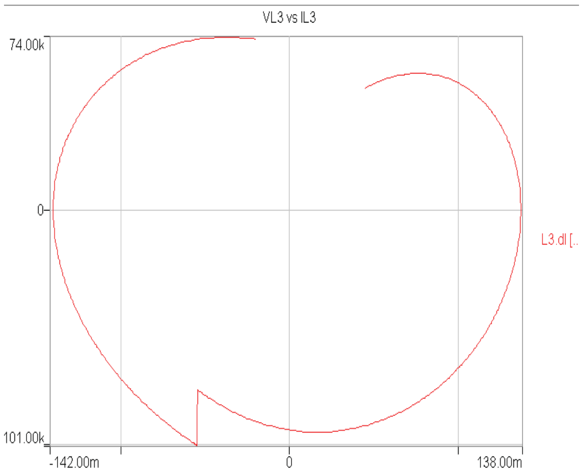


Figure 5. Resulting phase plot of the voltage at coil $L3$ and the current at the inductor

C-Task: Doing the phase plots for the last part of the exercise can be performed in a similar manner. One just has to choose another variable for the x-axis, so that time is not used. For the y-axis one can easily choose the relevant derivate. In addition, the initial values can be changed for the simulation using the graphical modelling interface of each component.

The results of one setting as defined in task C of the benchmark can be seen in Figure 5. A convenient way to add all of the phase plots into one graphical representation was not found; instead we had to do one after the other.

Resumé: Simplorer is a tool that offers intuitive modelling of physical systems. Although features like eigenvalue calculation are missing in the student version, the product is easy to handle.

The calculated results fit the results given by the sample solution of the comparison. The reinitializations in task C were done manually. No script files were used.

Corresponding author: Matthias Auchmann,
Department of Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
e0125454@student.tuwien.ac.at

Received: June 28, 2007
Revised: October 10, 2007
Revised: January 14, 2008
Accepted: January 25, 2008

An Object-oriented Solution to ARGESIM Benchmark C4 ‘Dining Philosophers Problem’ implemented with AnyLogic

Michael Gyimesi, Andreas Dielacher, Thomas Handl, Christian Widtmann,
Vienna University of Technology, Austria, *mgyimesi@osiris.tuwien.ac.at*

Simulator: AnyLogic is a JAVA based general purpose simulator supporting System Dynamics, agent based and discrete event modelling approaches under the object-oriented model design paradigm. Users may apply the concepts of objects, interfaces, hierarchy, message passing and more to generate their own modelling constructs.

AnyLogic supports a large number of pre-defined probability distributions for stochastic modelling and can be used to design deterministic models as well. Analyses of output data can be performed and visualized directly within the simulator. AnyLogic enables the user to generate interactive 2D and 3D animations of the simulation as well as portable web-enabled models for use as web applets. The model described in this article was created in AnyLogic Version 5.5.

Model: This comparison C4 describes the problem of five philosophers competing for five chopsticks on a shared table. The used model is just as classical, since it features the philosophers, the chopsticks and the table modelled in one object class, respectively.

The table is the top object, instancing the five philosophers and the five chopsticks. In this object oriented approach, each philosopher is connected to each of the two adjacent chopsticks by one message queue. The normal procedure is requesting at first the left and then the right chopstick, which in return are granted to the philosopher by an according message. The philosopher’s two requests for the chopsticks are sent with a relative delay (pause). The duration of the activities *thinking*, *pause* and *eating* is distributed uniformly across a certain range of values. The philosophers competing for an insufficient number of chopsticks can be seen as a metaphor on an arbitrary distributed system whose components compete for limited shared resources.

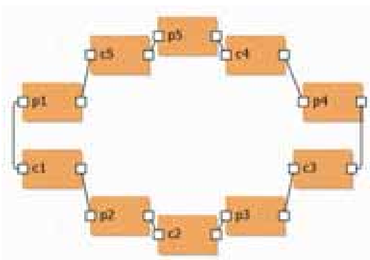


Figure 1: Top-Level Model

So far the model resembles the example included with AnyLogic. An expansion was made, building a hybrid model since the philosophers maintain a certain level of “saturation”, which increases or decreases according to a simple differential equation while the philosophers are eating or not eating. We discarded this hybrid approach in favor of a purely discrete model by replacing the continuous “saturation” function of the example model by a discrete “calories” function, described in the context of task b.

In the original model, philosophers can reach a deadlock situation when for example all philosophers simultaneously request their respective left chopstick, causing them to wait forever for their respective right chopstick.

A-Task: A deadlock occurs when all five philosophers, having requested and received the left chopstick, simultaneously enter the *pause* state with no right chopstick available. Thus a timeout has been introduced, indicated by the edge leading from *waitingRight* back to *thinking*, after which a philosopher returns the left chopstick and thereby avoids a deadlock for the whole system. The parameter *maxPause* which influences the time between the left and the right request has a significant impact on the likelihood of a deadlock because it controls the amount of time available for the adjacent philosopher to seize the right chopstick.

Various simulation runs under variation of *maxPause*, left and right timeout illustrated that a high maximum pause has a crucial impact on the blocking of one chopstick and that the consecutive high waiting times result in easy starvation. This can be alleviated by using high timeout values particularly while in *waitingRight*, since this value has a direct impact on the time that the left chopstick is seized without actually being used. The timeout value for waiting for the left chopstick directly affects the probability that an available chopstick is going to be seized. The sum of

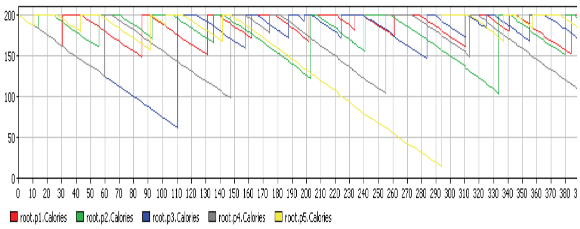


Figure 2: Results for symmetric timeouts for waitingLeft and waitingRight

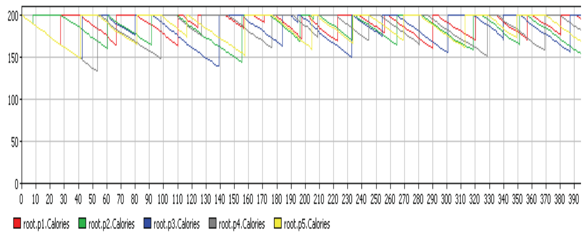


Figure 3: Results for asymmetric timeouts for waitingLeft and waitingRight

both timeouts and *maxPause* affects the fraction of time that is available for actually doing work. Generally speaking, high timeout values favor quick and efficient use of chopsticks, keeping in mind that this is lost for doing actual work. After adding timeouts, *maxPause* is only relevant for the number of calories consumed while inactively waiting to seize the right chopstick.

The results in figure 2 and 3 reflect the caloric distribution for symmetric timeouts of 10 for waiting for both chopsticks as well as for asymmetric timeouts of 2 for the left and 50 for the right chopstick, with 200 calories and the value of 20 for all max values. It can be clearly seen that the asymmetric variant shows better caloric characteristics.

B-Task: Figure 4 depicts a philosopher's state machine incorporating all elements for modelling task a and b.

A philosopher always leaves the *eating* state with the full number of calories while in the original approach, he didn't necessarily have to eat until full saturation. This adaptation resembles that once the critical section is reached, a component will only leave the section again when it has fulfilled its task whose deadline will then be fully restarted. The duration of remaining in the *eating* state is still randomly distributed, since the duration of the critical section may not be constant but e.g. data dependent. The notion of calories thus resembles the timeout that each component (or philosopher) is able to wait until he is required to enter the critical section. When calories reaches zero (i.e. the component misses its deadline for the critical section), the philosopher enters the state *dead* and is no longer participating in the dinner i.e. the system, indicating a potential system failure.

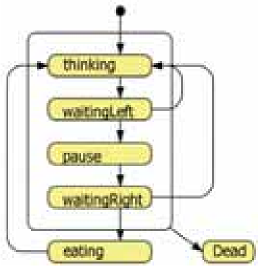


Figure 4: Philosophical behaviour

Thus care has to be taken to initialize a philosopher with a reasonable number of calories with regard to the maximum thinking, pause and eating periods. In our experiments, initial 200 calories turned out to be adequate for maximum thinking and eating periods of 20 time units each, with varying maximum pause and timeout values. Despite being a discrete variable, the used version of AnyLogic did only accept equations being attributed to a state rather than program statements. Thus the discrete decrease of calories had to be modelled as a first order derivate with a constant value of -1.

C-Task: An important aspect of a distributed system is the amount of time that it can spend on its actual task. Thus the timeout value used for the edge from *waitingLeft* to *thinking* is significant for this ratio. When the timeout is too long, much time is wasted waiting for a chopstick. If it is unreasonably short, the philosopher will quickly enter the state *thinking* again, spend the next period working and thus risk running out of calories.

Resumé: The modeling approach used in this solution is an object-oriented process approach. AnyLogic provides a powerful environment with different well suited possibilities for building models of distributed parallel systems and performing experiments with different settings.

Corresponding author: Michael Gyimesi,
Department of Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
mgyimesi@osiris.tuwien.ac.at

Received: June 28, 2007
Revised: February 10, 2008
Accepted: February 20, 2008

Comparing ODE Solvers for to ARGESIM Benchmark C5 'Two State Model' using MATLAB

Florian Judex, Florian Plug, Ido Yehieli, Vienna University of Technology, Austria, *efelo@fsmat.at*

Simulator: Matlab is a well-known general-purpose mathematical programming language. It offers a wealth of predefined functions especially suited for implementing numerical algorithms. One of those is `ode15s`, which implements a BDF method for computing numerical solutions of stiff ordinary differential equations and supports event detection/location. We compared solutions of C5 obtained with `ode15s` with those obtained using a handwritten solver using an implicit Runge-Kutta scheme based on Gauss-Legendre quadrature and bisection (for event location). Since C5 is easily solved analytically (either manually, or using a computer algebra system like Maple), we also compared those solutions to an analytically obtained solution to judge their accuracy.

Model: Comparison 5 is a simple system of two linear ordinary differential equations, given by

$$\dot{y} = \begin{pmatrix} -c_1 & c_1 \\ 0 & -c_3 \end{pmatrix} \cdot y + \begin{pmatrix} c_1 \cdot c_2 \\ c_3 \cdot c_4 \end{pmatrix} \qquad (1)$$

Parameters c_2 and c_4 depend on the current state of the System (which can either be A or B), while c_1 and c_3 are kept constant over the whole integration interval. The state is determined by two bounds Y_A and Y_B – whenever y_1 grows larger than Y_A in state A, the system switches to state B, and conversely when y_1 becomes smaller than Y_B in state B the system switches back to state A. The interesting cases arise for values of c_2 and c_4 which produce periodic state changes between A and B, and values of c_1 and c_3 which turn the system into a stiff one.

