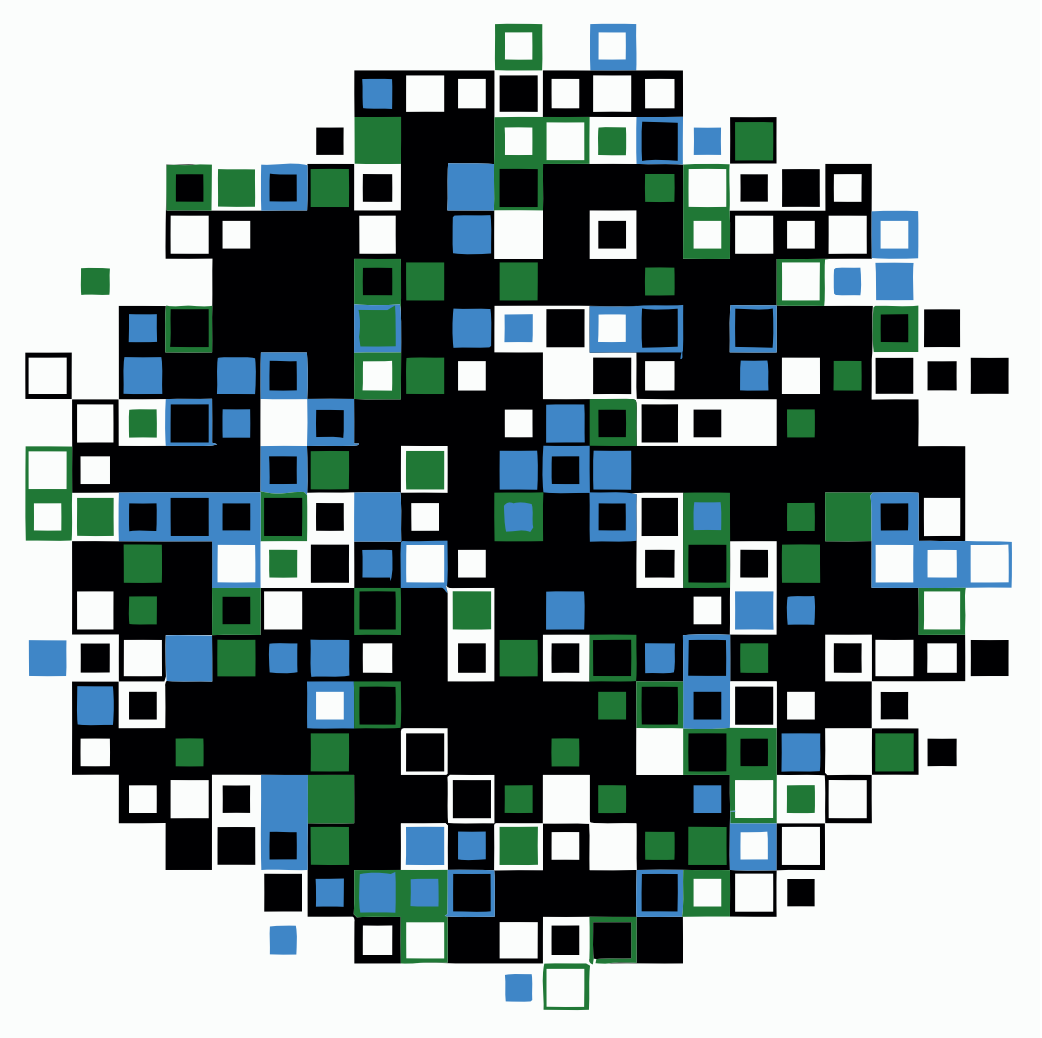
SNE SIMULATION NOTES EUROPE



Volume 21 Number 1
April 2011

doi: 10.11128/sne.21.1.1004

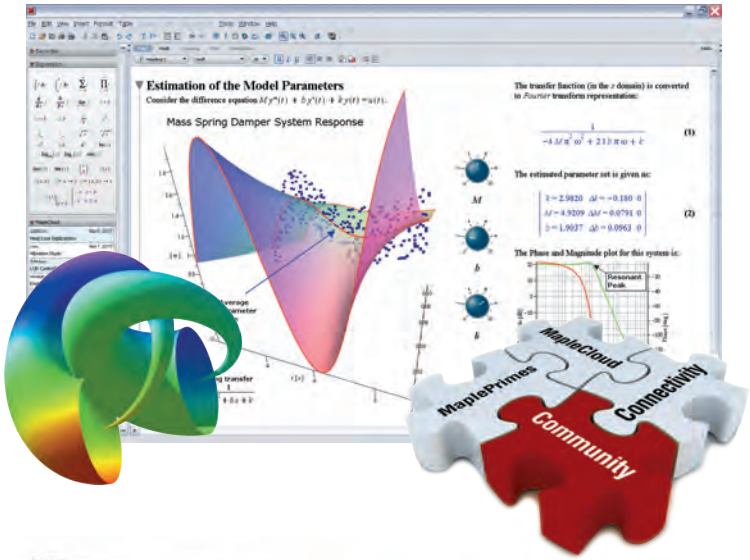
Print ISSN 2305-9974
Online ISSN 2306-0271

Journal on Developments and Trends in Modelling and Simulation

Membership Journal for Simulation Societies and Groups in EUROSIM

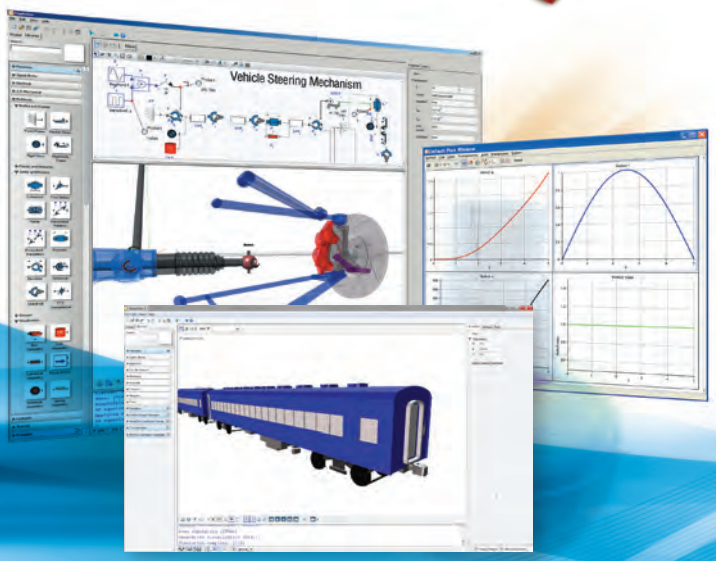
ARGESIM

New challenges in modeling and simulation require powerful tools



Maple™ 15

The result of over 30 years of cutting-edge research and development, Maple combines the world's most powerful mathematical computation engine with an intuitive, "clickable" user interface.



MapleSim™ 5

MapleSim is a physical modeling and simulation tool built on a foundation of symbolic computation technology.

Get a Price Quote:
www.maplesoft.com/sne

Contact us:

+49 (0)241/980919-30

Maplesoft
Mathematics • Modeling • Simulation
A Cybernet Group Company

www.maplesoft.com | germany@maplesoft.com

Editorial

Dear Readers – *This SNE issue starts with novelties and changes in SNE content, structure, layout and distribution. First, the change of SNE's name from Simulation News Europe to Simulation Notes Europe underlines the focus of SNE; second, SNE is now clearly structured as Print SNE (Print ISSN 2305-9974) and Online SNE (Online ISSN 2306-0271), where we follow the Open Access strategy for basic publication and EUROSIM member subscription for extended publication; third, SNE contributions are identified by a DOI (Digital Object Identifier) assigned to the publisher ARGESIM (DOI prefix 10.11128), allowing better citation and cross referencing; and fourth, SNE Volume 21 starts with a new layout. Furthermore, we are glad, that Vlatko Ceric, past president of CROSSIM, is providing his algorithmic art as design for SNE Volume 21 cover pages. The artist choose the series 'Compositions' (see below), and each 2011 SNE issue's cover shows a specific algorithmic art print from this series - for SNE 21(1) we have decided for a print of this series in shade of blue – as the new cover layout. Vlatko Ceric's contribution on 'Algorithms, Mathematics and Arts' in this issue gives insight into algorithmic art – a kind of modeling and simulation (invited talk from MATHMOD 2009).*
The contributions in this issue show the broad area of modeling and simulation – from TCP network simulation to room resource management. Furthermore the series of solutions to the ARGESIM Benchmarks are continued – here SNE allows now more pages for comparative solutions, as with the C17 solution in this issue.

I would like to thank all authors for their contributions and all people who are helping to manage the novelties and changes in SNE publication.

Felix Breitenecker, Editor-in-Chief, eic@sne-journal.org

Contents

Reader's Info

Simulation Notes Europe publishes peer reviewed *Technical Notes*, *Short Notes* and *Overview Notes* on developments and trends in modelling and simulation in various areas and in application and theory. Furthermore **SNE** documents the **ARGESIM Benchmarks** on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions (*Benchmark Notes*). Special *Educational Notes* present the use of modelling and simulation in and for education and for e-learning.

SNE is the official membership journal of **EUROSIM**, the Federation of European Simulation Societies. A News Section in **SNE** provides information for **EUROSIM** Simulation Societies and Simulation Groups.

SNE is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With **Online SNE** the publisher **ARGESIM** follows the **Open Access** strategy, allowing download of published contributions for free. Since 2011 **Online SNE** contributions are identified by a DOI (Digital Object Identifier) assigned to the publisher **ARGESIM** (DOI prefix 10.11128). **Print SNE**, colour high-resolution **Online SNE**, full **SNE Archive**, and source codes of the *Benchmark Notes* are available for subscription via membership in a **EUROSIM** society.

→ *www.sne-journal.org*
office@sne-journal.org, eic@sne-journal
SNE Editorial Office, c/o ARGESIM / DWH, Neustiftgasse 57-59, 1070 Vienna, Austria

SNE Editorial Board

Author's Info

Authors are invited to submit contributions which have not been published and have not being considered for publication elsewhere to the SNE Editorial Office. SNE distinguishes different types of contributions (*Notes*):

- *Overview Note* – State-of-the-Art report in a specific area, up to 14 pages, only upon invitation
- *Technical Note* – scientific publication on specific topic in modelling and simulation, 6 – 8 (10) pages
- *Education Note* – modelling and simulation in / for education and e-learning; max. 6 pages
- *Short Note* – recent development on specific topic, max. 4 pages
- *Software Note* – specific implementation with scientific analysis, max 4 pages
- *Benchmark Note* – Solution to an ARGEIM Benchmark; basic solution 2 pages, extended and commented solution 4 pages, comparative solutions on invitation

Interested authors may find further publication and layout information at **SNE's** website → *www.sne-journal.org* (templates for *Notes*, requirements for benchmark solutions, etc.).

SNE Contact & Info

→ *www.sne-journal.org*
✉ *office@sne-journal.org, eic@sne-journal.og*
✉ SNE Editorial Office, c/o ARGESIM / DWH, Neustiftgasse 57-59, 1070 Vienna, Austria

SNE SIMULATION NOTES EUROPE

ISSN SNE Print ISSN 2305-9974, SNE Online ISSN 2306-0271

WEB: → *www.sne-journal.org*, DOI prefix 10.11128/sne

Scope: Technical Notes, Short Notes and Overview Notes on developments and trends in modelling and simulation in various areas and in application and theory; benchmarks and benchmark documentations of ARGESIM Benchmarks on modelling approaches and simulation implementations; modelling and simulation in and for education, simulation-based e-learning; society information and membership information for EUROSIM members (Federation of European Simulation Societies and Groups).

Editor-in-Chief: Felix Breitenecker, Vienna Univ. of Technology, Inst. f. Analysis and Scientific Computing, Div., Math. Modelling and Simulation, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria; ✉ *Felix.Breitenecker@tuwien.ac.at,* ✉ *eic@sne-journal.org*

Layout / Administration: Anna Mathe, ARGESIM /ASIM ✉ *anna.mathe@tuwien.ac.at*, ✉ *office@sne-journal.org*

Print SNE: Grafisches Zentrum, TU Vienna, Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria

Online SNE: DWH Simulation Services, Neustiftgasse 57-59, 1070 Vienna

Publisher: ARGESIM ARBEITSGEMEINSCHAFT SIMULATION NEWS - WORKING COMMITTEE SIMULATION NEWS, Neustiftgasse 57-59, 1070 Vienna, Austria; → *www.argesim.org,* ✉ *info@argesim.org* on behalf of ASIM(→ *www.asim-gi.org* and EUROSIM → *www.eurosim.info*

Algorithms, Mathematics and Art

Vlatko Ceric

Faculty of Economics & Business, University of Zagreb, Trg J.F. Kennedya 6, 10000 Zagreb, Croatia; *vceric@efzg.hr*

Abstract. Algorithms, mathematics and art are interrelated in an art form called algorithmic art. Algorithmic art is visual art generated by algorithms that completely describe creation of images. This kind of art is strongly related with contemporary computer technology, and especially computer programming, as well as with mathematics used in algorithms for image generation. We first discuss two aspects of algorithmic art: the fact that it is based on rational approach of constructing algorithms, and that it involves a strong constraint that image has to be created by an algorithm. We are then describing mathematical influences to art during history, and especially in Renaissance and during the 20th century. Besides we present works of several artists influenced by mathematics and most notably work of M. C. Escher who attempted to visually express various mathematical concepts like infinity, recursion and self-similarity and studied the use of regular periodic divisions of a plane, convergence to a limit and various transformations of shapes. Then we are describing the main characteristics of algorithmic art as well as its most important proponents and their works, and make distinction between visualization of mathematical objects and algorithmic art. Besides we discuss algorithmic art as the form of visual art notation and compare it with musical notation.