`ode15s` operates on systems of ODEs given in the form $\dot{y}(t) = f(t, y(t))$ together with zero or more event functions $e_i(t, y)$. While integrating, the algorithms monitors the event functions, and locates their zeros-crossings. Two flags per event function specify if only rising, only falling, or both directions of zero-crossing shall be considered an event, and if the integration is to be continued or stopped upon detecting an event. The step size is controlled by a local error target passed to the algorithms. For more efficient and accurate operation, the jacobian $\partial f / \partial y$ of $f(y,t)$ can be specified.

A-Task: Comparison 5 is easily brought into the form required by `ode15s`. The definitions of $f(y,t)$ and $\partial f / \partial y$ of are obvious, and for state-change detection the two event functions $e_A(t,y) = y_1 - Y_A$ and $e_B(t,y) = Y_B - y_1$ can be used – both set to consider only rising zero-crossings. Because the description of `ode15s` is not entirely clear on how the algorithm continues after a state change, both functions were set to stop integration upon event detection, and `ode15s` was called in a loop until the whole integration interval was covered.

The implemented algorithm treats state changes differently – it requires the system of ODEs to be given as $(\dot{y}(t), s(t)) = f(t, y(t), s(t), p)$, with $f(t,y,s,p)$ linear in y (It could be extended to support nonlinear systems rather easily, though) and $s(t)$ specifying the current state of the system. The value for p is specified when calling the solver algorithm, and it passed down to the individual evaluations f. This allows parameterization of the system of ODEs without resorting to global variables. $f(t,y,s,p)$ must not only calculate the derivative of y, but also the new state of the system. Whenever the returned value of $s(t)$ differs from the one that was passed to $f(t,y,s,p)$, the algorithms treats this as a state event. The algorithm takes another argument specifying a global error target that it tries to meet.

The author's algorithm integrates along the integration interval, at each step controlling the step size by computing a local error estimate and comparing it to some local error target. When it encounters a state change event ($s(t)$ changes), it locates the precise time of the state change using bisection. It then recomputes the solution starting from the last state change (or the start of the integration interval) on a twice as fine grid, and computes a global error estimate by comparing the two solutions. If this estimate meets the requested global error target, it proceeds by restarting the integration using the last computed $y(t)$ and $s(t)$ as the new initial values. If the estimate doesn't meet the global error target, the integration restarts at the last state change, with a suitably reduced local error target.

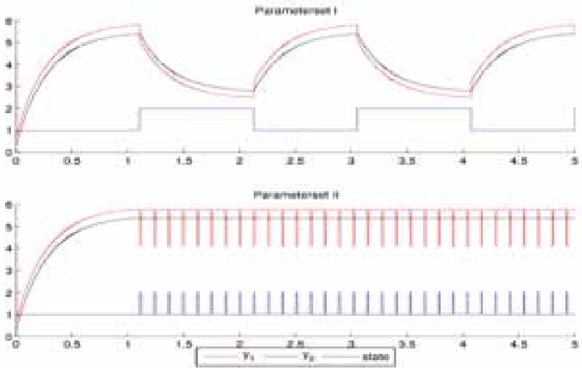


Figure 1. Plot of the solution

Each step is computed using a fully implicit 4-state Runge-Kutta scheme to accommodate the stiffness of the problem. The coefficients for that 4-state IRK scheme were obtained by Gauss-Legendre quadrature, and therefore yield an 8th order scheme.

Figure 1 shows the results obtained with the implemented solver.

B-Task: For better results, the analytical solution was used to compute the zero Crossings given in Table 1.

	Analytical
t_1	1.1083061677711285586
t_2	2.1296853551547112460
t_3	3.0541529069957142895
t_4	4.0755320943792971988
t_5	4.9999996462203002423
$y_1(5)$	5.3693121180964613615
$y_2(5)$	5.3999967644598712013

Table 1. Zero Crossings

C-Task: Since only our handwritten solver supports global error estimation, the accuracy was interpreted as the global error target for our handwritten solver, but at the local error target for `ode15s`.

Table 2 shows the results produced by `do_test` for three different accuracies (10^{-6}, 10^{-10} and 10^{-14}).

D-Task: Using the same test setup as for the tasks b and c also the results for the other set of parameters werce computed, yielding a much higher oscillation frequency. The results of this experiment can be found in Table 3.

	10^{-6}		10^{-10}		10^{-14}	
	IRK (rel. err.)	ode15s (rel.err.)	IRK (rel. err.)	ode15s (rel.err.)	IRK (rel. err.)	ode15s (rel.err.)
t_1	4e-14	2e-9	4e-15	2e-9	7e-15	2e-9
t_2	2e-14	7e-7	1e-15	4e-9	2e-15	4e-9
t_3	6e-15	5e-7	3e-16	6e-9	2e-16	6e-9
t_4	5e-15	7e-7	1e-15	6e-9	2e-16	6e-9
t_5	4e-15	6e-7	1e-15	7e-9	5e-16	7e-9
$y_1(5)$	4e-9	5e-2	9e-10	4e-3	4e-10	4e-3
$y_2(5)$	3e-14	5e-6	1e-14	6e-8	5e-15	7e-9

Table 2: Accuracy of the Algorithms

		10^{-11}	
	Analytical	IRK (rel. err.)	ode15s (rel.err.)
t_1	1.108306167	1e-14	2e-9
t_2	1.121729967	1e-14	2e-7
t_{n-2}	4.809306109	5e-14	2e-6
t_{n-1}	4.923040107	5e-14	2e-6
t_n	4.936463907	5e-14	2e-6
$Y_1(5)$	5.780402520	2e-14	6e-7
$Y_2(5)$	5.380402678	2e-14	9e-7

Table 3: Results for Task D

These results show a similar relationship between the errors produces by ode15s compared to those of our handwritten algorithmn as tasks b and c. Since the last state change (t_N in the results table) lays further away from 5 for this set of parameters, $\dot{y}$ returns to a smaller value before reaching the end of the integration interval. This eliminates the large difference between the relative errors of $y_1(5)$ and $y_2(5)$.

Resumé: The implemented IRK algorithm fulfilled the authors expectations fully. It only remains to be implemented according to the MATLAB standard for ODE solvers.

Corresponding author: Florian Judex,
Department of Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
efelo@fsmat.at

Received: June 28, 2007
Revised: November 17, 2007
Revised: March 3, 2008
Accepted: March 10, 2008

ASIM - Buchreihen / ASIM Book Series

Fortschritte in der Simulationstechnik (FS) / Series Frontiers in Simulation (FS) - Monographs, Proceedings:

W. Borutzky: *Bond Graphs Methodology for Modelling Multidisciplinary Dynamic Systems*. FS 14, ISBN 3-936150-33-8, 2005.

M. Becker, H. Szczerbicka (eds.): *19th Symposium Simulation Techniques*. Proceedings Tagung ASIM 2006, Hannover; FS 16, ISBN 3-936150-49-4, 2006.

S. Wenzel (Hrsg.): *12. Fachtagung Simulation in Produktion und Logistik*. Proceedings Tagung ASIM SPL 2006; ISBN 3-936150-48-6, 2006.

F. Hülsemann, M. Kowarschik; U. Rüde: *18th Symposium Simulation Techniques*. Proceedings Tagung ASIM 2005 Erlangen; FS 15, ISBN 3-936150-41-, 2005.

Available / Verfügbar: SCS Publishing House e.V., Erlangen, WWW.SCS-PUBLISHINGHOUSE.DE
Download ASIM Website WWW.ASIM-GI.ORG (partly; for ASIM members)

Fortschrittsberichte Simulation (FB) / Advances Simulation (AS) / ASIM Mitteilung (AM) ARGESIM Reports (AR) - Special Monographs, PhD Theses, Workshop Proceedings

C. Deatcu, S. Pawletta, T. Pawletta (eds.): *Modelling, Control and Simulation in Automotive and Process Automation*. Proceedings ASIM Workshop Wismar 2006, ARGESIM Report 31, AM 101; ISBN 3-901-608-31-1, 2006.

H. Ecker: *Suppression of Self-excited Vibrations in Mechanical Systems by Parametric Stiffness Excitation*. ARGESIM Report FB 11, ISBN 3-901-608-61-3, 2006.

M. Gyimesi: *Simulation Service Providing als Webservice zur Simulation Diskreter Prozesse*. ARGESIM Report FB 13, ISBN 3-901-608-63-X, 2006.

J. Wöckl: *Hybrider Modellbildungszugang für biologische Abwasserreinigungsprozesse*. ARGESIM Report FB 14, ISBN 3-901608-64-8, 2006.

Th. Löscher: *Optimisation of Scheduling Problems Based on Timed Petri Nets*. ARGESIM Report Vol. 15, ASIM / ARGESIM Vienna, 2007; ISBN 978-3-901608-65-0.

Available / Verfügbar: ARGESIM/ASIM Publisher, TU Vienna, WWW.ARGESIM.ORG
Download / Bestellung zum Mitgliederpreis € 10.- ASIM Website WWW.ASIM-GI.ORG

Reihen der ASIM-Fachgruppen / Series of ASIM Working Groups

S. Collisi-Böhmer, O. Rose, K. Weiß, S. Wenzel (Hrsg.): *Qualitätskriterien für die Simulation in Produktion und Logistik*. AMB 102, Springer, Heidelberg, 2006; ISBN 3-540-35272-4.

M. Rabe, S. Spiekermann, S. Wenzel (Hrsg.): *Verifikation und Validierung für die Simulation in Produktion und Logistik*. AMB 103, Springer, Heidelberg, 2006; ISBN 3-540-35281-3.

J. Wittmann, M. Müller (Hrsg.): *Simulation in Umwelt- und Geowissenschaften - Workshop Leipzig 2006*. Shaker Verlag, Aachen 2006, AM 106; ISBN 978-3-8322-5132-1.

A. Gnauck (Hrsg.): *Modellierung und Simulation von Ökosystemen - Workshop Kölpinsee 2006*. Shaker Verlag, Aachen 2007, AM 107; ISBN 978-3-8322-6058-3.

Available / Verfügbar: Bookstore / Buchhandlung, ermäßigter Bezug für ASIM Mitglieder
Info at ASIM webite WWW.ASIM-GI.ORG

A DEV Approach to ARGESIM Benchmark C6 'Emergency Department' using Enterprise Dynamics

Thomas Löscher, Joachim Aichinger, Wolfgang Ebner, TU Wien, Austria

thomas@loescher.at, e0025538@student.tuwien.ac.at, e0225538@student.tuwien.ac.at

Simulator: Enterprise Dynamics (ED) from Incontrol Enterprise Dynamics is a software program for discrete event simulation. Furthermore, Enterprise Dynamics is an object-oriented software program for modelling, simulation, visualization, and control of dynamic processes. The users can pick up elements—called atoms—from standard libraries in order to build their own model. ED is based on this concept of atoms as modelling objects in each model. An atom can represent a machine, a counter or a product but can also have a non-physical character like a graph. Thanks to the open structure of ED, the advanced user can build and use its own atoms, for example to model a machine with very specific characteristics. At this point, ED includes 100 standard atoms, but this number is ever increasing. The beginner may typically only need to select from around 30 frequently used atoms to have enough material for his or her applications. Atoms thus are pre-defined modelling objects used to build models quickly and to carry out studies. ED also has a built-in programming language called 4DScript, which can be used for processing specific conditions from reality in the model.

Model: Casualties from accidents are admitted to an emergency department for dressing of wounds. Broken limbs are put in plaster. After a few days a follow-up examination must be performed to monitor the healing process. If necessary, additional treatment will be administered. Follow-up treatment in the emergency department of a hospital is the discrete process to be investigated in this comparison.

In this case a process orientet approach is chosen whereas all treatment points for the patients are modeled by the use of server and queue atoms. Figure 1 shows a screenshot of the model containing the used atoms and their connections. The patient's way starts in the Source atom (S) where they are created. To define the certain types of patients an empirical distribution is used on creation in the Source atom. The possible ways through the system are represented by the atom connections and they are more or less predefined for each type at the beginning. Next follows the registration (R) which is built up by one queue and one server. In the registration server the patients will be admitted to the two casualty wards (CW) with the distribution given in the definition of comparison 6. Each casuality ward is modeled by one queue and two servers respresenting the waiting room and the two doctors. Also for plaster (P) and x-ray (X) rooms servers and queues are used. After treatment the patients are leaving the emergeny department through the sink atom (Sk). To consider that the doctors will start half an hour later a time schedule availability control (T) has to be placed.

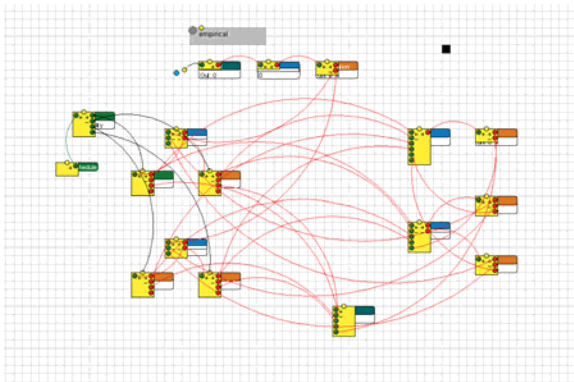


Figure 1: Model with Connections

In addition to the correct linking it is also necessary to program the atoms with certain queries. Several labels are set and used to realize the correct routing of the patients. For example the label called (xray) is defined at the beginning and initialized with the value 1. This label is necessary to know if the patients have already been to the x-ray station. Because after entering this station the value of the label will be increased to 2. So if the patients will be sent to casualty ward again the query of the doctor server will notice the increased value and therefore the output will be controlled.

Such query can look like that:

```
{ If already been to x-ray (xray=2) send to
output 3 (sink) }
if(ddb([xray],rank(1,c))=2,3,...
```

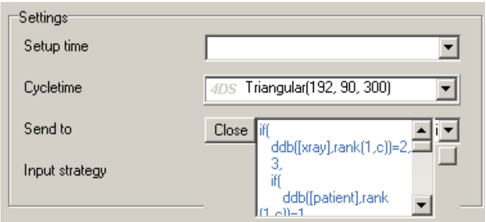


Figure 2: Settings for doctor server

Figure 2 shows the screenshot of the parameter window for the doctor server atom. The "send to" parameter is changed depending on the different label values and the patient is routed to the next station corresponding to the number of the output channel.

A-Task: A single simulation run is proceeded in task a. This means that the simulation runs until the stopping criterion is reached. Figure 3 shows the correct setting for the trigger on entry parameter of the sink for 250 patients. If the last patient is leaving the emergency department through the sink after treatment the system directly stops the simulation.

Figure 3: Trigger on entry – Sink

Depending on the type of patient the treatment time takes between 94 and 220 minutes (further results see Table 1).

B-Task: As soon as the queue before causality ward 2 contains more than 20 patients the more experienced doctor should take over. Each doctor is modelled as separate server and therefore the physical exchange can be easily implemented by the change of the parameters of the used distribution functions. To change the doc-tor from an unexperienced to an experienced one the following expression must be set on the cycle time parameter of the doctor server representing the unex-perienced doctor (see Figure 2):

```
if(
   Content(Rank(7, Model))>20,
   Triangular(246, 168, 378),
   Triangular(192, 90, 300))
```

The simulation is controlled as in task A and the results and effects can be seen in table 1. Because of the 60 % admission of patients to casualty ward 1, task B (changing Doctors) is not an effective way to lower

mean time	task a	task b	task c
patient 1	274.2	262.2	185.8
patient 2	164.7	182.5	202.1
patient 3	290.5	297.9	233.1
patient 4	154.0	146.3	193.3
standard deriv.	93.3	94.5	92.5
overall treatm. time	372	382	367
close hour	14:12	14:22	14:07

Table 1: Results for task a-c: mean for

the overall treatment time even the mean treatment time of patient 3 is very high.

C-Task: On creation a consecutive number is assigned as label id to the patient atom. In the queues of the casualty wards only patients are arriving with consecutive numbers. Therefore a priority ranking can be established in the casualty ward queues. If a patient is set back from x-ray or plaster with a lower consecutive number he will be ranked in the queue corresponding to his label id. The standard queuing strategy FIFO (first in first out) usually represents the consecutive sequence. On this account a queuing strategy can be used which is based on the consecutive numbers. To do a priority ranking in the casualty ward queues the queue discipline parameter must be set in the following way:

```
t-findqueuepos([id],2)
```

Sort by Label Ascending: The atoms with the lowest value of the label named id are put in front.

But task c (priority ranking) can reduce treatment time a bit, in average of about 289 seconds and also all types of patients have more or less the same treatment time. The results (table 1) show a decrease in treatment time for patients of type 1 and 3 and an increase for the two other types of patients. The standard deviation of the overall treatment time decreases.

Resumé: This work contains a classical event oriented modelling approach using the simulator Enterprise Dynamics.

Corresponding author: Thomas Löscher,
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
thomas@loescher.at

Received: June 28, 2007
Revised: November 12, 2007
Accepted: November 20, 2007

SNE 18/1, December 2007

Statechart Modelling for ARGESIM Benchmark C10 'Dining Philosophers Problem II' using Simulink/Stateflow

Voin Legourski, Yilin Huang, Ondrej Cevan, F. Breitenecker, Vienna Univ. of Technology, Austria

Simulator: Simulink is a MATLAB extension allowing rapid and accurate building of computer models of dynamical systems using block diagram notation. With Simulink one can model complex nonlinear systems using continuous and discrete-time components. Particularly important for our comparison is the state-flow extension to Simulink. Stateflow provides a powerful environment with which one can add finite state machines into the Simulink models. It is build around the state-chart formalism.

Model: The system described by the 'Dining philosophers' Problem" does consist of five philosophers who are sitting at a round table, on the table in front of each of them is located a bowl of food. To eat from the bowl a philosopher needs two chopsticks. The problem lies within the sticks, as there are only five sticks available to the philosophers– one between each bowl, two neighbors cannot eat at the same time. The philosophers can be in 3 resp. 4 states: thinking, eating and waiting (waiting for left and right stick; Figure 1- Stateflow model).

If any of the philosophers wants to grab a chopstick a request needs to be sent to the chopstick subsystem in order to check whether the chopstick is available. A chopstick subsystem controls use of the chopsticks, with two inputs and two outputs. The inputs are function calls that trigger two function-call subsystems: *chopstickR* and *chopstickL* (Figure 2). The outputs signal whether the chopstick is available for one of the two philosophers that compete for the chopstick.

The subsystems (*chopstickR/L*) are modeled by a state-chart and if a philosopher calls the subsystem once, the state-chart is activated and checks whether the subsystem is already using the chopstick. The calling philosopher will receive a "ticket" for the chopstick in case the condition returns true, which means that the chopstick can be used. If the philosopher calls the subsystem the second time (while using the stick) the condition *[havingChopstick]* returns the ticket for the chopstick.

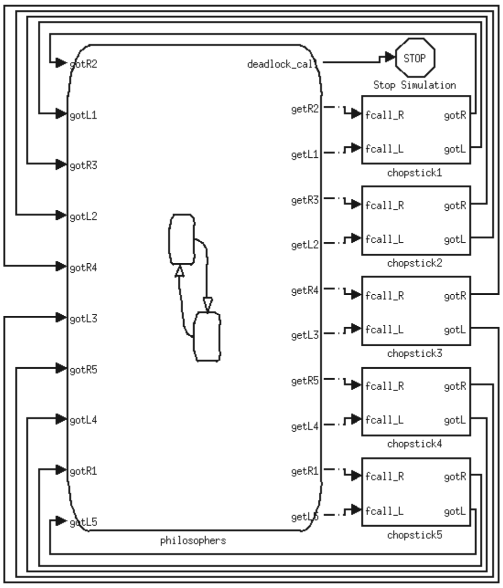


Figure 1. "Dining Philosophers" model, top level

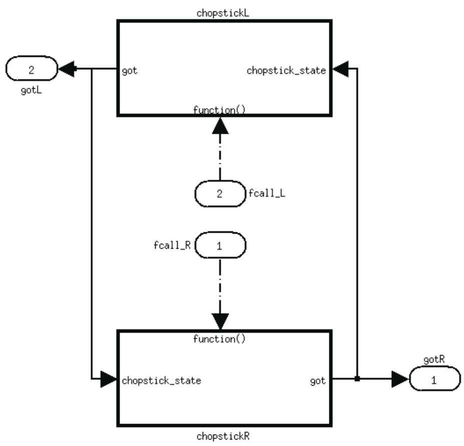


Figure 2. Function call subsystems of a chopstick

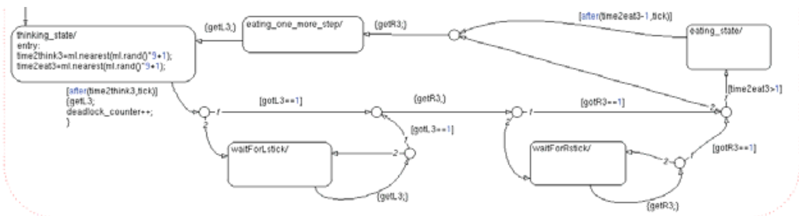


Figure 3. Stateflow diagram of philosopher 3

The philosophers' state-chart consists of one graphical function and one super state called philosophers and which contains six parallel (AND) states (Figure 3). These are independent and can be active at the same time. Five of these states represent philosophers and the sixth one has a deadlock monitoring function.