Introduction

Mathematics and science had influenced arts from ancient times, and this influence became intensive in Renaissance with artists like Leonardo da Vinci and Albrecht Durer who were studying and using knowledge on perspective, human body proportions, optics and science of colours. Still the most intensive influence of mathematics and science on arts began in recent times, in the 20th century with artists like M. C. Escher, Max Bill and many others. M. C. Escher, a leading figure among artists that was influenced by mathematics, was developing strange perspectives and perspectives on the sphere, and studied the use of tessellations of a plane, convergence to a limit, as well as various transformations of shapes.

Best known mathematical computing approach to art in 20th century is probably due to mathematician Benoit Mandelbrot who invented fractals, a family of self-similar and scale invariant objects that have a simple recursive definition. He illustrated fractals with computer-based visualizations, and these images were so interesting that they soon became extremely popular and stay like this to the present days.

Algorithmic approach to art existed in classical visual arts, and probably the best known example of an artist using algorithmic approach is again M. C. Escher. Many of his artworks are structured in such a way that it is possible to write algorithms that generate them and indeed a number of algorithms and computer programs that generate Escher's work were developed. It is interesting too that works of an op art pioneer Victor Vasarely were created on the basis of a deliberate plan of their structure and colouring, and it can be said that he was using a kind of algorithmic approach to his work.

Algorithms, mathematics and art are interrelated in an art form called algorithmic art. Algorithmic art is visual art generated by algorithms that completely describe creation of images. This kind of art is strongly related with mathematics used in algorithms for image generation as well as with contemporary computer technology, and especially computer programming.

The rest of this paper is organized as follows. In Section 1 we describe influence of constraints on art, while Section 2 discusses rational approach to art by presenting approaches of several well known artists. Section 3 presents rich relations of mathematics and art. Section 4 presents use of computers and mathematics in art that gave rise to such visual structures as fractals. Section 5 presents the role of algorithms in generation of visual art pieces from the technical side, while Section 6 gives an overview of algorithmic art, its history and main proponents.

Section 7 illustrates some algorithmic art methods and techniques, while Section 8 gives conclusions. Author of this paper used part of the text from his earlier paper [2].

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 1-10 | doi: 10.11128/sne.21.on.10037

1 Art and Constraints

Constraints are present in all aspects in life, and they usually limit humans in accomplishing their goals. Art itself is usually related with freedom, or lack of constraints. However, is it really true that constraints complicate or even make impossible artistic creation? Famous Russian composer Igor Stravinsky wrote following about constraints in art in his essay 'The Poetics of Music. [12]: My freedom thus consists in my moving about within the narrow frame that I have assigned myself for each one of my undertakings'. And: 'I shall go even further: my freedom will be so much the greater and more meaningful the more narrowly I limit my field of action and the more I surround myself with obstacles.

Decades later another artist, British painter and computer artist James Faure Walker in his book 'Painting the Digital River' [14] wrote about algorithmic art: 'Its strength lie in its limitations.', and continues about comparison of work of algorithmic artists with other digital artists: 'When shown alongside the photo collages and 'processed' images in digital exhibitions, the work of these artists does look more assured'.

2 Art and Rationality

Creating of artworks is not (only) an intuitive activity related mostly with emotions and inspiration. During the history of art numerous techniques were devised in order to make the faithful representation of objects or to obtain particular artistic effects.

Perspective is example of the analytic achievement that enabled projection of three-dimensional objects onto a two-dimensional surface. Linear perspective was used by Islamic artists in the Middle Ages and was rediscovered in Italy in the 15th century, during Renaissance. Golden ratio, so popular in visual arts and architecture, is known from ancient times and was largely employed in Renaissance.

Various optical advices like *camera obscura* and *camera lucida* helped artists to make accurate representation of objects they draw or paint. A recent study of British painter David Hockney [6] showed that these kinds of tools were used earlier and in much higher degree than it was previously thought. He found that some of the world's most famous painters like Ingres, Velazquez or Caravaggio used optics and lenses in creating their masterpieces.

It is probably needles to say that Hockney considers that the use of optics does not diminish the value of artistic achievement since it is the artist's hand and creative vision that produce a work of art. Pointillism and cubism are also based on rational approach. Thus Georges-Pierre Seurat in late 19th century, using knowledge of colour theorists on scientific approach to painting, developed pointillism as an analytical technique in which painting is done using discrete colour dots with complementary colours being next to each other. Cubists like Pablo Picasso and Georgeos Braque devised in the beginning of 20th century technique of painting objects from several points of view that provided a novel view on portrayed objects.

A remarkable evidence of rational method of creation of an artwork is given in Edgar Alan Poe's essay 'The Philosophy of Composition' [11]. In this text E. A. Poe gives reconstruction of steps of development of his famous poem 'The Raven'. He literary wrote that '⋯ no one point in its composition is referable either to accident or to intuition', and that 'the work proceeded step by step, to its completion, with the precision and rigid consequence of a mathematical problem'. In this text Poe described how he decided about the poem length, the effect to be expressed by the poem, the refrain and its sound ('Nevermore'), a creature repeating his sound (Raven), etc.

It is time now to concentrate on relation of art with mathematics, one of strongest foundations of human rational thought.

3 Mathematics and Art

Mathematics and science had influenced arts from ancient times: e.g. facade of Parthenon built in the fifth century B.C. contains a number of proportions such as the golden ratio and the square root of 2, while periodic patterns frequently occur in Islamic and Moorish ornaments. However, it was only in Renaissance that artists began using science and mathematics more intensively. They were influenced by the rediscovery of Greek philosophy and were convinced that mathematics was the true essence of the physical world and that the universe was ordered and explainable in geometric terms [8].

Thus for example two famous Renaissance artists, Leonardo da Vinci and Albrecht Durer were studying and using knowledge on perspective, human body proportions, optics and science of colours in creation of their works.

However, the most intensive influence of mathematics and science on arts began in recent times, in the 20th century. Importance of mathematics for artists is emphasized by Swiss artist, architect and graphical designer Max Bill in his well-known text on the mathematical approach in contemporary art [1]. He wrote that he is convinced that 'it is possible to evolve a new form of art in which the artist's work could be founded to quite a substantial degree on a mathematical line of approach to its content'. The best known 20th century artist that was influenced by mathematics was by no means Dutch graphic artist M. C. Escher. Although he was not mathematically trained he was interested in mathematics. In mid 1930th he started to develop mathematical approach to structure his prints, and in construction of his famous 'impossible objects' prints he was influenced by mathematician Roger Penrose. In his works Escher developed strange perspectives and perspectives on the sphere and studied the use of tessellations of a plane (i.e. regular periodic divisions of a plane), convergence to a limit, various transformations of shapes, mathematical concepts of infinity, recursion and self-similarity, and he also used polyhedra and Mobius strip in his works. The first stage in process of creation of his works Escher devoted to developing a geometric model of image.

In his work with tessellations Escher used both regular and irregular tessellations and he especially liked 'metamorphoses' in which the shapes changed. He exploited basic patterns, triangles, squares and hexagons, applying reflections, translations, and rotations to obtain a greater variety of patterns and also changes the basic shapes into animals or birds that sometimes even left the plane and go to the third dimension. Escher wrote about his work in this field in the lecture about the 'Regular division of a plane' [4].

The shape of space fascinated Escher and he created many representations of a non Euclidean hyperbolic space (like in Circle Limit III). Escher also had interest in visual aspects of topology, a branch of mathematics working with properties of space that remain unaffected by distortions (like stretching or bending), and was making various graphical representations of Mobius strip.

Several of his works present intermingling of inner and outer spaces that sometimes include singularities (like in Print Gallery), again revealing his interest in the topology of space.

One of Escher's favourite topics was visualisation of paradoxes of space logic and geometry that was often displayed in his perspective drawings. In perspective drawings one chooses vanishing points which represent for the eye the points at infinity. Escher introduced unusual vanishing points and was using them in positioning of objects of the scene. This gave the effect that up-down and left-right orientation change with the way the viewer's eye look at them (like in High and Low). Another type of his paradoxical drawings relies on the brain's persistence in using visual clues to construct a three-dimensional object from a two-dimensional representation (like Waterfall). Douglas R. Hofstadter in his famous book Godel, Escher, Bach: An Eternal Golden Braid [7] draw attention to Escher's use of self-reference, the concept that seems to be the key for understanding intelligence and consciousness.

Thus in Escher's work Drawing Hands each of the two hands draw another one, while in other works self-reference is not so direct or appears as mutual mirroring of several worlds (like in Three Spheres II where three spheres reflects one another and the world around them in their spherical mirrors). Among interesting mathematically based artistic objects are sculptures based on mathematical objects, as well as different type of perspectives painted on the sphere. Helaman Ferguson is well known mathematically oriented sculptor who is using mathematics as a design language for his sculptures [3]. His sculptures in stone and bronze have been named 'theorems in bronze and stone'. They can be found on many public spaces and were shown in a number of exhibitions. It is interesting that Ferguson is also an internationally known mathematician whose algorithm has been listed as one of the top ten in the twentieth century.

One of the leading artists working with perspectives painted on a sphere is Dick Termes. He developed his own six-point perspective system that he calls Termespheres, being 'visual environments painted on the surface of spheres that hang and rotate from ceiling motors' [3]. These perspectives show everything that one can see from one point looking in all directions.

It is well known that geometries that fit on the sphere are totally different from geometries that fit on the plane, and they are related to the study of polyhedra. Six-point perspective is related to the cubical polyhedron that extends in all six directions of the space.

4 Mathematics and Computer Technology in Art

Development of computer technology and particularly graphic cards and software in the second part of the 20th century made an enormous influence on the ability of visualizing mathematics and using these visualizations in design and art.

One of the best known mathematical approaches created in the era of computer graphic and exceptionally suited for generating attractive visual structures are fractals, a family of self-similar (they appear similar at all levels of magnification) and scale invariant objects that have a simple recursive definition. Fractals were invented by a mathematician Benoit Mandelbrot who illustrated them with computer-based visualizations, and these images were so interesting that they soon became extremely popular.

Some interesting examples of fractals are Mandelbrot set and Julia set. Mandelbrot set is defined by a family of complex quadratic polynomials. It is a subset of a complex plane, i.e. a set of complex numbers or points in the complex plane the boundary of which does not simplify at any magnification. A picture of the Mandelbrot set can be made by different type of colouring, e.g. by colouring all the points which belong to the Mandelbrot set black, and all other points white. In the centre of the Mandelbrot set image is the characteristic large cardioidshaped region. Various algorithms are used for producing computer drawings of the Mandelbrot set. They are quite simple, and e.g. the so called 'escape time algorithm' has only about a dozen of steps. This is a good example of a simple mathematical model generating complex results.

Some of the other mathematically based methods used for generation of attractive images are genetic algorithms and cellular automata. A number of interesting examples of cellular automata visualization are presented in the Stephen Wolfram's book 'A New Kind of Science' [15].

Cellular automata are also used for generation of random numbers for the Mathematica software, which is again an excellent example of complex structures generated by a simple algorithm.

As can be seen, a lot of work in this field has been done with the goal of visualizing mathematical objects. This approach is certainly valuable for the sake of research in mathematics, but it doesn't necessarily lead to interesting visual solutions to the so called fine art or 'high art' [3]. Too much precision, symmetry or repetition doesn't have much chance to be regarded as art. However, mathematical approach can inspire artists to use their imagination and creativity and to use it in the process of creation artworks.

Quite a different approach is to use mathematics as a tool in developing artists' ideas. Following this approach artists typically use rather simple mathematics that enable accomplishment of their idea but is of no interest from the mathematical point of view. In such case a lot of experimenting is typically needed in order to achieve visually attractive result. It is this kind of approach that is mostly used in algorithmic art.

5 Algorithms in Art

Algorithmic art is based on generation of visual artworks using algorithms, i.e. precise defined procedures that computer executes step by step. Algorithms are implemented in computer programs coded in some programming language. Algorithm, or computer program implementing it, describe the process of generation of image that we can see either on the screen or as printed. Program for image generation contains the author's idea about the image as well as the technique by which this idea is transformed into an image. It also has to define which graphical elements and their structures should be generated (straight or curved lines, shapes, a group of elements with a specified structure, etc.), what are values of their parameters (e.g. position of a rectangle, its dimensions and its elevation toward axes), colours of lines and shapes, etc. Algorithms use computational structures such as loops, subprograms and recursion, as well as various mathematical expressions.

Algorithmic artists (also called 'algorists') as a rule don't use special graphical software but rather general program languages that enable drawing basic graphical elements like line, circle or rectangle. Such approach requires more work but offers much more flexibility and freedom in expression. Using general program languages algorists develop their own software in which they embed their artistic ideas. After the image is generated no further intervention with image processing tools like Photoshop is done. Interventions for changing image are done merely by changing the algorithm (i.e. the corresponding computer program). The algorithms could also be executed by human hand, and they have been executed by humans before computers came into arena (and sometimes even after that).

However, the evident advantage of computer and its output units is in enormous processing power of contemporary computers and precision of its output units. Because of that they can execute very complex algorithms that include a huge number of graphical operations in a rather short time and without mistake. This enables routine execution of algorithms that would require years to be executed by hand. All this gives computer technology a role of enabler of a real revolution in algorithmic art.

Another particularly important fact is that analysis of consequences of changes in the algorithm, i.e. in the corresponding computer program, is considerably simpler and faster in comparison with repeating of manual construction of somewhat changed image. This enables intensive experimentation needed for discovering the most appropriate artistic result.

Algorithmic approach to art existed in classical visual arts. Probably the best known example is M. C. Escher whose mathematical inclination was already mentioned here. Many of his artworks are structured in such a way that it is possible to write algorithms (computer programs) that generate them. This was indeed done by several authors from the computer science field. One example is Michael Trott who in his book 'Mathematica Guidebook for Graphics' [13] constructed a computer program that generates image of the Escher's well-known lithograph Reptiles.

Works of op art pioneer Victor Vasarely were created on the basis of a deliberate plan of their structure and colouring, as can be seen from Vasarely's blueprints. Robert C. Morgan in his book .Vasarely. [10] wrote: .He created algorithmic systems that in many ways parallel the development of the computer., and also: .Vasarely made his own visual software in the form of mathematics, referring to the blueprints of his works as programs..

Since algorithmic art consists of generation of images on the basis of algorithms, algorithms can be viewed as a notation, and notation is something that music has but visual arts in general miss. There are further parallels of algorithmic notation with music. First, is not the author but the computer that executes the algorithm. Moreover, algorithm can be executed by different computers, using different operation systems, programmed in different software and presented by diverse output devices with various resolutions and other characteristics.

6 Algorithmic Art

We review here some of the key events in the history of computer generated art and present some of the best known authors from the field of algorithmic art. First experiments with usage of computers in visual art were performed in late 1950th and early 1960th. More intensive development of algorithmic art began in early 1960th, and pioneers were two researchers in the area of computer graphics, Dr. A. Michael Noll from USA and Prof. Frieder Nake from Germany. Dr. Noll also wrote the first computer generated ballet in 1965. In 1963 first computer art competition was held, sponsored by the U.S. journal 'Computers and Automation'. In 1965 first computer art exhibitions were held at Technische Hochschule in Stuttgart and at Howard Wise Gallery in New York. In 1971 Herbert Franke published the book 'Computer Graphics - Computer Art' [5] in which he described the tools and principles of computer generated graphics, its artistic applications, history of computer art and its theoretical foundations. In 1976 Ruth Leavitt edited the book 'The Computer in the Visual Arts' [9] which presented texts of three dozen of early computer artists, and among them algorithmic artists like Vera Molnar, Manfred Mohr, Edward Zajec and Charles Csuri. Several authors proposed the name 'algorithmic art', and Benois Mandelbrot wrote that '⋯ astonishingly complex and beautiful graphics can be generated by surprisingly plain algorithms. Hence the term algorithmic art, which I use at present' [3].

Some of the best known algorithmic artists we present here are Vera Molnar, Jean-Pierre Hebert, Roman Verostko, Manfred Mohr and Charles Csuri. What is particularly interesting is their background and how they approach algorithmic art.

Vera Molnar was born in Budapest, Hungary in 1924. She was a classical artist using geometric themes, and her aim was to create valuable works of art in a conscious way. In 1968 she began to use a computer to assist her. She started to create a series of abstract works generated from a procedure in which simple geometrical shapes and their combination were successively altered in small steps, and she wanted to find which alteration lead to the aesthetically most appealing result.

Jean-Pierre Hebert was born in Calais, France, in 1939. He is educated as a classic artist and he started to create digital conceptual algorithmic art in 1974, while in 1979 he engaged in intensive experimentations in creating plotter drawings with computers. His professional background in engineering and computer programming languages originally led to a consulting career, though he maintained his interests in arts and mathematics. In 1983 he moved to United States and ceased consulting to devote himself fully to art. Hebert is experimenting with various media from paper, glass and mirror to wood, steel, and sand. He is also working on digital video animations, computerized kinetic sculptures, and organic algorithmic drawing devices such as paint-dripping pendulums. In 2003 Hebert was appointed Artist in Residence at the Kavli Institute for Theoretical Physics at University of California at Santa Barbara.

Roman Verostko was born in Tarrs, Pennsylvania, in 1929. He was educated as artist and historian, but spent 16 years as a Benedictine monk. In the beginning of 1980th, after 30 years of work as a traditional visual artist, he started working with algorithmic graphics using classical pen plotter. He is one of the most original algorithmic artists, and besides a voluminous art production he also wrote a number of essays on algorithmic art and on his own methods of work. He is using the term epigenetic to describe the algorithmic procedures in his work, identifying the biological analogues for art works executed with algorithms created by artists. In his own words: 'The epigenesis of organisms is the process whereby a mature life form grows from its seed. In this analogy the software may be viewed as genotype or the seed that contains all the information necessary for growing the mature form.'

Manfred Mohr was born in Pforzheim, Germany, in 1938, He was educated as an artist, but he also used to play tenor-saxophone and oboe in a jazz group. In 1969 he made his first drawings with a computer. From early 1970ties he began his artistic work on a cube structure, and in 1977 he begins to work with the 4-D hypercube and graph theory. In late 1980ths he extends work to the 5-D and 6-D hypercube. In 1998, after using exclusively black and white colours for three decades, he starts using colours to show the complexity of the work through differentiation. In his own words: 'What interests me, are the two-dimensional signs (graphics) and their visual ambiguity resulting from the projection of the lines of the cubes from higher dimensions into twodimensions'. Manfred Mohr received a number of international prizes for his work. Since 1981 he lives and works in New York.

Charles Csuri is an artist and computer graphics pioneer. He joined the faculty at the Ohio State University and also exhibited his paintings in New York City from 1955-1965. In 1964 Csuri became interested in the digital computer and experimented with computer graphics technology, while in 1965 he began creating computer animated films. His research activity in computer animation and graphics has received international recognition and he directed basic research in computer graphics for over twenty years. The results of the research have been applied to flight simulators, computer-aided design, visualization of scientific phenomena, magnetic resonance imaging, education for the deaf, architecture, and special effects for television and films. He cofounded Cranston/ Csuri Productions which produced animation for all three major U.S. television networks. Csuri exhibited at the 42nd Biennale de Venezia, Italy, 1986. Ars Electronica, a major international competition on computers and the arts held in Austria each year, awarded him prizes in 1989 and 1990.

7 Some Algorithmic Art Techniques

In order to provide better understanding about generation of algorithmic art we will illustrate some algorithmic art methods and techniques, and these will be techniques used by the author of this text since it is rather difficult to describe approaches and techniques used by other artists because of the lack of appropriate information.

The author of this text uses several methods for generation of algorithmic artworks: let us roughly call them constructivist approach, mathematical modelling approach, and digital manipulation of photographs (or in fact any kind of images). There are no strict boundaries between these methods, and especially use of mathematics is definitely not limited to mathematical modelling approach but appears in two other categories too. Here is a description of these methods along with a few examples of images generated with each method.

We will start with a constructivist approach as the technically simplest one. This approach is based on rather simple preconceived structures that can be rather straight described by algorithms. Example of such structures is a network of regularly positioned squares (or some other geometric figures) that can have different sizes and colours. Algorithms for generation of such structures are in principle easy to formulate and programs developed on basis of these algorithms are primarily useful in fast analysis of numerous possible variations of image form, thus providing a rich source of variants for a selection of visually most interesting images. Besides, algorithmic approach power also lies in enabling generation of extremely precise images. Let us demonstrate this approach by example of a graphics from the series 'Cartesian rhythm' based on a matrix of squares, as can be seen on Figure 1.

Algorithmic approach enables one to study influence of various parameters on generated image. One can e.g. analyse how many groups of squares of different sizes give the best visual result. In this case it was found that the best visual rhythm is obtained when only two groups of squares of different sizes are used. After that one would like to know in which manner these two types of squares should be spatially intermixed, what is the optimum ratio of sizes of two squares and what is the optimum distance between squares. After resolving these questions follows the study of colours to be used for a pleasing aesthetical result: should all squares be of the same colour or should e.g. smaller squares have different colour than bigger ones.

As can be seen on Figure 11 for graphics 'Cartesian rhythm 11' it was decided that the colour of both groups of squares should vary in such a way that a particular variant of blue colour from the central square looses its colour in a centrally symmetric manner and becomes darker and darker toward edges, finally disappearing on a black background.

This radial fall of colour intensity of squares leads to an interesting circular effect obtained by using squares. Somewhat more complicated image also based on a matrix of squares is graphics 'Order 1' from the series 'Order and Chaos' shown in Figure 2.

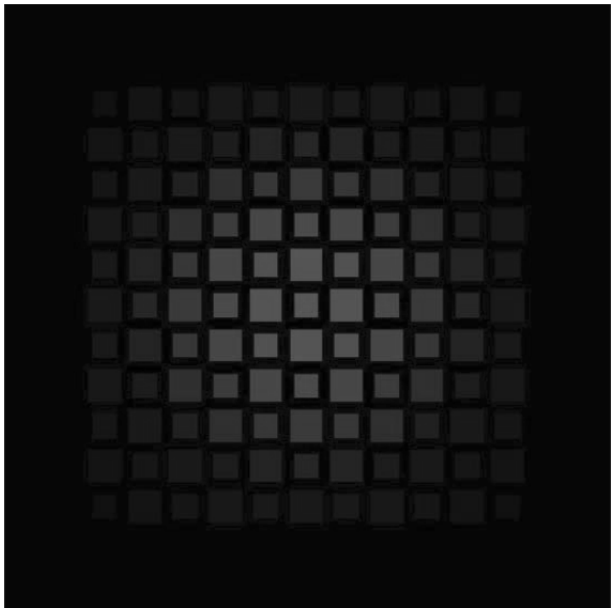


Figure 1. Example of a constructivist approach – graphics 'Cartesian rhythm 11' by V. Ceric.

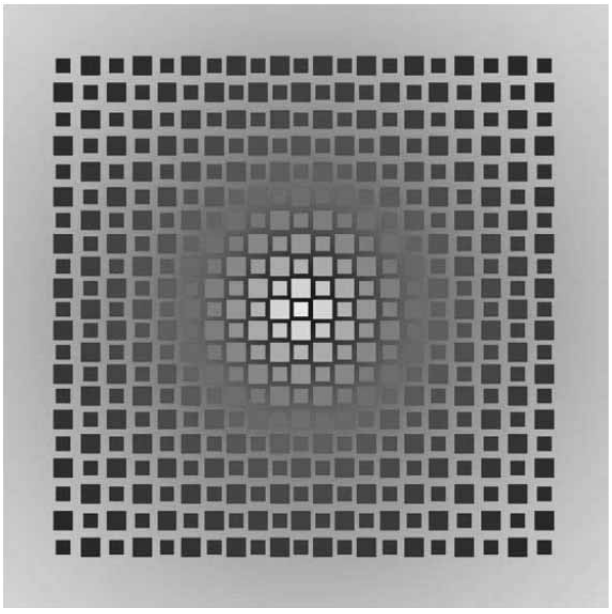


Figure 2. Example of a mixture of constructivist and mathematical modelling approach – graphics 'Order 1' by Vlatko Ceric.

This image can be regarded as a mixture of constructivist and mathematical modelling approach. Two sets of squares of different sizes are used again, but now the colour of both sets of squares is from almost white in the centre to dark brown on the periphery. Colour of the background is changing in opposite direction, from dark blue in the centre to light blue on the periphery. As a consequence of such behaviour of colours the distance between squares in the centre should be smaller than distance between squares on the periphery of the plane (and this distance should behave as a centrally symmetric function) because in that case more densely populated brighter centrally positioned squares would together give stronger shining effect. Such arrangement of squares also gives a more dynamical appearance of the whole image. So we see that algorithm that generates this image requires somewhat more mathematics than one for generation of *Cartesian rhythm 11*.

Mathematical modelling approach starts from the rough idea about how the image should look like and requires intensive experimenting in order to find an appropriate mathematical model that gives interesting visual result. So it should be clear that in this approach mathematical modelling is not used in order to visualize mathematical objects but rather to obtain interesting images by more intuitive approach.