Although parallel states execute concurrently, they are not activated at the same time. State-flow determines when to activate them during simulation according to priorities. This means, the states are activated in a defined order which can be configured by the user. This feature was used to solve the conflict of simultaneous access. Here the priorities were set in such a way that philosophers to the right side of a stick have higher priority than those to the left of it.

Because the philosophers are sitting in a ring (*philosopher1-philosopher2-philosopher3-philosopher4-philosopher5-philosopher1*), the *philosopher5* has lower priority then *philosopher1* and still needs to win the access to its left chopstick. This was solved by letting *philospher5* pick up its left chopstick one time step before entering "waiting for left stick" state. The same consideration leads us to a very similar solution when our philosophers return their chopsticks.

With this way of assigning and returning chopsticks to/from philosophers we created a model that needs 0 time steps for "housekeeping" in which it is possible to receive both chopsticks during one transition (from "thinking" into the "eating" state) – if both sticks are available. Eating state is divided into two states: *eating_state* and *eating_one_more_step*. Entering and leaving a state takes at least one time step.

A**-Task:** In order to evaluate the model, we needed to add some more variables into the state-flow charts for measuring time to wait and to export the data to the MATLAB workspace. The chopstick utilization was evaluated with an extra state-flow chart which simply counted the number of time steps a chopstick was used. The simulation was started by MATLAB and stopped if a deadlock was. The simulation ran for 5 356 896 time steps. MATLAB evaluation results in Table 1 and Table 2.

B**-Task:** In order to perform this task as fast as possible we removed from the model commands needed for the evaluation of time to wait and chopstick utilization. The simulations were executed with a simple FOR loop from the MATLAB environment.

state	p1	p2	p3	p4	p5	all
thinking	5.4935 2.6344	5.5095 2.6300	5.4968 2.6350	5.4961 2.6276	5.5084 2.6306	5.5008 2.6315
waiting	11.7189 7.8350	11.7295 7.8203	11.7178 7.8299	11.7276 7.8398	11.7213 7.8277	11.7230 7.8305
eating	5.5118 2.6265	5.4954 2.6252	5.5045 2.6280	5.4900 2.6303	5.5053 2.6321	5.5014 2.6284

Table 1: Philosophers times in respective states (mean and standard deviation)

c1	c2	c3	c4	c5	all
0.9245	0.9245	0.9241	0.9244	0.9243	0.9244

Table 2: Chopstick utilization (mean)

Before the start, the *time2deadlock* variable was initialized. The resulting times are exported into MATLAB workspace. After performing 50 runs the maximum termination time lay with 21,788,589 time steps, the minimum with 68,994 and the average simulation took 4,666,830 time steps.

A deadlock state can be reached only if all philosophers decide to leave the thinking state at the same time step, and thus grab their left chopsticks concurrently. To detect this, every time a philosopher leaves thinking state the global variable *deadlock_counter* is increased by one. This variable is checked at the end of each time step by the state *deadlock_checking* (the sixth (AND) state in the philosophers super state). If equaling less than five, the value of the variable is reset to 0. In the case all the philosophers increased the variable, *deadlock_counter* equals 5 (equivalent to a deadlock). Thus the simulation is terminated and the current time sent to MATLAB.

R**esumé:** Stateflow indeed provided a solid base for modeling and performing the tasks of this benchmark. The problems of simultaneous access respectively the correct implementation of accurate access to the chopsticks was achieved by a simple, yet efficient workaround. Simulation results were easily acquired and exported into MATLAB.

F. Breitenecker, Vienna Univ. of Technology, Austria

Received: June 28, 2007
Revised: November 17, 2007; January 18, 2008
Accepted: February 10, 2008

An Object-oriented DEV Approach to ARGESIM Benchmark C16 'Restaurant Business Dynamics' using Enterprise Dynamics

Shabnam Michèle Tauböck, Philipp Jahn, Thomas Polzer, Alexandra Schuster, TU Vienna, Austria;
shaby@osiris.tuwien.ac.at, *{e0325871,e0325077,e0100211}@student.tuwien.ac.at*

Simulator: Enterprise Dynamcis (ED) is a discrete-event simulation tool for modelling, simulating, visualizing, and monitoring dynamic-flow process activities and systems. It is based on an object-oriented concept represented by so called atoms. Atoms are objects with four dimensions (x, y, z, and time) that can be dragged and dropped from a library into the modelling window e.g. source, sink, queue, server. It is also possible to create new atoms by using Taylor ED's Atom Editor. Defining the dependencies between atoms is done by connecting atoms through channels. Each atom may have multiple input and output channels. For a certain behaviour und functionality of an atom its parameters have to be set accordingly.

Model: An event driven approach was used and the following atoms were defined:

- `InitializeRestaurantBusinessDynamics`
- `Town`
- `tableRestaurant`
- `tablePeople`
- `Restaurant`
- `People`

The atom `InitializeRestaurantBusinessDynamics` places all the cities (atom `town`) and restaurants in the model window and randomly calculates the places where the people will be positioned. The probability of each position will depend on the catchment area of the cities.

The simulation is initialized by loading all atoms into the library of Taylor ED, dragging the initialization-atom into the model window and resetting the simulation using the *Reset* button. An example screenshot of the situation directly after the initialisation is shown in Figure 1. During initialization two tables (atoms) `tableRestaurant` and `tablePeople` are created to simplify the calculations. The first table provides a list of active restaurants with some actual values like profit and tax. The second table contains the number of people living in each 20x20 cell.

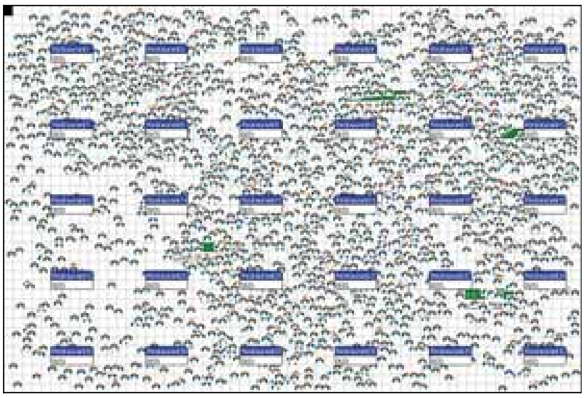


Figure 1. Screenshot of an initial state

The people atom is placed 3000 times into the 600x400 area; it triggers an event for visiting a certain restaurant nearby. After visiting a restaurant the next dining event is randomly set within an interval of 8 days. The visited restaurant is also selected randomly from those which are in proximity (distance < 100) of the person. For faster simulation runs only people having at least one restaurant nearby are taken into account during the calculations.

The atom `restaurant` calculates its revenue every week. The revenue is incremented by one each time a restaurant gets a visit event from a people atom. After each week a fixed amount of running costs (= 150) is subtracted and if still positive the tax of the remaining revenue is paid to the government. If the remaining profit is higher than the profit margin (=350) then with a certain opening probability a new restaurant-atom is created. For calculating the new position with the best ratio of People Density / Restaurant Density ratio (see Task c) the `tablePeople` atom is needed. If the profit is below the threshold the active restaurant-atom is destroyed with some closing probability.

A-Task: Several simulation runs were executed with times of 1, 5, and 10 years. The results are shown in Table 1.

The number of restaurants levels off at 4.6 after a short period of time. Therefore there is no advantage in simulating over a greater period of time. Interest-

period	mean	minimum	all
1 year	4.61	2	45
5 years	4.61	2	111
10 years	4.6	2	195

Table 1. Results of Task a

ingly to see is that only about 3.3 of the 4.6 restaurants exist over a longer period of time and make appreciable profit while the others only exist for a shortl time. The minimum number of existing restaurants was 2. The column 'all' shows the number of all restaurants having existed up to now (even if they existed only for a short period of time). So every year about 15 new restaurants are opened.

B-Task: Table 2 and Figure 2 show the results of task b. During several simulation runs the tax income for the government and the profit of the restaurants was recorded while varying the tax rate from 15% to 45%. As expected the higher the tax rate the less restaurants survived during the first year. Taking 45% as tax rate only 0.6 restaurants survived, while using 40% increased the number of active restaurants after one year to 1.8. As seen in Figure 2 there are some local maxima but the optimum tax rate for the government is 40%.

	10%	15%	20%	25%
Tax income	12831	19085	21784	21146
Profit	112198	105320	84337	61299

	30%	35%	40%	45%
Tax income	33366	29776	34394	21705
Profit	76032	53794	49795	25254

Table 2. Results of Task b

C-Task: Again several simulation runs were executed over the time period of one year. This time we were varying the weight coefficient k from 0 to 6 to get the highest profit for new restaurants. We

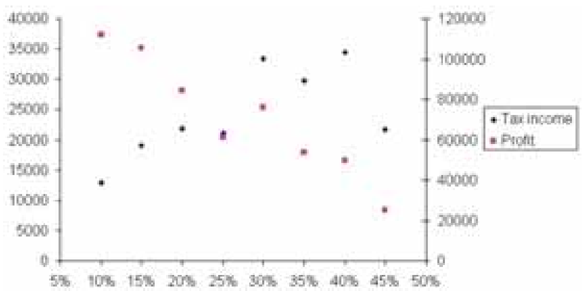


Figure 2. Results of Task b – Tax Income and Profit in Dependency on Tax Rate

K	0	0.5	1	2	3	4	5	6
Profit	20	45	37	25	18	17	8	3

Table 3. Profit of new Restaurants in Thousands

expected that a k of zero will not lead to a maximum since the calculation of the new location of a restaurant only depends on the people density within a certain cell ignoring the number of restaurants nearby completely. On the other hand the higher k is set the worse it is for cells with a restaurant in close proximity, but a high people density, to open a new restaurant.

Our simulation results have shown that 0.5 is the optimum k for the maximum profits of new restaurants (see Table 3 and Figure 3 for details).