We will demonstrate this approach by example of the graphics from the series 'Nexus' shown on Figure 3. The technique used in generation of works from this series was often used in 20th century by geometrically inclined artists. This technique is based on the division of squares on two right triangles with two equal sides.

There are just two ways this division can be done, by dividing triangle with a diagonal going from lower left vertex to upper right vertex or from lower right vertex to upper left vertex. If we use just two colours for colouring triangles (e.g. black and white) then there are evidently just four different divisions of a square. Although such small number of possibilities may seem insufficient for producing interesting images, it has a potential for complexity if image contains enough square divisions.

However, if one works in a traditional manner then one can either use a small number of square divisions in order to be able to construct a pleasing image. If larger number of square divisions were used then traditional artists were only able to employ simple rules governing which of four divisions will be used for a square on a particular location in the network - otherwise it would be an enormous job to calculate results of a more complex formula by hand and apply it.

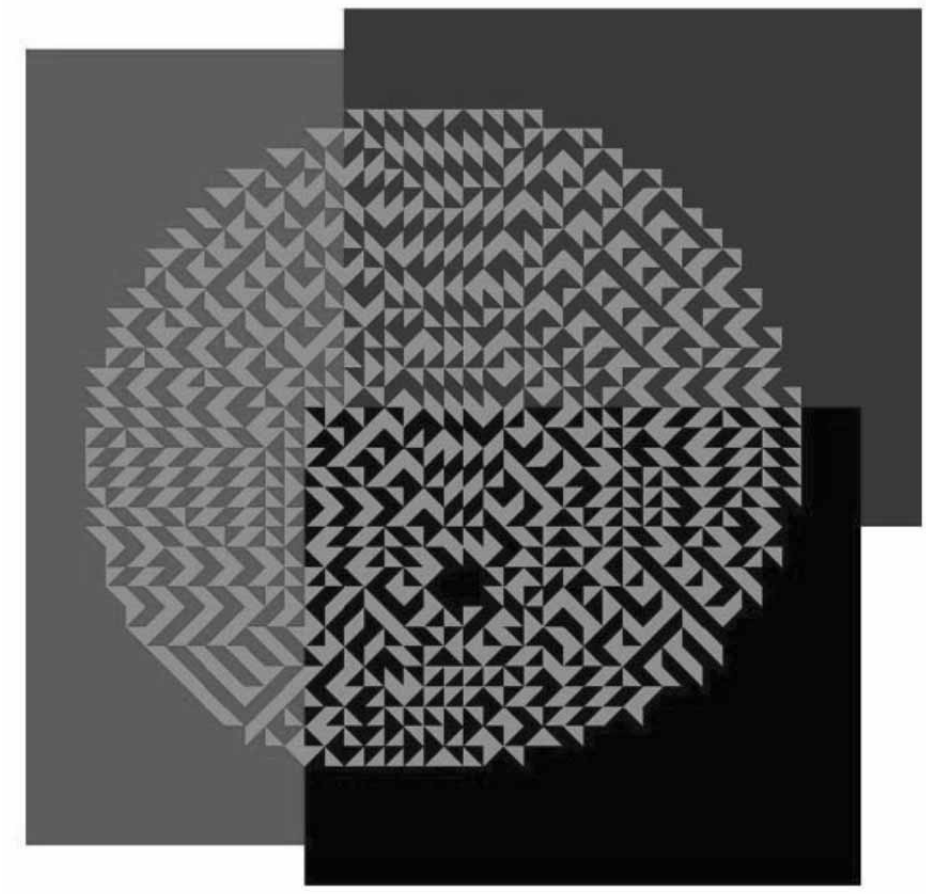


Figure 3. Example of a mathematical modelling approach – graphics 'Nexus 1' by V. Ceric.

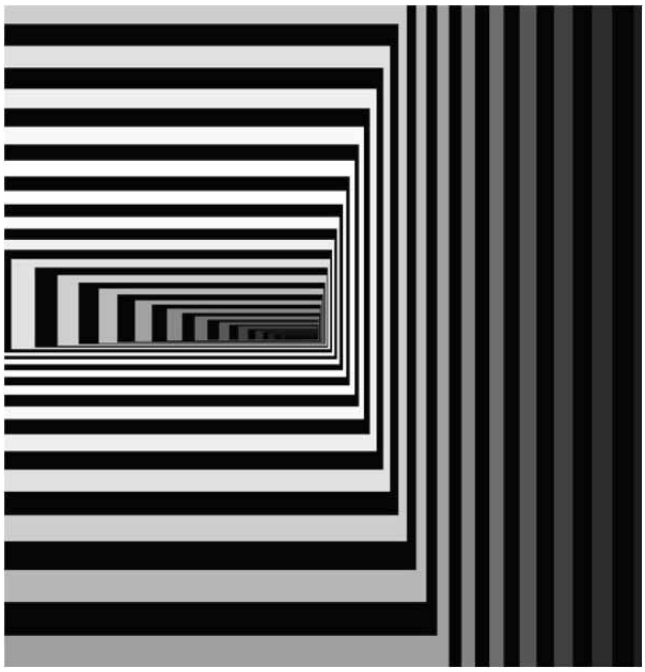


Figure 4. Another example of a mathematical modelling approach – graphics 'Unclassified objects 14' by V. Ceric.

Now, a power of mathematical and algorithmic approach married with the use of high-speed processors enables one to work with a large number of square divisions and simultaneously use as complex mathematical relations as one wish in order to obtain interesting visual structures. The basic idea behind Nexus series was that the type of square division would depend on the distance of the particular square from one or more points on the plane.

So the whole complexity of Nexus graphics structure was translated into the selection of the mathematical form of distance of the square, and this distance was defined as some function of distances of the square in some location from the two selected points in a plane. Exact form of this function was determined after analysing visual structures obtained in numerous experiments. As can be seen, the particular selection of the distance function in Nexus 1 is giving interesting and nontrivial graphics giving the impression of some kind of circular flow of triangles on the plane.

Another quite different type of mathematical modelling approach was used in generation of graphics from the 'Unclassified objects' series, one example of which is shown in Figure 4.

These geometrical structures in strong black and gray transitional tones are based on a relatively simple idea of successive covering of a plane with rectangles, where positions of rectangles in the plane, their shapes and sizes as well as level of gray color applied to them are governed by mathematical models based on parametric equations that use periodic functions. Interplay of sharp geometrical structures and black and gray tones leads to intensive feeling of volume of these works. For generation of some images in this series only a few dozen of iterations of rectangle generation were needed, while for others hundreds of iterations of rectangle generation were performed. Careful selection of initial and final iteration number is required in order to produce a clean, closed and interesting image. However, plentiful experiments have to be carried out in order to achieve this goal.

Digital manipulation of photographs (or in fact any other images) is using algorithmic approach for manipulation of pixels of a photograph. Because of an enormous richness of colour spectrum and structure of the world surrounding us, richness that surpasses almost anything that can be created artificially, the images obtained in such a way can be quite complex and interesting. Besides that, the author discovered that this approach enables transformation of figurative input photograph into an abstract generated image. We will illustrate one specific digital manipulation approach by example of the graphics from the series 'Spectral variations' shown on Figure 5.

In the first pre-processing stage we transform input photograph (or any kind of input image) into a twodimensional pixel matrix using the Digital Image Processing add-on to the Mathematica software. After that we divide this matrix into n x n smaller pixel submatrices made from the corresponding parts of the input photograph. Then we use Mathematica to transform pixels of all created sub-matrices using algorithms based on some mathematical procedures that enable displacement of sub-matrices pixels to different locations (coordinates) in the corresponding sub-matrix. In the final post-processing stage we again use Digital Image Processing add-on in order to compose a final image from the n x n transformed parts of the original photograph.

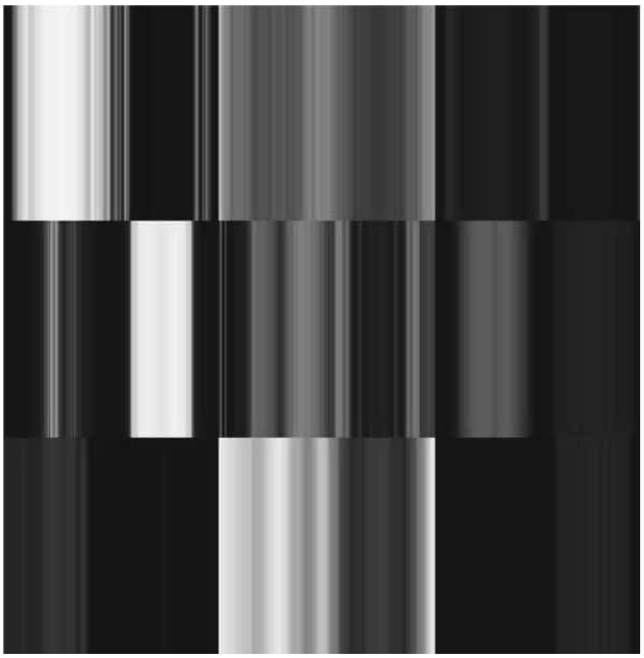


Figure 5. Example of a mathematical modelling approach graphics 'Spectral variations 10' by V. Ceric.

For the 'Spectral variations' series input photograph was a small 420 x 420 pixels photograph of a face of the author, so we can say that graphics from this series present the very author of the work.

8 Conclusions

We have seen that not only mathematics and algorithms have influenced art for a long period of time, and particularly in the 20th century, but also that constraints in general can be fruitful in forcing artist to concentrate his creative forces and search for the artistic possibilities hidden in particular constraint. We also realized that rational approach is not so rare in art and that it has the power to open doors of art.

Algorithmic art itself, a discipline that unites mathematics, computing and art, has a rather specific character since the author has to possess rational abilities required to compose the algorithm and write the corresponding computer code correctly from both the syntactic and semantic point of view, but he also needs intuitive and aesthetic abilities required to select visually promising alternatives.

The author of algorithmic artworks constantly evaluates visual outputs obtained during experimentation with the program and on the basis of these evaluations makes changes to the program until satisfactory visual results are obtained. Undoubtedly, experimentation is also present and important in a traditional art, but algorithmic art is using its full potential.

References

[1] M. Bill. *The Mathematical Way of Thinking in the Visual Art of Our Time*. Werk, No. 3 (1949).

[2] V. Ceric. *Algorithmic Art: Technology, Mathematics and Art*. In: Proc. 30th International ConferenceInformation Technology Interfaces ITI'2008, Cavtat, 2008, 75-81.

[3] M. Emmer , editor. *The Visual Mind: Art and Mathematics.* The MIT Press, 1995.

[4] M. C. Escher. *Escher on Escher: Exploring the Infinite*. New York: Harry N. Abrams, Inc., 1989.

[5] H. W. Franke. *Computer Graphics –Computer Art*. Phaidon Press Ltd., 1971.

[6] D. Hockney. *Secret Knowledge: Rediscovering the Lost Techniques of the Old Masters*. Thames & Hudson, 2001.

[7] D. R Hofstädter. *Gödel, Escher, Bach: An Eternal Golden Braid*. Basic Books, 1979.

[8] M. Kline. *Mathematics in Western Culture*. Oxford University Press, 1964.

[9] R. Leavitt. *The Computer in the Visual Arts, Harmony Books.* 1976.

[10] R. C. Morgan. *Vasarely*. New York: George Braziller, Inc., 2004.

[11] E. A. Poe. *The Philosophy of a Composition*. Graham's Magazine, April 1846.

[12] I. Stravinsky. *The Poetics of Music – in the form of six lessons*. Oxford University Press, 1947.

[13] M. Trott. *Mathematica Guidebook for Graphics*. Springer, 2004.

[14] J. F. Walker. *Painting the Digital River*. Upper Saddle River, NJ: Prentice Hall, 2006

[15] S. Wolfram. *A New Kind of Science*. Wolfram Media; 2002.

Invited MATHMOD 2009, February 2009
Invited Submission: September 2010
Accepted: October 10, 2010

The <morespace> Project: Modelling and Simulation of Room Management and Schedule Planning at University by Combining DEVS and Agent-based Approaches

Shabnam Tauböck [1] *, Felix Breitenecker[1], Dietmar Wiegand[2], Nikolas Popper[3]

[1] Inst. for Analysis and Scientific Computing, Vienna University of Technology, Wiedner Haupstraße 8-10, 1040 Vienna
[2] RED Real Estate Development, Vienna University of Technology, Vienna, Austria; *shabnam.tauboeck@tuwien.ac.at
[3] dwh simulation services, Neustiftgasse 57-59, 1060 Vienna, Austria

Abstract In this work an approach to integrating agent based modeling into a discrete event simulation system is developed and tested in the course of a project done at the Vienna University of Technology.
Discrete Event Simulation can be efficient and fast but people's behavior and movement is not easy to model. It would be convenient to develop a model, where both systems - discrete event and agent based - coexist. During the last years the feasibility of finding an equivalent discrete event model for any agent based model has already been discussed and formally proven in several publications. This implies that it should be possible to integrate any agent that is part of an agent based system into a discrete event system.
An application of this theory is introduced in this work, showing in detail how the agent based approach was integrated in the discrete event simulation environment Enterprise Dynamics. The <morespace> project supports and controls lecture room management and lecture schedule planning for many curriculas at Vienna University of Technology.

Introduction - The <morespace> Project

The project MoreSpace was launched to develop a software tool to assist the department of Gebäude und Technik during the planning phase of "University2015", a project of the Vienna University of Technology (VUT) to renovate all university buildings and to improve the existing infrastructure and the inherent processes by determining and evaluating the (spatial) resources required. The team responsible for this project used static methods to calculate an assessment of the number of square meters each faculty of the VUT needs for lectures and other student related activities.

This calculation was based on the number of students that took exams and the hours of teaching for each faculty. The resulting numbers did show the need of square meters required to accommodate the number of student during the lectures but did not take into consideration how these square meters are used over time. One thing that complicates things considerably is the fact that lectures at the VUT do not take place one after the other in a strict pattern as it is done at schools or colleges but are set at times favored by the lecturer. This results in a weekly timetable for the student that contains times where several lectures may be very close to each other, even overlapping, and times where no lectures at all are taking place, leaving a big time gap. Over the whole of the VUT one can say that there are certain times during the week that are more or less preferred by many lecturers, resulting in a high demand of rooms during certain time periods, where other times – i.e. Friday afternoons – are not that popular. The fact that the required amount of square meters is available, does not mean it is used in a way that ensures that it will be indeed possible to acquire a room for a lecture at a certain time.

The idea to develop a simulation was born, a tool to reproduce the situation as it currently is concerning the lectures and their demand on room and experiment with the room structure. Opposite to several other approaches used to solve the problem of limited space resources at a university that quite often focus on the timetabling problem [4, 5] this approach focuses mainly on the facility management point of view [6]. The problem of generating a timetable for lectures and courses is not the center of interest for this simulation tool. It uses the existing timetable as a basis to evaluate the current situation according to room utilization and room capacity utilization. It also identifies potential solutions that may result

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 11-20 | doi: 10.11128/sne.21.tn.10039

in a room assignment that frees space that seemed to be occupied. Possible modifications that can be generated automatically to the given schedule are time shifts within intervals that can be set by the user and the splitting of a lecture into two parallel events. Further changes to the timetable can only be done with user interactivity.

One major condition to be considered that proved to be a major constraint was to not decrease the quality of teaching. That means that the impact of possible modifications to the time schedule on the students has to be reflected in the simulation result. Instead of trying to generate a timetable to increase the utility of rooms the emphasis of this work lays in the analysis of the space management and the assignment of rooms to lectures is modeled in great detail. Using discrete event simulation in the field of facility management is still a relatively new concept [7].

The first step to developing MoreSpace was to use the data of two of the eight faculties of VUT and test if the main building on Karlsplatz as it was planned would be able to hold all lectures of these two faculties. To be able to experiment with the room structure was quickly advanced by the possibility to use different strategies for the space management. It proved to make a big difference which rules are used in the booking process.

This shows very nicely the analogy to the classical application of DEVS, the simulation of logistic processes in production lines or manufacturing facilities where the order of processing may become crucial for production times and the feasibility of manufacturing certain quotas in time. Here simulation is one of the most evident ways of analyzing and enhancing the existing system.

The classical discrete event simulator Enterprise Dynamics was chosen to develop the dynamic simulation including features like different possibilities of room selection, different management of the resources, variability of classes, class structures and number of students - only to mention a few of the features. This discrete dynamic simulator was combined with - and is guided by - methods and procedures of the real estate management like business process models.

In the course of developing the first studies the working group identified two main problems that are different to tasks which are normally solved by classical discrete event models. On one hand the interfaces to data collection and booking system had to be improved and standardized. Booking features and also the representation of data was not sufficiently optimized neither to the needs of the simulation tool that has to be implemented nor to the needs of the future booking system that has to be simulated. On the other hand the buildings of the Vienna University of Technology – as the university will not move out of the city of Vienna – remain very scattered over several districts of the city. Therefore a simulation of room booking and facility management has to be aware the problem of differently structured buildings within the university and therefore consider travelling times, sometimes even between classrooms within the same building.

1 Basics DEVS and AB

Considering the basic formalism of DEVS [1] a Discrete Event System Specification (DEVS) is a ture $M = (X, S, Y, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta)$, where X is the set of input values, S the set of states, Y the set of output values, δ_{int} the internal transition function, δ_{ext} the external transition function, δ_{con} the confluent transition function, λ the output function, ta the time advance function. Several of this so called atomic models can be put together to form a coupled model.

An agent based model [2] is defined as a tuple $\langle A, E \rangle$ where A is a set of agents with $A = \cup a^k$ and $1 \leq k \leq N_{agents}$ and E is the Environment. The agent k itself is defined as a function $a^k: R_S^k \rightarrow \Lambda^k$; an Environment E is defined as tuple $\langle \Sigma, \tau \rangle$ where Σ is the system state, $\tau: R_\Lambda \rightarrow \Sigma$ is a state transformer function that changes the system state based on $r_i^k \in R_\Lambda$ with:

- s_j^k is the j^{th} set of variables that is seen by agent k where $1 \leq k \leq N_{agents}$
- α_j^k is the j^{th} action done by agent k in response to a set of state variables s_j^k
- $\Lambda^k = \cup \alpha_j^k$ all actions done by agent k
- r_i^k, the i^{th} run of agent a^k, is the i^{th} sequence of interleaved $s_0^k, \alpha_0^k, s_1^k, \alpha_1^k, \ldots$
- $R^k = \cup r_i^k$, the set of runs of agent k, where $1 \leq i \leq N_{runs}^k$
- $R_S^k = R^k$ that ends with an s_j^k

During the last years the feasibility of finding an equivalent discrete event model for any agent based model that conforms to the specification given above has been discussed and shown in several publications [3]. This implies that every agent that is part of such an agent based system can be integrated in a discrete event system.

It is obvious that both modeling techniques have their advantages and drawbacks. Discrete Event Simulation is known to be efficient and fast as long as the concept of event driven time steps is able to use its advantage of jumping over time intervals where no changes to the system state occur and only update the system elements of the time points where events are schedules or triggered. People's behavior and movement is hard to model in such a system. Movement i.e. is usually not along a certain foretold line – people do tend to take the shortest route from A to B but the easiest way that seems to be free of obstacles and decisions between different ways are due to individual personal preferences – some people prefer to take the stairs, some rather wait for the elevator, some take on the longer walk to the escalator to save themselves from having to take the stairs. The goal here would be to develop a model, where both systems - discrete event and agent based - coexist.

In this project the discrete event simulation environment Enterprise Dynamics (ED) is used to develop the simulation model, which is designed to develop discrete event simulation systems. The formalism shown by DEVS is very much reflected in the basic make up of ED. Basic elements called atoms would refer to the atomic models described by DEVS. They are all identical in their basic structure, their behavior defined by the transition functions. A simulation model is build by using atoms again, again a concept conform to the coupled models in DEVS. The main idea now is to create an atomic model that behaves like an agent to integrate it in the discrete event simulation.

The main characteristic of an agent is its ability to make its own decisions based on its own state and its environment. To model the human decision making in this work the Utility Theory is used: it assumes that the decision process has two elements: the options and the evaluation function, called utility function that maps each option in the choice set to a numerical value. The function $u: \mathcal{X} \mapsto \mathbb{R}$ is a utility function if $\mathcal{X}$ is the set of choices. Preferences of the modelled individuals can be: no preferences ($\backsim$), prefer the first over the second option ($\succ$) or the second over the first ($\prec$). If the preferences observed in the individuals modeled correspond to the relations given by $u(\circ)$, $u(\circ)$ is called a valid utility function for the given decision problem.

As application of this theory the MoreSpace project done for the VUT is introduced: The simulation of the booking management for all courses held at the VUT and the student flow to determine the utilization and accessibility of lectures and the allocated space. Dealing with the simulation of the whole Campus in Vienna as well as the student behavior it needs the best of both worlds, DEVS and ABM, to cover all characteristics of this system. Entering and leaving rooms can be best modeled using a queuing system whereas students are best represented by agents to model their individual behavior regarding the attendance of lectures as well as the travelling between lecture halls. The main idea is to use an ED atom to create an agent that interacts with and moves through the discrete event simulation, regarding it as its environment. .

2 The MoreSpace Model

The MoreSpace model actually simulates two different but interlaced things: the booking/assignment of rooms for lectures and the dynamic behavior of students attending these lectures and the resulting utilization of space.

The MoreSpace simulation model uses three different sources for the data basis of a simulation run: the information about courses, all information regarding the lecture halls and their status and information about the number of students attending a course are estimated based on historical data as well as actual trends.

The MoreSpace simulation uses this data for calculating a suggestion for the assignment of rooms according to the selected booking management rules. This leads to successful and not successful booking attempts, both which are part of the simulation result. Other results are i.e. the expected utilization of rooms.

This suggestion is used as a basis for the dynamic simulation of the semester that includes the movement of students through the system. Students are simulated as single agents with their own 'intelligence'; they are able to make decisions i.e. which lecture to attend if two lectures overlap.

During the course of the semester they attend their lectures and move through the university campus, causing the dynamic behavior of the system. This delivers additional results like the capacity utilization of the rooms – the number of students attending the lectures held in a room opposed to the capacity of the room. This gives a greater understanding of the real demand on room compared to the available space. Also a result of the dynamic simulation based on the student - agents is the accessibility of lectures – spatial as well as temporal

and the utilization of rooms. Accessibility is of great interest for it takes the size and layout of the VUT Campus into consideration. This is one of the major advantages of MoreSpace: it does not stop at the best possible assignment of rooms but also considers the individual problems that may occur for students to be able to attend a lecture.

The benefit of using a simulation tool in general is that ways of enhancement can be tested without a risk to the real system. The same can be said about MoreSpace: strategies for booking can be experimented with, in scenarios rooms can be closed, added, joined or redecorated with no risk or costs. Results can be achieved very quickly; the comparison of different scenarios can be easily done.

The model is designed to allow three different angles of experimenting:

- The room structure containing the type of rooms and their capacity and location,
- The list of courses held with their time and expected number of students and the required setup of the room
- The booking management used for the assignment of rooms to the single lectures of each course

A main task of creating a simulation of a real system is simplifying the system as it is observed in the real world as much as to reach a model that is only as complex as it is needed to achieve the desired results. In the case of MoreSpace the main focus was on the different rooms available for lectures and the long list of events that had to be managed. Only after working with the input data and analyzing it the realization dawned that one aspect had not been taken into consideration so far: the students. Just as important as it is to find the best equipped room with high enough capacity for every event, is it to ensure that students are able to attend this perfectly planned event. The collision of lectures is a very common problem that forces many a student to decide with lecture he will not attend. This may be due to overlapping times of lectures but even more often it is the time it takes to get from one lecture room to the next. This movement time is often underestimated and causes lectures to interfere with each other although they are not really taking place at the same time. But enabling students to attend their lectures as unopposed as possible is one important factor for keeping the quality of education on the VUT at its best.

So the focus in the model was shifted to also include the students. In a classic DEVS system the students would have been regarded as entities that are routed through the system, in this case through the rooms and the lectures occurring there. In the case of two overlapping lectures the entity would have always followed its designated path: attend the first lecture until it is finished and then proceed to the next. But students are no mindless entities; they have their own priorities and preferences and usually every student is a unique and to some extent unpredictable person. The classical DEVS approach seemed not satisfying and considering a group of people with certain behavior quickly leads to the most common approach for problems of that kind: Agent Based Modelling.

As already described before agent based modeling works with individual ‘agents’ that have a predefined behavior that results from an evaluation of their current situation and a set of rules.

According to this students can be considered as agents, their current situation results from the list of lectures they want to attend and the decision of where they will go is derived from the given set of rules. If certain possibilities are entered for certain decisions the individuality of the single persons simulated can be achieved.

The MoreSpace Model is designed to simulate the behavior of about 20 000 students. The high number of agents to be simulated proved to be too much to remain within reasonable computing time of a simulation run. Therefore the simulation of the exact movements of each agent based on the layout of rooms and corridors was not implemented in ED but in an external simulation model developed in JAVA. The ED model is able to use average travelling times or to interact with the JAVA model by sending the student agents to the external model of corridors and have them re-entering the system as soon as they have reached their destination. The agent based approach is the key for the successful interaction of both models as the agent simply crosses the borders to the other system and returns to the ED model at a later time.

One thing the CA model was able to show was the dependency of the time needed to reach a certain lecture room from the ‘student traffic’. It seems logical that the time will increase if a lot of people move through the same corridors, obstructing each other. The extent of the delay due to a high people density is shown in **Figure 1**. This result could not have been calculated in ED model.