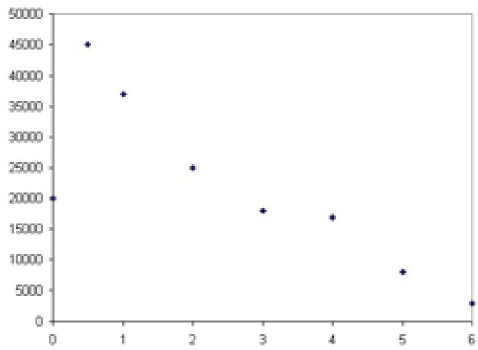


Figure 3. Results of Task c – Profit of new Restaurants Depending on the Weight Coefficient k

Resumé: Enterprise Dynamics is able to handle the high amounts of events created by the restaurants as well as the people. For executing the Tasks the Experiment Atom could be used, that allows to execute several simulation runs quick and efficiently.

Corresponding author: Shabnam Michèle Tauböck,
Department of Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
shaby@osiris.tuwien.ac.at

Received: June 28, 2007
Revised: October 18, 2007
Revised: January 20, 2008
Accepted: January 25, 2008

SNE News Section

Data & Quick Info

Contents

Simulation News Europe is the official journal of EUROSIM and sent to most members of the EUROSIM Societies as part of the membership benefits. Furthermore **SNE** is distributed to other societies and to individuals active in the area of modelling and simulation. SNE is registered with ISSN 1015-8685. Circulation of printed version is 3000 copies.

SNE at Web recent issues of SNE are also available via internet at `www.argesim.org`. Members of EUROSIM Societies may have access to the SNE Archive. This special *News Section* compiles date from EUROSIM and EUROSIM societies: addresses, weblinks, officers of societies with function and email. This *EUROSIM Data & Quick Info* is published regularly in SNE issues (except for special issues).

SNE Reports Editorial Board

INFO: WWW.ARGESIM.ORG; sne@argesim.org

If you have any information, announcement, etc. you want to see published, please contact a member of the editorial board in your country or sne@argesim.org.

Editorial Information/Impressum - see front cover

Information EUROSIM

EUROSIM Federation of European Simulation Societies

General Information. *EUROSIM*, the Federation of European Simulation Societies, was set up in 1989. The purpose of EUROSIM is to provide a European forum for regional and national simulation societies to promote the advancement of modelling and simulation in industry, research, and development.

→ www.eurosim.info

Member Societies. EUROSIM members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation. At present EUROSIM has eleven full members and three observer members:

ASIM	Arbeitsgemeinschaft Simulation *Austria, Germany, Switzerland*
CROSSIM	Croatian Society for Simulation Modeling *Croatia*
CSSS	Czech and Slovak Simulation Society *Czech Republic, Slovak Republic*
DBSS	Dutch Benelux Simulation Society *Belgium, Netherlands*
FRANCOSIM	Société Francophone de Simulation *Belgium, France*
HSS	Hungarian Simulation Society *Hungary*
ISCS	Italian Society for Computer Simulation *Italy*
PSCS	Polish Society for Computer Simulation *Poland*
SIMS	Simulation Society of Scandinavia *Denmark, Finland, Norway, Sweden*
SLOSIM	Slovenian Simulation Society *Slovenia*
UKSIM	United Kingdom Simulation Society *UK, Ireland*
CEA-SMSG	Spanish Modelling and Simulation Group *Spain, Observer Member*
LSS	Latvian Simulation Society *Latvia, Observer Member*
ROMSIM	Romanian Society for Modelling and Simulation, *Romania, Observer Member*

Contact addresses, weblinks and officers of the societies may be found in the information part of the societies.

EUROSIM board/EUROSIM officers. EUROSIM is governed by a board consisting of one representative of each member society, president and past president, and representatives for SNE and SIMPRA. The President is nominated by the society organising the next EUROSIM Congress. Secretary and Treasurer are elected out of members of the Board.

President	Mikuláš Alexík (CSSS), *alexik@frtk.fri.utc.sk*
Past president	Borut Zupančič (SLOSIM) *borut.zupancic@fe.uni-lj.si*
Secretary	Peter Fritzson (SIMS) *petfr@ida.liu.se*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
SIMPRA Repres.	Jürgen Halin *halin@iet.mavt.ethz.ch*
SNE Repres.	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

SNE – Simulation News Europe. EUROSIM societies are offered to distribute to their members the journal *Simulation News Europe* (SNE) as official membership journal. SNE is a scientific journal with reviewed contributions in the *Notes Section* as well as a membership newsletter for EUROSIM with information from the societies in the *News Section*. Publisher are EUROSIM, ARGESIM and ASIM.

Editor-in-chief	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

→ www.argesim.org, menu SNE

→ www.asim-gi.org, menu International

EuroSim Congress. EUROSIM is running the triennial conference series EUROSIM Congress. The congress is organised by one of the EUROSIM societies. EUROSIM 2010 will be organised by CSSS in Prague, September 5-10, 2010.

Information	Mikulas Alexik (CSSS) *alexik@frtk.utc.sk*
Chair OC 2010	Miroslav Šnorek *snorek@fel.cvut.cz*

→ www.eurosim.org

ASIM
German Simulation Society
Arbeitsgemeinschaft Simulation

ASIM (Arbeitsgemeinschaft Simulation) is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria. ASIM was founded in 1981 and has now about 700 individual members, and 30 institutional or industrial members. Furthermore, ASIM counts about 300 affiliated members.

→ `www.asim-gi.org` with members' area
info@asim-gi.org, admin@asim-gi.org
ASIM – Inst. f. Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

ASIM Working Groups. ASIM, part of GI - Gesellschaft für Informatik, is organised in Working Groups, dealing with applications and comprehensive subjects:

GMMS	Methods in Modelling and Simulation Peter Schwarz, *schwarz@eas.iis.fhg.de*
SUG	Simulation in Environmental Systems Wittmann, *wittmann@informatik.uni-hamburg.de*
STS	Simulation of Technical Systems H.T.Mammen, *Heinz-Theo.Mammen@hella.com*
SPL	Simulation in Production and Logistics Sigrid Wenzel, *s.wenzel@uni-kassel.de*
SVS	Simulation of Transport Systems U. Brannolte, *Brannolte@bauing.uni-weimar.de*
SBW	Simulation in OR C. Böhnlein, *boehnlein@wiinf.uni-wuerzburg.de*
EDU	Simulation in Education/Education in Simulation W. Wiechert, *wiechert@simtec.mb.uni-siegen.de*

ASim Publications

SNE – Simulation News Europe. ASIM is publishing (co-publishing) SNE, which is regularly published and sent to all ASIM members (as part of their membership; 900 issues) and for promotion purposes (300 issues). Since 2006, the ASIM Working Groups publish *SNE Special Issues* with state-on-the-art reports on modelling and simulation in their workscope.

ASIM News. In December 2005, the ASIM Nachrichten has been replaced by an electronic news-letter - ASIM Newsletter. Editors are Th. Pawletta and C. Deatcu, Univ. Wismar, `pawel@mb.hs-wismar.de`.

ASIM Notes/ASIM Mitteilungen. The trademark ASIM Mitteilungen (ASIM Note) stands for all publications of ASIM and of the the ASIM Working Groups. Each publication gets an identification as ASIM Notes, independent of the publisher, and independent of the publication medium (printed books, CD, Web). ASIM Notes range from printed books (with CDs) published by Springer, via workshop publication published in SNE or ARGESIM, to compiled abstracts publishes at the ASIM weberver.

ASIM Books. ASIM co-operates with the SCS Publishing House e.V., with ARGESIM (Vienna University of Technology), and with Shaker Verlag Aachen in publication of two book series (Fortschritte in der Simulationstechnik - Frontiers in Simulation and Fortschrittsberichte Simulation - Advances in Simulation) and in publication of Proceedings. Publications in these series range from monographs via proceedings to PhD theses.

ASIM Board and Officers: The ASIM board consists of officers (elected all three years), of the chairpersons of the ASIM Working Groups (independently elected all three years), and of co-opted specialists.

Prsident	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*
Vice presidents	Sigrid Wenzel *s.wenzel@uni-kassel.de*
	Thorsten Peawletta, *pawel@mb.hs-wismar.de*
Secretary	Claus-Burkhard Böhnlein, *boehnlein@wiinf.uni-wuerzburg.de*
Treasurer	Ingrid Bausch-Gall, *Ingrid@Bausch-Gall.de*
Membership affairs	S. Wenzel, *s.wenzel@uni-kassel.de*
	W. Maurer, *werner.maurer@zhwin.ch*
	I. Bausch-Gall, *Ingrid@Bausch-Gall.de*
	F. Breitenecker (*mail address above*)
Universities	W. Wiechert *wiechert@simtec.mb.uni-siegen.de*
Industry	S. Wenzel, *s.wenzel@uni-kassel.de*
	K. Panreck, *Klaus.Panreck@hella.com*
Conferences	Klaus Panreck *Klaus.Panreck@hella.com*
	Albrecht Gnauck *albrecht.gnauck@tu-cottbus.de*
Publications	Th. Pawletta, *pawel@mb.hs-wismar.de*
	F. Breitenecker (*mail address above*)
Repr. EUROSIM	F. Breitenecker (*mail address above*)
Deputy	W. Wiechert *wiechert@simtec.mb.uni-siegen.de*
Edit. Board SNE	Thorsten Pawletta, *pawel@mb.hs-wismar.de*
Web EUROSIM	Anna Mathe, *anna.mathe@tuwien.ac.at*

CROSSIM – Croatian Society for Simulation Modelling

CROSSIM-*Croatian Society for Simulation Modelling* was founded in 1992 as a non-profit society with the goal to promote knowledge and use of simulation methods and techniques and development of education. CROSSIM is a full member of EUROSIM since 1997.

→ www.eurosim.info

✉ CROSSIM / Jadranka Božikov
Andrija Stampar School of Public Health,
Medical School, University of Zagreb
Rockefeller St. 4, HR-10000 Zagreb, Croatia

President	Jadranka Božikov, *jbozikov@snz.hr*
Vice president	Vesna Dušak, *vdusak@foi.hr*
Secretary	Vesna Bosilj-Vukšić, *vbosilj@efzg.hr*
Executive board members	Vlatko Čerić, *vceric@efzg.hr*
	Tarzan Legović, *legovic@irb.hr*
Repr. EUROSIM	Jadranka Bozikov, *jbozikov@snz.hr*
Edit. Board SNE	Jadranka Bozikov, *jbozikov@snz.hr*
Web EUROSIM	Jadranka Bozikov, *jbozikov@snz.hr*

CSSS – Czech and Slovak Simulation Society

CSSS -The *Czech and Slovak Simulation Society* has about 150 members working in Czech and Slovak national scientific and technical societies (*Czech Society for Applied Cybernetics and Informatics*, *Slovak Society for Applied Cybernetics and Informatics*). The main objectives of the society are: development of education and training in the field of modelling and simulation, organising professional workshops and conferences, disseminating information about modelling and simulation activities in Europe. Since 1992, CSSS is full member of EUROSIM.