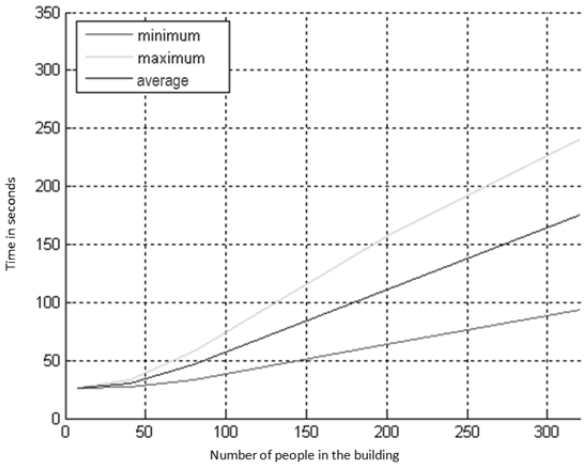


Figure 1. Walking time from room HS1 to room HS2.

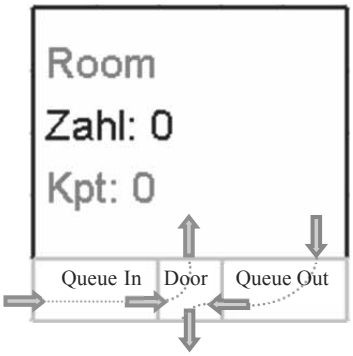


Figure 2. Structure of Room in ED.

To model the act of entering and leaving the room a server – queue combination is used consisting of two queues – one for the people leaving the room, another for those who want to enter – and a multi-server that represents the doorway as shown in **Figure 2**. A multi-server in Enterprise Dynamics is a server with capacity greater than one. A regular server in ED is also able to process several entities at once but only in a so called batch-run. This means all entities are processed simultaneously. The multi-server is able to handle several entities even with deferred entry times. The capacity of the multi-server representing the door corresponds to the size of the doorway. A small door where only one person at a time can pass through will have capacity 1; larger doors an accordingly higher capacity.

Entering and leaving has no fixed priority. Both queues are connected to the Server. A student entering the Queue_In will pass through the Door atom and enter the Room. A student leaving the room will enter the Queue_Out, pass through the Door atom and leave for their next destination.

Each room is represented as a single object within the simulation with its own attributes capacity, category and divisibility as well as its list of events with their date and time.

3 The Student Agent Atom

According to the aforementioned definition an agent goes through a sequence of states s_j^k following certain actions α_j^k; from the DEVS point of view an action α_j^k of an agent is a time consuming activity; therefore it is possible to replace it with an event triggered at the beginning of this activity using δ_{ext} the external transition function; the time consumed by the activity needs to be represented by ta the time advance function. So basically every action α_j^kneeds to be replaced by an event $x \in X$ and $\delta_{ext}(x)$ to ensure the correct update of the state variables of the atom itself as well as its environment. The state transformer function τ that updates the environment of the agent needs to be ingrained in the interaction of the agent with the DEVS based system.

The main problem is to coordinate the behavior of the DEVS system with the AB elements. In the case of the project MoreSpace the solution to this problem lays partly in the definition of the problem itself and partly in the simulation package used. Enterprise Dynamics is a discrete event simulation environment that represents the DEVS formalism in its basic makeup. All elements, called atoms, share the same layout and basic functionality. Several atoms can be used to build a more complex atom; several of them can be combined again, reflecting the principle of coupled models.

A given set of events may cause an update of the atoms state, the transition functions can be defined using the ED internal programming language. The time advance function ta is handled using an eventlist that controls the adjustment of time steps.

But additional to everything that forms an atomic or coupled model, ED offers spatial attributes as well as the functionality to move objects through space. And the set X of incoming values to trigger an event includes two elements that assist the ABM as well: the timed event x_t and the message event x_m . Both are not caused by a change of state of a model but in the first case by reaching a certain point in time and in the second by a message from outside of the system. Especially the latter option makes it possible to control elements in the DEVS system from external.

In case of the MoreSpace project this option was used to transfer the agents out of the ED model into a JAVA model for the calculation of travelling times at the movement between two spatial locations.

The agents represent students that attend lectures at a given time; therefore it is possible to use timer events for controlling the activity of the agent atoms. Triggered by a timed event the activity of the agent takes place: they either change their location by moving towards a certain lecture hall, take a decision whether they should attend a lecture and chose one in case of several overlapping, they enter a queue, or leave the ED model for the more exact simulation of peoples movement in a JAVA model.

But it is the agents themselves that control the setting of these events: each atomic agent has its own personal settings and takes its own decisions. The student is considered an atomic model with the behavior of an agent. It is a member of both worlds, DEVS as well as ABM because it still interacts with the DEVS environment as a regular input and output of the DEVS system represented by Queues and Server.

Considering the definition of a run r_j^k an action $\propto_j^k$ takes place in response for every state $s_j^k$the student takes in. $S^k$is the set of all state variables an agent a^k can see. The actions as well as the change to the state variables have to be defined by using the structure of the atomic model.

For the interaction with the discrete event simulation system the agent has to act as an input to the DEVS elements. I.e. in ED an external event of a Queue is triggered by an incoming atom.

The student agent entering the queue atom is treated as the positive input value $x \in X$ and therefore triggers the time advance function and the external translation function, thus the autonomous behavior of the atomic model 'Queue'. The 'Queue' treats the agent atom just like any other input, resulting in the output value $\lambda(s)$ that causes the student atom to be moved into the next atom, the 'Door'. Here again the agent atom is treated as any regular input, generating the output ue $\lambda(s)$ after $ta(x)$ has passed from the 'Door', that causes the agent atom to be positioned in the 'Lecture Room'. Here the control is returned to the agent atom: the lecture room has no further functionality but to count the number of student atoms it contains, it does not generate an output value that influences the agent atom.

The next action of the agent atom is completely independent from the activity of the discrete event system elements: the student will leave again at the time either the lecture ends or it has another activity planned that is of higher interest than its current activity. Leaving is the identical procedure: the student agent enters the queue leaving the room, passes through the door and then moves on to its next activity.

The main characteristic of students is their ability to make their own decisions based on their own state and their environment. To model the human decision making in this work the Utility Theory is used: it assumes that the decision process has two elements: the options and the evaluation function, called utility function that maps each option in the choice set to a numerical value.

In case of the MoreSpace model one decision the student has to take is the case of overlapping lectures. If a lecture x and a lecture y take place at the same time the student has to decide which one to attend.

Basically the utility function for the decision making of the student for overlapping lectures can be derived from the type of lectures the student has to choose between. Of course the individual preferences need to be added as well: the option not to go anywhere can be added to model the possibility that a student might not attend any lecture at all. The decision between lectures may be done stochastically.

But a tendency towards one lecture has to be remembered – the decision the next time may depend on the decisions felled in the past. The decision also depends on the quality of the lecture itself: Are there enough seats? Does the course have a good scriptum? Is it an interesting topic? The depth of the utility function can be easily altered.

4 Student Numbers

A general problem in formulating the model was the lack of distinct information regarding the number of students attending each lecture. Due to the lack of mandatory attendance at most lectures the numbers differ quite strongly from the number of passed exams or even the number of enrolled students.

It is a common occurrence that the numbers of students that attend a lecture tend to drop after the initial weeks of a new semester. This effect is especially distinctive observable in lectures without mandatory attendance.

The real number of students following the course is mostly not matching the number of students really attending each lecture; the best approximation for this number is n_F, the number of students taking the exam for passing the course.

But the number n_{A_t} of students attending a lecture at time t depends on several factors:

- The perspicuity of the lecturer – the quality of the lecture influences the attendance of students considerably. A lecturer that is hard to understand or whose explications are hard to follow will have fewer students than one who presents his lectures in a comprehensible and interesting way.
- Of course the topic itself is always a factor as well
- The quality of the accompanying material: a lecture that offers high quality manuscripts that contain anything the student needs to pass the exam usually lead to a drop in attendance. The strongest factor for attending a lecture after it being mandatory is the need for lecture notes. If they are provided one main factor simply falls away.
- The importance of the lecture: compulsory lectures are usually highest in priority for they must be done. Some courses are precondition for getting a place in a tutorial
- Assessment of student's performance: if during the semester several interim tests are done for the assessment of the students this usually leads to a different behaviour: at the date of the tests most students attend the lecture to take part in the test. Right before and after these dates the number raises and drops respectively.

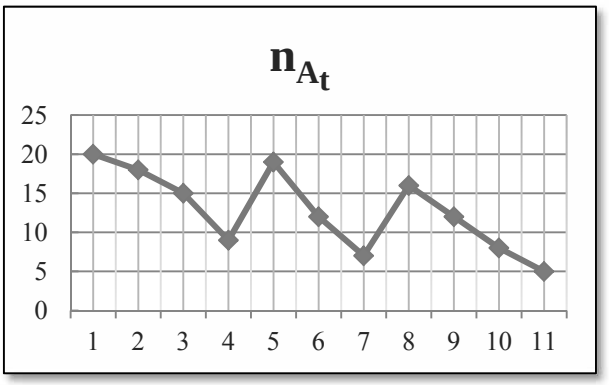


Figure 3. Course with Interim Tests.

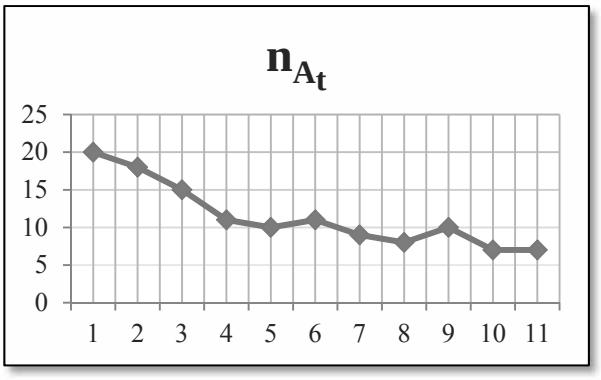


Figure 4. Course without Interim Tests.

Using the agent based approach did allow to use the known data as basis for the number of students. Students have their own behaviour in attending lectures that recreates the attendance behaviour of the real system. Known are:

n_E … Number of enrolled students
n_F … Number of students finishing the course by passing the exam

Not known are

n_{A_t} … Number of attending students at time t
n_{D_t} … Number of drop out students at time t
n_{N_t} … Number of not attending students at time t

We can say for sure:

$$n_E \geq n_{A_t} \geq n_F$$

Considering the current situation on the TU Vienna one may even be sure to say:

$$n_E > n_{A_t} \geq n_F$$
$$n_{A_t} = n_F * F_S$$
$$n_F = n_{A_t} + n_{N_t} - n_{D_t}$$

For courses with a mandatory attendance one might say:

$$n_E > n_{A_t} = n_F + (n_{D_{t_E}} - n_{D_t})$$

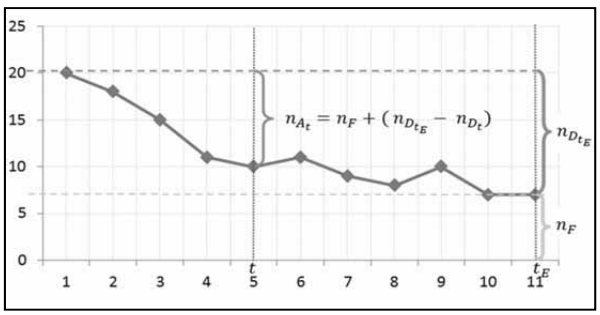


Figure 5. Number of Attending Students.

Using the agent based approach did allow to use the known data as basis for the number of students. Students have their own behavior in attending lectures that recreates the attendance behavior of the real system.

The dynamic simulation involves the student going through the semester and attending their courses. They allocate space in the assigned room and add to the capacity utilization of the room.

The system state changes during the simulation due to events caused by the elements of the simulation model. There are two additional events that influence the system state but are not triggered by the occurrences within the system:

- Assignment of rooms to lectures
- Blocking of rooms during certain time period due to reservation or reconstruction

The booking of rooms for lectures will generally take place before the actual semester begins, but there will always be a demand of space during the semester as well, resulting in additional room assignment as well as an adjustment in the student behavior. Additional lectures or other events will cause student to attend them and therefore make them take new or different decisions than they would have before this change happened.

The unavailability of a room for a certain time period due to reconstruction or other reasons will lead to a reassignment of all lectures booked in this room. This will also affect the students as they have to be informed about the change in location.

The student atom contains the behaviour of the student as well as their list of lectures they have to attend. This list is generated at the time of creation of the student and results from their semester as well as field of study. The generation is based on probability functions to achieve a wide variety of lecture combinations to represent the inhomogeneous groups of students that occur especially in courses in higher semesters.

The behaviour of the students is based on their goals – attending their lectures – as well as on the overall situation. In case of collision of events they wish to attend students have to make a decision.

4.1 Event Matrix

Each student atom owns a matrix of lectures it will attend. The size of the matrix *n* x 4 is given by the number of lectures *n* the student is assigned to. This matrix L_s contains the basic information for the behaviour of the student s during the simulation:

$$L_s = \begin{pmatrix} l_{1,1} & \dots & l_{1,4} \\ \vdots & \vdots & \vdots \\ l_{i,1} & \dots & l_{i,4} \end{pmatrix}$$

- The element $l_{i,1}$ contains the begin time of the lecture i.
- The element $l_{i,2}$ contains the end time of the lecture i.
- The element $l_{i,3}$ contains the ID of the course attended at lecture i.
- The element $l_{i,4}$ contains the pointer to the room of the lecture i.