→ www.fit.vutbr.cz/CSSS

✉ CSSS / Miroslav Šnorek, CTU Prague
FEE, Dept. Computer Science and Engineering,
Karlovo nam. 13, 121 35 Praha 2, Czech Republic

President	Miroslav Šnorek, *snorek@fel.cvut.cz*
Vice president	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Treasurer	Evžen Kindler, *ekindler@centrum.cz*
Scientific Secr.	A. Kavička, *Antonin.Kavicka@upce.cz*
Repr. EUROSIM	Miroslav Šnorek, *snorek@fel.cvut.cz*
Deputy	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Edit. Board SNE	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Web EUROSIM	Petr Peringer, *peringer@fit.vutbr.cz*

DBSS – Dutch Benelux Simulation Society

The Dutch Benelux Simulation Society (DBSS) was founded in July 1986 in order to create an organisation of simulation professionals within the Dutch language area. DBSS has actively promoted creation of similar organisations in other language areas. DBSS is a member of EUROSIM and works in close cooperation with its members and is further affiliated with SCS International, IMACS, and the Chinese Association for System Simulation and the Japanese Society for Simulation Technology.

→ www.eurosim.info

✉ DBSS / A. W. Heemink
Delft University of Technology, ITS - twi,
Mekelweg 4, 2628 CD Delft, The Netherlands

President	A. Heemink, *a.w.heemink@its.tudelft.nl*
Vice president	W. Smit, *smitnet@wxs.nl*
Treasurer	W. Smit, *smitnet@wxs.nl*
Secretary	W. Smit, *smitnet@wxs.nl*
Repr. EUROSIM	A. Heemink, *a.w.heemink@its.tudelft.nl*
Deputy	W. Smit, *smitnet@wxs.nl*
Edit. Board SNE	A. Heemink, *a.w.heemink@its.tudelft.nl*

FRANCOSIM – Société Francophone de Simulation

FRANCOSIM was founded in 1991 and aims to the promotion of simulation and research, in industry and academic fields. Francosim operates two poles.

- Pole Modelling and simulation of discrete event systems. Pole Contact: *Henri Pierreval, pierreva@imfa.fr*
- Pole Modelling and simulation of continuous systems. Pole Contact: *Yskandar Hamam, y.hamam@esiee.fr*

→ www.eurosim.info

✉ FRANCOSIM / Yskandar Hamam
Groupe ESIEE, Cité Descartes,
BP 99, 2 Bd. Blaise Pascal,
93162 Noisy le Grand CEDEX, FRANCE

President	Yskandar Hamam, *y.hamam@esiee.fr*
Treasurer	François Rocaries, *f.rocaries@esiee.fr*
Repr. EUROSIM	Yskandar Hamam, *y.hamam@esiee.fr*
Edit. Board SNE	Yskandar Hamam, *y.hamam@esiee.fr*

HSS – Hungarian Simulation Society

The Hungarian Member Society of EUROSIM was established in 1981 as an association promoting the exchange of information within the community of people involved in research, development, application and education of simulation in Hungary and also contributing to the enhancement of exchanging information between the Hungarian simulation community and the simulation communities abroad. HSS deals with the organization of lectures, exhibitions, demonstrations, and conferences.

→ www.eurosim.info

✉ HSS / András Jávor,
Budapest Univ. of Technology and Economics,
Sztoczek u. 4, 1111 Budapest, HUNGARY

President	András Jávor, *javor@eik.bme.hu*
Vice president	Gábor Szűcs, *szucs@itm.bme.hu*
Secretary	Ágnes Vigh, *vigh@itm.bme.hu*
Repr. EUROSIM	András Jávor, *javor@eik.bme.hu*
Deputy	Gábor Szűcs, *szucs@itm.bme.hu*
Edit. Board SNE	András Jávor, *javor@eik.bme.hu*
Web EUROSIM	Gábor Szűcs, *szucs@itm.bme.hu*

ISCS – Italian Society for Computer Simulation

The Italian Society for Computer Simulation (ISCS) is a scientific non-profit association of members from industry, university, education and several public and research institutions with common interest in all fields of computer simulation.

→ www.eurosim.info

✉ ISCS / Mario Savastano,
c/o CNR - IRSIP,
Via Claudio 21, 80125 Napoli, ITALY

President	MarioSavastano, *mario.savastano@unina.it*
Vice president	F. Maceri, *Franco.Maceri@uniroma2.it*
Secretary	Paola Provenzano, *paola.provenzano@uniroma2.it*
Treasurer	Pasquale Arpaia
Repr. EUROSIM	F. Maceri, *Franco.Maceri@uniroma2.it*
Edit. Board SNE	Mario Savastano, *mario.savastano@unina.it*

PSCS – Polish Society for Computer Simulation

PSCS was founded in 1993 in Warsaw. PSCS is a scientific, non-profit association of members from universities, research institutes and industry in Poland with common interests in variety of methods of computer simulations and its applications. At present PSCS counts 264 members.

→ www.ptsk.man.bialystok.pl

✉ PSCS / Leon Bobrowski, c/o IBIB PAN,
ul. Trojdena 4 (p.416), 02-109 Warszawa, Poland

President	Leon Bobrowski, *leon@ibib.waw.pl*
Vice president	A. Chudzikiewicz, *ach@it.pw.edu.pl*
Treasurer	Z. Sosnowski, *zenon@ii.pb.bialystok.pl*
Secretary	Zdzislaw Galkowski, *Zdzislaw.Galkowski@simr.pw.edu.pl*
Repr. EUROSIM	Leon Bobrowski, *leon@ibib.waw.pl*
Deputy	A.Chudzikiewicz, *ach@it.pw.edu.pl*
Edit. Board SNE	Z.Sosnowski, *zenon@ii.pb.bialystok.pl*

SIMS – Scandinavian Simulation Society

SIMS is the *Scandinavian Simulation Society* with members from the four Nordic countries Denmark, Finland, Norway and Sweden. The SIMS history goes back to 1959. SIMS practical matters are taken care of by the SIMS board consisting of two representatives from each Nordic country. Iceland will be represented by one board member.

SIMS Structure. SIMS is organised as federation of regional societies. There are FinSim (Finnish Simulation Forum), DKSIM (Dansk Simuleringsforening) and NFA (Norsk Forening for Automatisering).

→ www.scansims.org

✉ SIMS/Peter Fritzson, IDA, Linköping University,
58183, Linköping, Sweden

President	Peter Fritzson, *petfr@ida.liu.se*
Treasurer	Vadim Engelson, *vaden@ida.liu.se*
Repr. EUROSIM	Peter Fritzson, *petfr@ida.liu.se*
Edit. Board SNE	Esko Juuso, *esko.juuso@oulu.fi*
Web EUROSIM	Vadim Engelson, *vaden@ida.liu.se*

SLOSIM – Slovenian Society for Simulation and Modelling

SLOSIM - Slovenian Society for Simulation and Modelling was established in 1994 and became the full member of EUROSIM in 1996. Currently it has 69 members from both slovenian universities, institutes, and industry. It promotes modelling and simulation approaches to problem solving in industrial as well as in academic environments by establishing communication and cooperation among corresponding teams.

→ msc.fe.uni-lj.si/SLOSIM

✉ SLOSIM / Rihard Karba, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

President	Rihard Karba, *rihard.karba@fe.uni-lj.si*
Vice president	Leon Žlajpah, *leon.zlajpah@ijs.si*
Secretary	Aleš Belič, *ales.belic@fe.uni-lj.si*
Treasurer	Milan Simčič, *milan.simcic@fe.uni-lj.si*
Repr. EUROSIM	Rihard Karba, *rihard.karba@fe.uni-lj.si*
Deputy	Borut Zupančič, *borut.zupancic@fe.uni-lj.si*
Edit. Board SNE	Rihard Karba, *rihard.karba@fe.uni-lj.si*
Web EUROSIM	Aleš Belič, *ales.belic@fe.uni-lj.si*

UKSIM – United Kingdom Simulation Society

UKSIM has more than 100 members throughout the UK from universities and industry. It is active in all areas of simulation and it holds a biennial conference as well as regular meetings and workshops.

→ www.uksim.org.uk

✉ UKSIM / Alessandra Orsoni,
Kingston Business School, Kingston Hill,
Kingston-Upon-Thames, Surrey, KT2 7LB, UK

President	David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Secretary	Alessandra Orsoni, *A.Orsoni@kingston.ac.uk*
Treasurer	B. Thompson, *barry@bjtcon.ndo.co.uk*
Membership chair	K. Al-Begain, *kbegain@glam.ac.uk*
Univ. liaison chair	R. Cheng, *rchc@maths.soton.ac.uk*
Ind. liaison chair	Richard Zobel, *r.zobel@ntworld.com*
Conf. venue chair	John Pollard, *j.pollard@ee.ucl.ac.uk*
Repr. EUROSIM	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Edit. Board SNE	A. Orsoni, *A.Orsoni@kingston.ac.uk*

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control In order to improve the efficiency and to deep into the different fields of automation, the association is divided into thematic groups, one of them is named 'Modelling and Simulation', constituting the group.

→ www.cea-ifac.es/wwwgrupos/simulacion

✉ CEA-SMSG / María Jesús de la Fuente,
System Engineering and AutomaticControl department,
University of Valladolid,
Real de Burgos s/n., 47011 Valladolid, SPAIN

President	María J. la Fuente, *maria@autom.uva.es*
Repr. EUROSIM	María J. la Fuente, *maria@autom.uva.es*

LSS – Latvian Simulation Society

The Latvian Simulation Society (LSS) has been founded in 1990 as the first professional simulation organisation in the field of Modelling and simulation in the post-Soviet area. Its members represent the main simulation centres in Latvia, including both academic and industrial sectors.

→ briedis.itl.rtu.lv/imb/

✉ LSS / Yuri Merkuryev, Dept. of Modelling
and Simulation Riga Technical University
Kalku street 1, Riga, LV-1658, LATVIA

President	Yuri Merkuryev, *merkur@itl.rtu.lv*
Repr. EUROSIM	Yuri Merkuryev, *merkur@itl.rtu.lv*

ROMSIM – Romanian Modelling and Simulation Society

ROMSIM has been founded in 1990 as a non-profit society, devoted to both theoretical and applied aspects of modelling and simulation of systems. ROMSIM currently has about 100 members from both Romania and Republic of Moldavia.

→ briedis.itl.rtu.lv/imb/

✉ LSS / Yuri Merkuryev, Dept. of Modelling
and Simulation Riga Technical University
Kalku street 1, Riga, LV-1658, LATVIA

President	Florin Stanciulescu, *sflorin@ici.ro*
Vice president	Florin Hartescu, *flory@ici.ro*
Secretary	Zoe Radulescu, *radulescu@ici.ro*
Repr. EUROSIM	Florin Stanciulescu, *sflorin@ici.ro*
Deputy	Florin Hartescu, *flory@ici.ro*
Edit. Board SNE	Florin Stanciulescu, *sflorin@ici.ro*

Report Congress EUROSIM 2007

EUROSIM 2007

6th EuroSim Congress on Modelling and Simulation
Sept. 9–13, 2007

Ljubljana, Slovenia

More information: www.eurosim2007.org

The EUROSIM Congress is organised every three years by a member society of EUROSIM. After Capri, Vienna, Helsinki, Delft and Paris the 6th EUROSIM Congress was organised by the Slovenian Society for Simulation and Modelling, SLOSIM, and took place in Sept. 9–13, 2007 in Ljubljana, the capital of Slovenia.

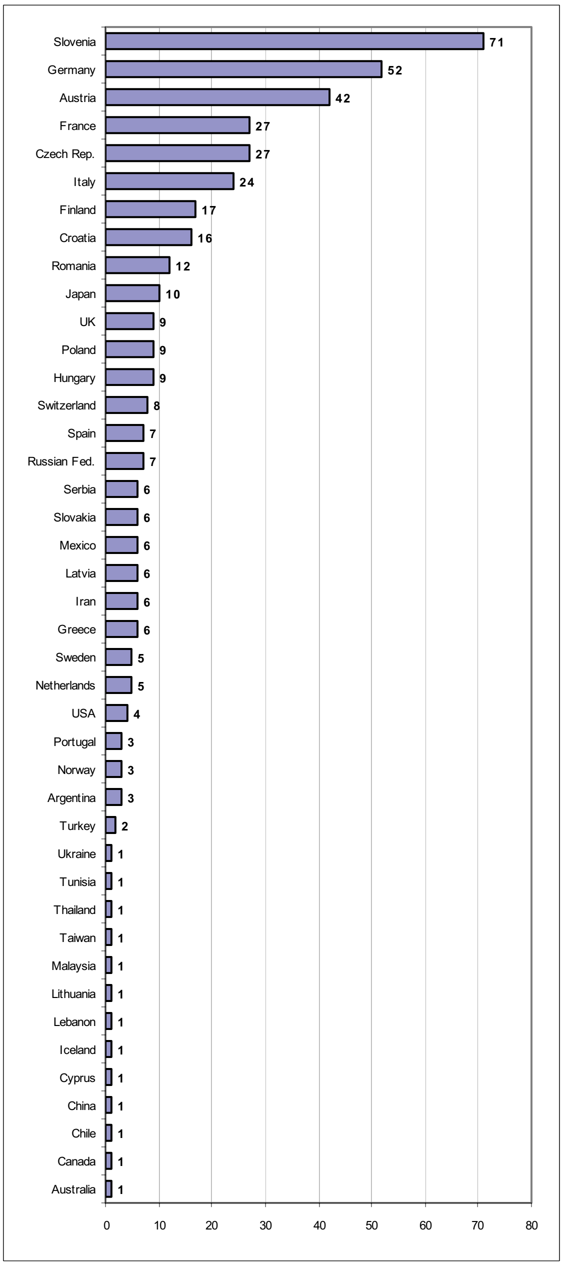
The efforts of the organisers have resulted in an unexpectedly large interest in the event, as much more than five hundred papers from almost 50 countries were submitted. Altogether **420 contributions from 42 countries** were included in the final programme—see graph below. This proves that modelling and simulation is, in spite of being a very traditional discipline, still attractive, modern, and with a wide range of applications and research possibilities, and is closely connected with many of the most sophisticated and modern disciplines.

The scope of the congress included all aspects of continuous, discrete (event) and hybrid modelling, simulation, identification and optimisation approaches. The programme scheme consisted of tutorials, plenary lectures, regular sessions, special sessions, poster sessions and a student competition.

Tutorials covered the more general areas of broader interest. Below you can find the titles and the authors of the tutorials. Each tutorial lasted 3 hours. The first and second tutorials were organised with hands-on working in the computer room.

Title of Tutorial	Presenter(s)
Introduction to Object-Oriented Modeling and Simulation with Modelica Using the OpenModelica Environment	Fritzson P., Lundvall H., Brugård J.
Super-Object-Oriented Programming and Model Nesting	Kindler E.
Inverse Simulation Methods and Applications	Murray-Smith D.

Plenary lectures were presented by five outstanding speakers, all of them well known in the field of mod-



Number of contributions from different countries

Prof. David Murray-Smith during the tutorial presentation

SNE 18/1, April 2008

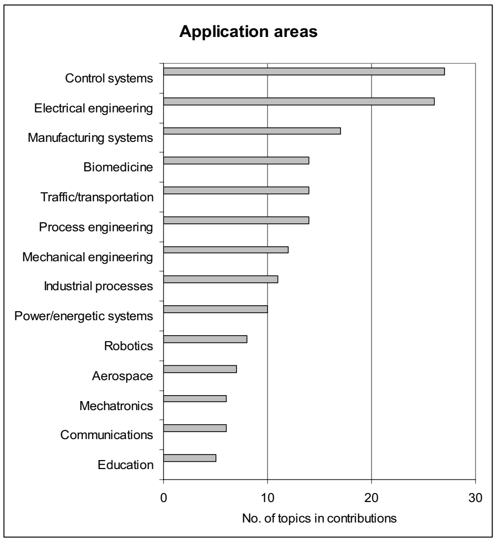
Unusual Prof. Felix Breitenecker's plenary lecture – a mixture of science and art

elling and simulation. The lectures covered a wide range of modelling and simulation topics.

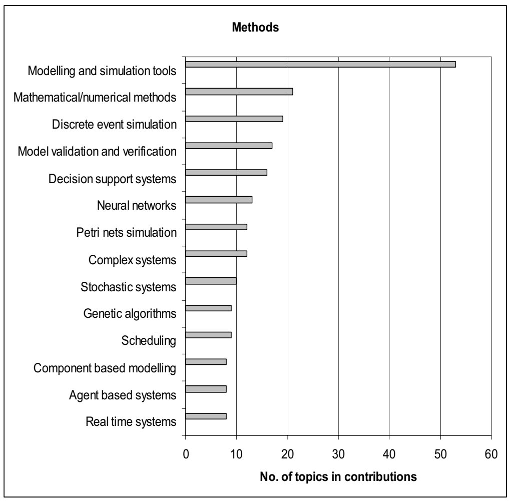
The Titles of the Plenary Papers	Presenters
Electronic Circuit Modeling and Simulation in Modelica	Cellier F.
Integrated Multiscale Simulation of Continuous Casting of Steel	Šarler B.
Experiences and Trends in Modelling and Simulation of Integrated Industrial Processes	Juslin K.
An ODE for the Renaissance	Breitenecker F.
The Challenge of Modeling High Speed Flows	Longo J.M.A.

The regular programme consisted of regular papers and posters. The only distinction was in the form of presentation: oral or poster. There were 34 regular sessions with oral presentation with 203 contributions and 1 poster session with 30 contributions, altogether 233 papers, which covered the congress' scope and topics. All the papers were revised by at least three members from the International Programme Committee. The figure below shows the number of congress topics declared by authors for theirs contributions with regard to methods and application areas. Only more frequently chosen topics are shown.

Prof. François Cellier during the plenary lectue

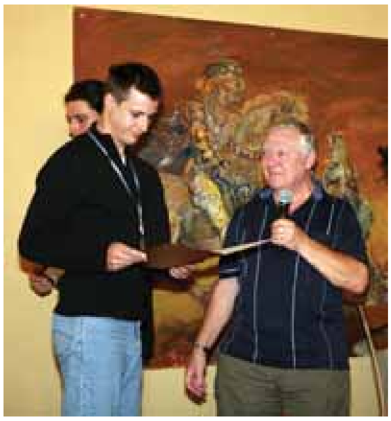


# of contributions describing different application areas



# of contributions describing different methodologies

Vladimir Košel from Slovakia, one of the poster session winners



Luka Teslič from Slovenia, one of the student competition winners

The best three posters were selected in the poster session. These were:

1. *Plastic Deformation of Aluminum Bonding Wire Impressed by Wedge Bonding Tool*
   Košel V., Zarbakhsh J., Glavanovics M.
2. *Model-Based Production Control*
   Gradišar D., Zorzut S., Jovan V.
3. *Augmented Reality Based Technologies for Supporting Assembly Work*
   Sääski J., Salonen T., Siltanen S., Hakkarainen M., Woodward C.

Twenty-three special sessions (some were organised as tracks) with 168 papers were organised by experts from the area of modelling and simulation and covered rather specific areas relating to modelling and simulation. Partly these were invited papers, and partly they were submitted after the session organisers' call. The review procedure was organised by the session organisers. You can see the names of the sessions and the organisers in the table below. At this point we would like to express our gratitude to the session organisers.

Title of Special Session	Organisers
Education in Simulation / Simulation in Education	Wiechert W.
Simulation in Economics and Business	Štemberger M., Orsoni A.
Modelling of Cryogenic Systems and their Applications	Rachid A., Chadli M., Coppier H.
Modelling and Simulation in Mechatronics	Schmucker U.
Computational Intelligence and Discrete Simulation	Huyet A.L, Pierreval H.

Title of Special Session	Organisers
Increased Predictability of Crash Models	Eichberger A.
Digital Factory/Simulation and Optimization of Industrial Processes	Jósvai J., März L.
Successful Application of Simulation in Industry	Juslin K.
Fuzzy Systems	Škrjanc I.
Simulations, Modelling and Optimization of VLSI Circuits	Strle D.
Multidisciplinary Design Optimization	Dellino G., Lino P., Meloni C., Rizzo A.
Algebraic Methods and Algorithms in Modelling Discrete Dynamical Systems	Li A.
Simulation in Electric Power Systems	Mihalič R.
Modelling and Simulation in Medicine and Pharmacy	Drinovec J., Mrhar A., Atanasijević-Kunc
Modelling and Simulation in Structural Mechanics	Lebon F., Maceri F.
Models Networks for Process Systems Simulation	Savković-Stevanović J.B.
Control and Decision for Complex Systems	Popescu D., Tanguy G.D.
Domain Modelling through Autonomous Discovery	Bratko I.
Simulation of Multifield and Multiscale Problems in Structural and Material Engineering	Callari C., Maceri F.
Agent-Based and Dynamic Approaches to Modelling in Economics	Wöckl J., Almeder C.