4.2 Attending a Lecture

Here the event for the attending of a lecture is set. The time the student will need to reach the location of the lecture hall has to be taken into consideration. We define:

- T_c current time
- T_i start time of the lecture
- $\hat{T}_i$ end time of the lecture
- $\tilde{T}_i = T_e + t_w$ start time of the lecture considering walking time
- T_e time for the student to start walking
- t_w Time it takes the student to reach the lecture hall from their current position
- t_e time until T_n

If $i \leq n$ we can assume that:

$$\exists\, T_i = L_s(i,1) = l_{i,1}$$
$$\exists\, C_i = L_s(i,3) = \mathrm{l}_{\mathrm{i},3}$$
$$\exists\, R_i = L_s(i,4) = \mathrm{l}_{\mathrm{i},4}$$

where

$$\mathrm{l}_{\mathrm{i},1} \geq T_c$$
$$C_i \neq C_c$$

Hence:

$$t_e = \max(T_e - T_c\ ,0)$$

If i > n the students have finished their list of lectures and can leave the simulation.

4.3 Queuing

During this activity the student moves into the queue in front of the lecture hall. The according event for leaving a room is set by evaluating the next lecture the student will visit:

$$T_i = L_s(\mathrm{i},1) = \mathrm{l}_{\mathrm{i},1}$$
$$\hat{T}_i = L_s(\mathrm{i},2) = \mathrm{l}_{\mathrm{i},2}$$

under the following conditions:

$$T_i \geq T_c$$
$$C_i \neq C_c$$
$$T_{i+1} < \hat{T}_i \Rightarrow Conflict \text{ !!!}$$

According to the given probability $\mathbb{P}$ a student will chose one of the two possibilities for the time T_e when they will leave the current lecture:

$$T_e = \begin{cases} \hat{T}_i \\ \max(T_i - t_w, 0 \end{cases}$$

and

$$t_e = \max(T_e - T_c, 0)$$

5 Results of MoreSpace Simulation

5.1 Utilization of Lecture Rooms

This data shows the number of hours each lecture room was booked by the booking procedure. This shows the theoretical utilization, the time the room is booked, but not the time the room is truly used. As past experiences have shown sometimes rooms may be booked for a lecture that does not take place.

5.2 Capacity Utilization of Lecture Rooms

This data shows how many students did attend a lecture in the simulation. The number expected is given; the according number of students is assigned. If the number of attending is lower than that it hints at a problem at the accessibility of the course.

5.3 Not Booked Events

The booking procedure tries to find a lecture room for every lecture planned. If it is not able to assign a room the according lecture is listed in this data. For the comparison of several simulation runs one has to make a distinct decision on which aspect the attention is focused. Depending on this the key data has to be selected.

The following example shown in **Figure 6** illustrates how easily data can be misinterpreted in the comparison of two scenarios:

Scenario 1: 1 lecture from 11.00 to 15.30 for 56 students could not be booked in any lecture room.

Scenario 2: 2 lectures from 10.00 to 11.30 for 23 students and from 15.00 to 16.00 for 41 students could not be booked in any lecture room.

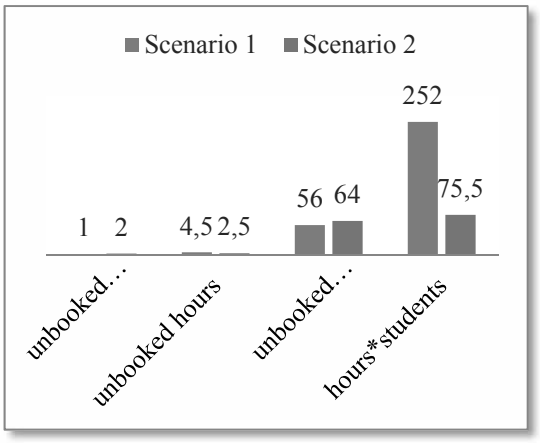


Figure 6. Key Data for Not Successful Booking.

This demonstrates the importance of defining the correct key data; Depending on which value is considered the assessment of the simulation results can be interpreted completely different.

Considering the number of not booked lectures Scenario 1 seems to deliver the better result. The number of not booked hours quickly shows another picture: where in scenario 2 both lectures together result in 2.5 hours that could not find a room, Scenario 1's 1 lecture requires 4.4 hours of time.

The picture again changes if one looks at the number of students that cannot attend a lecture without a room: Scenario 1 is the better one in this regard. But taking the hours of lecture each student misses into account Scenario 1 suddenly looses highly against Scenario 2 again.

5.4 Accessibility of Lectures

The accessibility of lectures can be interpreted in two different meanings:

- Temporal accessibility: this indicates if a lecture overlaps with another lecture.
- The spatial accessibility indicates if a lecture can be reached in time: this considers lectures that take place after each other, even with a time gap between them but the location of the rooms is such, that it is not possible to reach the second lecture on time.

While the first kind can be easily determined by evaluating the given data, the second is much more difficult to estimate: the real time it takes from one lecture hall to another depends on far more than the spatial distance: The density of people moving through the corridors, the waiting time at elevators, the distance to staircases influences the walking time.

This makes the evaluation of the spatial accessibility to one of the simulation results as it is able to deliver far more accurate results than estimation by distance.

6 Conclusion

Using an agent based approach for simulating the students in the <morespace> project proved to be a good approach to cover several demands:

- The possibility to model the exact movement of students during the corridors and across the TU Campus, even covering the eventuality of travelling between buildings that are further apart or emergency evacuation simulation.
- Students have to be regarded as entities with individual preferences and decisions based on their previous behaviour and state.
- The need to hand control over their actions to the students themselves. They are not routed through the system via a course of server and queues but move on their own.

The implementation in ED did result in a hybrid agent atom: it has the basic attributes that describe an agent:

- Autonomy: each agent acts on its own and decides its own behavior
- Social ability: agents are able to communicate with each other
- Reactivity: agents react to their environment and changes therein
- Pro-activeness: Agents do not only react to their environment but act on their own as well

But due to the ED configuration the agent atom still holds the basic functionality that relates to the DEVS concept. This enables the interaction with the ED model at certain points according to the general input/output procedure. This is the case every time a student enters the queue in front of a door leading to a lecture hall: the basic configuration of any atom is also present in the agent, thus making it able to react to the event of entering and exiting another atom. This is used to update the state of the agent

For the specific project MoreSpace this solution was the best combination as the atomic agent belongs to both worlds, making it some kind of double agent that is able to use the resources of DEVS as well as ABM.

The agent based approach provides a very detailed insight into the movements and activities of the single students and therefore allows an assessemesnt of effect of certain scenarios on the quality of teaching as well as on the averall state of the room situation.

Acknowledgements

Development of MoreSpace has been granted by ZIT Technologieagentur der Stadt Wien GmbH, within a FemPower Framework (www.zit.co.at). Implementation of Morespace has been granted by Vienna University of Technolgy. More information at www.morespace.at.

References

[1] B. P. Zeigler. *Theory of Modeling and Simulation*. John Wiley, New York, 1976.

[2] M. Wooldridge,. *An Introduction to Multi Agent Systems*. John Wiley and sons, Chichester, England 2002.

[3] B.S.S. Onggo. "Running Agent-Based Models on a Discrete-Event Simulator", in *Proceedings of the 24th European Simulation and Modelling Conference*, 25-27 October 2010, Hasselt, Belgium (ESM10). Ostend, Belgium: Eurosis-ETI, 2010.

[4] S. Abdennadher, M. Marte. *University course timetabling using constraint handling rules*. Applied Artificial Intelligence: An International Journal Volume 14 Issue 4, pp 311-325, 2000,

[5] C. Beyrouthy, E. K. Burke , B. McCollum, McMullan, D. Landa-Silva, A. Parkes. *Towards improving the utilisation of university teaching space*. Technical Report NOTTCS-TR-2006-5, School of Computer Science & IT, University of Nottingham, 2006.

[6] D. Wiegand, P. Mebes, V. Pichler. Real Estate und Facility Management– neue Anwendungsbereiche für diskrete ereignisgesteuerte Simulationen. *Procedings zur ASIM 2006, 19. Symposium Simulationstechnik*, pp. 15-20, 2.–14. September 2006, Hannover, Germany.

Submitted: September, 2010
Revised: March, 2011
Accepted: March20, 2011

Introducing Software Components to Road Traffic Modeling and Simulation

Petr Zelenka

University of West Bohemia, Dept. of Computer Science and Engineering, Univerzitni 8, 30614 Plzen, Czech Republic;
pzeli@kiv.zcu.cz

Abstract. This article presents an application of contemporary software engineering practices, in particular software comp onents, into the area of road traffic modeling and simulation. It shows that by building road traffic models from individual components, both the topology of the simulated traffic system and thebehavior of the simulation model can be easily changed according to the requirements. In fact, it is possible to develop a single road traffic model, the properties of which can be adjusted almost arbitrarily. This flexibility, resulting in a multi-adjustable simulation, is of a great advantage as opposed to the existing road traffic modeling and simulation tools that usually implement only one behavioral model and are thus predetermined for simulating a limited set of problems

Introduction

Modeling and simulation of road traffic has gained in popularity in recent years. This is not surprising, because it constitutes the only tool available so far for predicting behavior of a traffic system. Using traffic models, it is possible, for example, to evaluate the impact of the traffic restrictions related to a road work (whether construction or maintenance) and to determine which combinations of road works can be performed simultaneously without causing unnecessary congestion, or to compare the performance of several different road design alternatives. In connection with the recent onset of intelligent traffic management and information systems, traffic models can also be used to test these systems during their development in a real-like environment.

Basically, there are two different approaches to road traffic modeling (see [2], p. 25 ff.). The first one, based on physical theories of fluid dynamics, describes the traffic by differential (or, in the case of computerized models, difference) equations, using physical quantities such as traffic flow and traffic density. Such models are called macroscopic, because they deal with the traffic as a whole.

They can be used to evaluate the performance of a traffic system under different conditions, but they are somewhat limited in their capabilities (for example, it is not possible to measure travel times between a given pair of points, or, it is difficult to estimate the effect of making a traffic lane restricted). Moreover, they often suffer from statistically significant inaccuracies once the traffic system gets congested. The second approach makes use of a greater level of detail. It describes the behavior of the individual traffic participants (cars, trucks, streetcars, and so on) by a combination of difference equations and decision trees, using quantities such as acceleration and speed.

The quantities describing the traffic as a whole then need to be extracted using statistical measurements. On the other hand, any statistically measurable quantity can be obtained, including distance traveled by a traffic participant or the above-mentioned travel time between a pair of points. Such models are called microscopic and their capabilities are almost unlimited. However, their computational complexity (both in terms of time and space) can be enormous.

1 Software Components

The concept of software components became popular in 1990s, especially in connection with increasing demand for reusability. Object-oriented programming has some mechanisms for supporting reusability (namely inheritance and method overloading), but the level of reusability that can be achieved this way is still far from the level that is usual in other engineering disciplines. Machinery can serve as a great example. Suppose that you need to fasten some things together using a bolt and a nut. Further suppose that you have some bolts left in stock, but no nuts. Because bolt and nuts are standardized (they can be uniquely described by a set of their dimensions), you need to buy only nuts that correspond to the bolts you already have.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 21-26 | doi: 10.11128/sne.21.tn.10041

Moreover, they even do not have to originate from the same manufacturer.

Although machinery and software engineering are indeed not very similar, there is no good reason for software to be different in this respect. Machines are being constructed from components (that is, self-contained operational parts), which in turn can be constructed from subcomponents, and so on. These components are then assembled together in a way that allows for the intended function of the machine. Software is developed in a similar way.

The overall intended functionality is decomposed into smaller pieces (top-down approach), which are then designed and implemented (bottom-up approach). In object object-oriented programming, these pieces take the form of objects, whereas in procedural programming they take the form of modules. Either case, they communicate together using method (or function) calls, so that the overall functionality is achieved. The only important difference between the components on one side and either objects or modules on the other side is thus the formal description of their interfaces (the points where they meet each other).

There are many definitions what a software component is and what it is not. One of the most generally accepted one is that by Szyperski (see [7], p. 548):

> '*A component is a unit of composition with contractually specified interfaces and explicit context dependencies only. [...] A component can be deployed independently and is subject to composition bythird parties.'*

Because both objects and modules are units of composition and can be deployed independently (in the form of shared libraries, for instance), the fundamental part of this definition are the *"contractually specified interfaces"*. It means that a software component needs to explicitly define any functionality it provides as well as any functionality it requires from other components. For this purpose, a reasonable choice is Meyer's Design by Contract (DbC) principle (see [4]). Using DbC, the semantics of an operation (represented by a method or function) can be described by a union of preconditions, postconditions, and invariants. Preconditions describe the state expected before the operation can be used (that is, what the caller of the method or function must provide and what the provider of the implementation may rely on).