Title of Special Session	Organisers
Modelling and Simulation in the Vehicle Thermal Management System	Puntigam W.
Alternative Modelling and Comparisons and Benchmarking in Modelling and Simulation	Breitenecker F., Wiechert W.
Modelling of Structural Dynamic Systems – Model Reduction Methods	Schwarz P., Breitenecker F.

The student competition contained papers describing the work of undergraduate or bachelor/master students. This means that it was not intended for PhD students unless they presented their earlier work. Ten students competed in this session. The winners were:

1. *Simulation of a Mobile Robot With a LRF and Map Building*
   Teslić L., Klančar G., Škrjanc I.
2. *Bifurcations and Chaos in Automatic Control System*
   Kocewiak Ł. H.
3. *Configuration of UAV Autopilots' Dynamics Using a 3 Dof Aircraft Motion Simulator*
   Tomažič T.*, Matko D.
   * University of Ljubljana, Slovenia

Social events. Beside welcome evening, opening and closing ceremony there were two unforgettable social events – Slovenia evening at Ljubljana Castle and Congress dinner combined with the visit of world known Postojna cave.

We are pleased to announce that a **special issue of Simulation Modelling Practice and Theory** including 20 revised and extended papers selected from the programme of EUROSIM 2007 is already prepared. The selection from 420 EUROSIM 2007 Congress

Unforgettable artistic programme: Miha Debevec and Tomaž Rožanec with diatonic and classical accordion

Five EUROSIM presidents: Franco Maceri (1992-95), Mikulaš Alexik (new president), Borut Zupančič (2004-07), Felix Breitenecker (1995-98) and Kaj Juslin (1998-2001) – from the left

The Mayor of Ljubljana Zoran Janković addressing the EUROSIM 2007 participants on Ljubljana Castle

Mikuláš Alexík and Borut Zupančič, new and old EUROSIM president below the traditional umbrella during the Congress diner ceremony contributions hopefully resulted in a high quality special issue.

At the end we would like to express our gratitude for the large contribution of the reviewers of the papers, first for the programme of the congress, and later for the mentioned special issue. The reviewers have supplied the authors with valuable comments and suggestions for improvements of their papers. In addition we would like to thank Elsevier (especially Mrs. Ineke Kolen and to the Editor-in-Chief Dr. Helen Karatza) for the very fruitful cooperation.

We hope that the participation at the 6th EUROSIM Congress was pleasant and productive, scientifically and socially. **We are looking forward to seeing you in Prague in Sept. 2010, where the next 7th EUROSIM Congress will take place**.

Borut Zupančič
Chair of the Congress
borut.zupancic@fe.uni-lj.si

Rihard Karba
Chair of the IPC
rihard.karba@fe.uni-lj.si

Miroslav Šnorek, the general chair of the 7th EUROSIM Congress invited participants to the next congress in Prague in Sept. 2010

29th EuroSim Board Meeting,

Sept. 11, 2007, Ljubljana, Slovenia

Report of the President

As this was the last Board Meeting under my presidency, I briefly described the main activities in the period 2004-07 which were:

- arrangement of documentation, Internal rules,
- promotion and cooperation: visits - Modelica 2005, SCS 2005, MOSIS 2005, ECMS 2006, PSCS 2006, ASC 2006,
- motivation for new membership: PSCS (Poland), LSS (Latvia), CEA-SMSG (Spain),
- working on EUROSIM website: EUROSIM activities, calendar of events, links, …and stimulation of EUROSIM member societies,
- regular and detailed reports in SNE,
- organization of EUROSIM 2007 Congress.

Report of the Editor in Chief of SIMPRA

Throughout the last years the development of SIMPRA was quite positive. In 2006 the number of issues of SIMPRA per year was increased from 8 to 10. The ISI Impact Factor has increased from 0.361 (2005) to 0.474. SIMPRA has become one of the leading journals in the area of modelling and simulation. The number of papers submitted to SIMPRA/year is in the order of 300. The acceptance ratio is about 30%. Prof. Helen Karatza took became a second Editor-in-Chief. It is intended to approach the member societies of EUROSIM for getting their support in finding qualified reviewers so as to reduce the handling time of articles. It was agreed that the special SIMPRA issue of EUROSIM 2007 (with general topics) will be published with 15-20 papers.

Elections of EUROSIM officers for 2007-2010 – installation of the new president, election of the secretary and treasurer

As the next EUROSIM congress will be organised by CSSS, the president for 2007-2010 is also from CSSS- CSSS proposed Mikulaš Alexik. New officers were elected by the board members:

President:	Mikulaš Alexik from CSSS
Secretary:	Esko Juuso from SIMS
Treasurer:	Felix Breitenecker from ASIM

Some new board members were nominated:

Miroslav Šnorek, CSSS (for Mikulaš Alexik),
Richard Zobel, UKSIM (for Alessandra Orsoni)
Esko Juuso, SIMS (for Peter Fritzson)
Juri Merkuryev, LSS representative
Emilio Jiménez, CEA SMSG group representative

EUROSIM membership: discussion about full membership Spain, Latvia

Decision: Latvian Simulation Society (LSS) is accepted as the full member of EUROSIM.

Decision: CEA-SMSG is accepted as the full member of EUROSIM.

EUROSIM events between two congresses

EUROSIM urgently needs more events between congresses – discussion in Vienna and Bratislava. Expressed and approved candidates for the near future: UKSIM 2008 and MATHMOD 2009 will be organized as EUROSIM events.

EUROSIM Congress 2007 in Ljubljana

(see also more detailed report before)

Extremely successful event. 421 contributions from 42 countries, 3 tutorials, 5 plenary lectures, 23 special sessions, altogether 63 sessions, poster session, student competition, 456 registered participants.

EUROSIM Congress 2010

Decision in Bratislava: The 7th EUROSIM Congress will be organised by CSSS in Sept. 2010 in Prague. Chair will be Miroslav Šnorek. Preliminary Call for papers were disseminated. Šnorek reported about some general aspects of the organisation.

EUROSIM award

EUROSIM gives EUROSIM award for individuals, groups or societies for long standing exceptional service to the modelling and simulation community. Candidates are normally proposed by EUROSIM societies, EUROSIM board members or the EUROSIM president. The final decision is made by EUROSIM board.

At the end of the report I hope that the period of my presidency was successful and productive and that the image of EUROSIM organisation has increased. I would like to thank all societies representatives for the fruitful cooperation. I wish all the best to my successor Mikulaš Alexik and also to the chair of the next EUROSIM Congress Miroslav Šnorek.

Borut Zupančič
EUROSIM past president
borut.zupancic@fe.uni-lj.si

515.000.000 KM, 380.000 SIMULATIONEN
UND KEIN EINZIGER TESTFLUG.
DAS IST MODEL-BASED DESIGN.
Nachdem der Endabstieg der beiden Mars Rover unter Tausenden von atmosphärischen Bedingungen simuliert wurde, entwickelte und testete das Ingenieur-Team ein ausfallsicheres Bremsraketen-System, um eine zuverlässige Landung zu garantieren. Das Resultat – zwei erfolgreiche autonome Landungen, die exakt gemäß der Simulation erfolgten.
Mehr hierzu erfahren Sie unter:
www. mathworks.de/mbd
MATLAB®
&SIMULINK®
The MathWorks
Accelerating the pace of engineering and science

EUROSIM 2010

organised by CSSS

September 2010, Prague, Czech Republic

EUROSIM 2010

7th EUROSIM Congress on Modelling and Simulation

Eurosim Congress the most important modelling and simulation event in Europe

September 5-10, 2010, Prague, Czech Republic

Congress Venue

The Congress will take place in Prague, the capital city of Czech Republic, at the Congress Center of Masaryk College, part of Czech Technical University, in cooperation with the Faculty of Electrical Engineering of CTU.

About Czech Technical University in Prague

Czech Technical University celebrates 300 years of its history in 2007. Under the name Estate Engineering Teaching Institute in Prague was founded by the rescript of the Emperor Josef I of 18 January 1707 on the basis of a petition of Christian Josef Willenberg (1676-1731). This school was reorganized in 1806 as the Prague Polytechnic, and, after the disintegration of the former AustroHungarian Empire in 1918, transformed in to the Czech Technical University in Prague.

About EUROSIM

EUROSIM, the federation of European simulation societies, was set up in 1989. Its purpose is to promote, especially through local simulation societies, the idea of modelling and simulation in different fields, industry, research and development. At present, EUROSIM has 14 full members and 4 observer members.

Congress Scope and Topics

The Congress scope includes all aspects of continuous, discrete (event) and hybrid modelling, simulation, identification and optimisation approaches. Contributions from both technical and non-technical areas are welcome. Two basic tracks will be organized: M&S Methods and Technologies and M&S Applications.

Czech Republic - EUROSIM 2010 Host Country

The Czech Republic is a country in the centre of Europe. It is interesting for its 1,000-year-long history, rich culture and diverse nature. The country is open to new influences and opportunities thanks to a high level of industrial infrastructure, safety measures and plural media. The location of the Czech Republic in the very heart of Europe contributes to the fact that one can get there easily and fast. Usually all it takes to enter the country is a valid passport. The Czech Republic belongs to the Schengen zone. The need for a visas to enter the Czech Republic is very exceptional.

Prague - EUROSIM 2010 Host City

Prague is a magical city of bridges, cathedrals, gold-tipped towers and church spires, whose image has been mirrored in the surface of the Vltava River for more than a millennium.Walking through the city, you will quickly discover that the entire history of European architecture has left splendid representatives of various periods and styles. There are Romanesque, Gothic, Renaissance, Baroque and Classicist buildings, as well as more modern styles, such as Art Nouveau and Cubist. A poet once characterized Prague as a symphony of stones.

About CSSS

CSSS (The Czech and Slovak Simulation Society) has about 150 members in 2 groups connected to the Czech and Slovak national scientific and technical societies (Czech Society for Applied Cybernetics and Informatics, Slovak Society for Applied Cybernetics and Informatics). Since 1992 CSSS is a full member of EUROSIM.

Invitation

Czech and Slovak Simulation Society is greatly honored with the congress organisation and will do the best to organise an event with a high quality scientific programme with some other acompanied actions but also with some unforgettable social events.

Mikuláš Alexík, EUROSIM president,
Miroslav Šnorek, president of CSSS, EUROSIM 2010 Chair