Conversely, postconditions describe the state expected after the operation was used (that is, what the implementer must provide and what the caller of the method may rely on if he has fulfilled the corresponding preconditions). Finally, invariants describe the state that will not be affected by the operation itself and are therefore used as consistency constraints. These explicit statements represent a contract between the provider and the user of the component and allow for the above-mentioned third-party composition (where the first party set up the contract and the second party provided the implementation).

Contractually specified interfaces also have an interesting implication. Any two components implementing the same contract can be substituted one for the other (see [7], p. 83 ff.). In other words, replacing one of them with the other does not require any changes in the rest of the software. In fact, unless the components representing the rest of the software are explicitly told, they are not able to learn about such a change. Components thus make the functionality of the software much more flexible and adaptable to changes.

Implementing the same contract does not mean that the components must perform the functionality in exactly the same way. For example, consider a component dealing with logging. Such a component is expected to make records of important events that have occurred, but the resulting log may differ in the recorded level of detail (critical errors only or a full spectrum of events including debugging information) or even in its form (text file, binary file, or a set of records in a database). Another great example of a set of operations implemented in a possibly different ways are database connectors, which provide a uniform interface for performing database operations in order to allow for easy change of the underlying database engine.

2 Components in Road Traffic Models

In theory, both macroscopic and microscopic road traffic models are able to make use of the advantages of the component approach. Nevertheless, using software components for macroscopic modeling is much more reasonable in connection with hybrid models (see Section 3.3) than for standalone macroscopic models.

Every traffic system can be decomposed into two different aspects: its topology (the layout of road segments and intersections) and its behavior (the way in which the state of the system is changed).

These two aspect can be modeled each by a separate set of components.

2.1 Road Topology Change Using Substitution of Components

Road topology, constituted by so-called structural components, is more or less static, at least in the sense that it will usually not change during a simulation run. However, it is important that the topology can be modified easily in order to test and compare different road design alternatives. So, an appropriate decomposition of the road topology is needed.

Graph theory can help a lot in identifying structural components. Consider the following graph representation: each intersection (that is, a place where three or more road segments meet) corresponds to a vertex and each road segment corresponds to an edge. Then each vertex and each edge shall be modeled by a separate structural component (see Figure 1). In such a topology, components can be easily substituted. For example, a crossroad can be replaced by a roundabout (see Figure 22). More generally, in the terms of graph theory, any connected subgraph can be replaced by another connected subgraph.

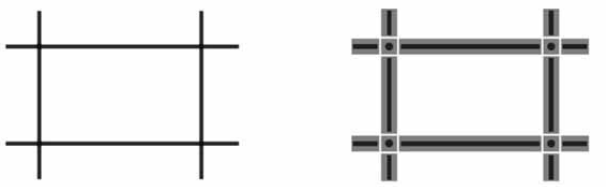


Figure 1. An example of using graph theory for identification of structural components. Here, the original road topology consisting of four crossroads (left) and its decomposition with structural components highlighted in grey (right).

2.2 Model Behavior Change Using Substitution of Components

Microscopic road traffic simulation models consist of several submodels, where each submodel handles a specific task in the simulation (see [5], p. 5). The most essential submodels are the car-following model and the lanechanging model. The car-following model describes movements of a vehicle in response to the behavior of the preceding vehicle traveling in the same lane, whereas the lane-changing model describes movements of a vehicle in connection with a lane change (which may be necessary for turning at the next intersection, for instance).

There are many different car-following models (Gazis-Herman-Rothery model, Gipps model, or Fritzsche model, to name a few), each taking a slightly different set of aspects into account (Fritzsche model considers driver's reaction time, while most other models do not, for instance; for more details, see [5] or [8]). These differences may seem to be small, however, depending on the problem being simulated, they can affect the results in a statistically significant way. The same applies to the lane-changing models, as well as to all other models not yet mentioned, such as overtaking models or yielding models.

Existing microscopic road traffic modeling and simulation tools (for example, Aimsun, Corsim, Mitsim, Paramics, or Vissim) usually select one from these models and stick to it in all cases. This limits their range of application, because, as said above, not all models are suitable for simulating a particular problem. Extracting each such model to a separate component (this time called behavioral component) allows to make use of the component substitutability described in Section 1. We can then select a set of models that are expected to provide the best results for a particular purpose.

Further, we can use several different models at the same time. This can be particularly useful when different classes of traffic participants shall behave differently. For example, streetcars cannot travel off the track, so they do not need a lane-changing model (track junctions can be simulated much more easily). Or, in most countries, streetcars have right of way in all situations, so the other traffic participants are required to yield. However, Germany is a notable exception. German traffic law considers a streetcar to be an ordinary traffic participant in this respect. As a consequence, yielding models for streetcars and the rest of traffic participants will be different, except where a German traffic system is simulated. Such changes in behavior are difficult to achieve in most of the existing road traffic models.

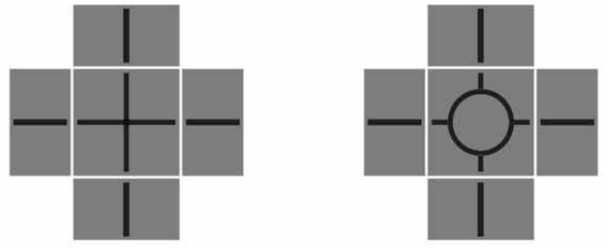


Figure 2. The topology of a traffic system can be changed easily using substitution of structural components. Here, a crossroad (left) has been replaced by a roundabout (right).

It is also obvious that although the traffic participants belonging to the same class share the same behavior, they can differ in some parameters. Unfortunately, these parameters may affect the behavior considerably. For example, mass and braking efficiency affect the braking distance. As a result, traffic participants may need to start braking at a different time if they need to stop at a given point. This means that the behavioral components need to be parametrized and each traffic participant needs its own instance with parameters set to the corresponding values.

However, there are many traffic participants in a typical microscopic road traffic simulation and creating an instance of all the necessary behavioral components for each of them would cause an extremely high consumption of memory. In such situations, the flyweight design pattern (see [1]) comes in handy. The idea of flyweight consists in extracting the shared part of two or more entities into a new entity and referencing this new entity from the original ones. In our case, a traffic participant would contain only a set of its own parameter values and a reference to the corresponding behavioral model (see Figure 3).

2.3 Multiple-Level-of-Details Modeling Using Component Adapters

So far, we have considered component substitution only when both the substituted and the substituting components implemented the same contract. In fact, any two components can be substituted one for the other provided that there is an adapter able to overcome the differences between their contracts. An adapter in this sense is actually an intermediate component (sometimes also called a connector) that implements both contracts and provides a logic to translate from one to the other. In other words, an adapter serves as an interpreter between components that could not otherwise communicate.

In a road traffic simulation, the impact of a traffic restriction (or any other traffic control measure) typically diminishes with increasing distance from its point of action. So, if a point of action is located somewhere in the heart of downtown, there is little interest in results from a distant suburbs. However, even these areas still need to be simulated, because they are an integral part of the traffic system and can therefore influence the areas of interest over time. But they indeed can be simulated at a much lower level of detail, for instance using a macroscopic model. Doing so can substantially reduce the computational intensity of the overall simulation.

Models capable of switching between the microscopic and the macroscopic level of details are also called hybrid. An introduction to the problematics of hybrid traffic modeling can be found in [3].

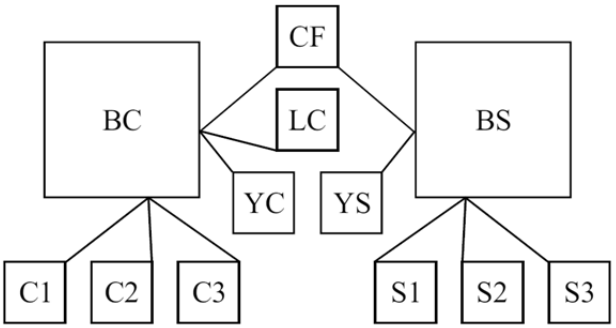


Figure 3: An example of applying the flyweight design pattern to behavioral components. Here, three cars (C1,C2,C3) and three streetcars (S1,S2,S3), each referencing the corresponding behavioral model (BC, BS). Both models share the car-following model (CF) and each has its own yielding model (YC, YS). In addition, the behavioral model for cars has a lane-changing model (LC), whereas the behavioral model for streetcars has none.

Implementing the adapters between microscopic and macroscopic models is actually not difficult at all. The only important difference is that macroscopic models work in terms of physical quantities, whereas microscopic models work in terms of individual traffic participants. As a result, adapters from microscopic to macroscopic models need to estimate the values of the respective physical quantities (traffic flow and traffic density).

Traffic flow is defined as the number of vehicles per a unit of time, so the corresponding value can be obtained using a counter and a timer. The counter is increased by every traffic participant passing through the adapter. The timer then periodically calculates the value of the traffic flow and to resets the counter to zero. Traffic density is defined as the number of vehicles per a unit of length, which is a little bit more problematic to measure.

The reason is that adapters do not represent a road segment or any other piece of road topology and are thus dimensionless. Fortunately, traffic density can be calculated from the traffic flow and the mean speed, which is measurable much more easily. It shall be obvious that during this conversion, a lot of information, such as the classes of traffic participants or their parameters, is lost.

However, it does not constitute a problem, because this conversion is only done in the direction to the areas out of interest and the lost information will not be needed anymore (see Figure 4). Conversely, adapters from macroscopic to microscopic models need to generate the individual traffic participants according to the values of traffic flow and traffic density. This can be done using a pseudorandom generator parametrized by these values.

The additional information about the traffic participants that either has been lost during the reverse conversion or has never existed needs to be generated at this moment. As a consequence, these adapters need additional sources of information, such as the probability of each class of traffic participants in the system. For an example of a boundary between macroscopic and microscopic models, see Figure 5.

At first glance, it may seem that the process of adapting between microscopic and macroscopic models can affect significantly the results of the simulation. But this is not necessarily true. In an effort to reduce the amount of information transferred between microscopic road traffic models in a distributed computing environment, Potužak (see [6], p. 3 ff.) has studied the process of extracting the macroscopic quantities and using them for generating the traffic participants again. He showed that the statistical deviation caused by this process can be kept under 5 %.

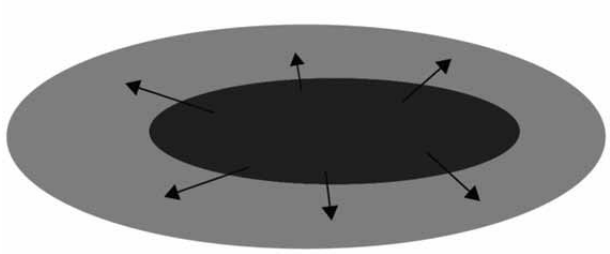


Figure 4. An example of a hybrid simulation model. The area of interest (dark grey) is simulated using a microscopic simulation model, whereas the area out of interest (light grey) is simulated using a macroscopic simulation model. The arrows show the direction in which the information is lost in microscopic-to-macroscopic adapters.

2.4 Transparent Distributability Using Component Adapters

Even if the areas out of interest are simulated using a macroscopic model, the overall computational intensity may still be too high for a single computer. In such situations, a distributed computing environment may be necessary to get the simulation results in a reasonable time. But this also means that the simulation model needs to be adapted for a distributed environment. In particular, the model must be divided into parts (in this case, disjoint subsets of structural components), it must be decided which part will be assigned to which node, and remote connections between the parts must be established.

This is another situation where software components can help. The remote connections can be made transparent to the components using remoting adapters, that is, connectors that encapsulate the additional logic needed for remote communication. In remote procedure call technologies, they are typically called stubs and skeletons. Because adapters can be placed between any two components in general, there are many places where the simulation model can be divided into parts (see Section 2.2). This fact could be utilized to implement some load-balancing mechanism.

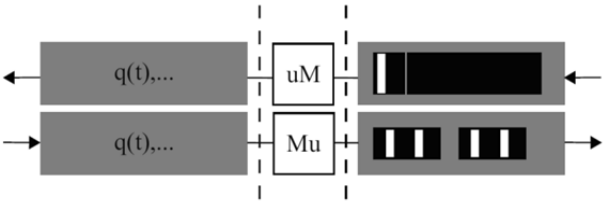


Figure 5. An example of a boundary between macroscopic and microscopic models. Here, the boundary lies within a road segment consisting of one lane in each direction. In the top half of the figure, the microscopic structural component (right) is connected to the macroscopic structural component (left) through an microscopic-to-macroscopic adapter (uM). Conversely, in the bottom half of the figure, the macroscopic structural component (left) is connected to the microscopic structural component (right) through an macroscopic-to-microscopic adapter (Mu).

3 Summary

In this paper, we dealt with the possibility of using software components in the area of road traffic modeling and simulation. We showed that building road traffic models from individual components can be of a great advantage and can lead to development of a single multi-adjustable simulation model. First, we introduced the concept of structural components and outlined how their substitution can be used to change the topology of the simulated traffic system. Then, we applied the same principle to the behavior of the model and showed that by introducing behavioral components, several different models can be used at the same time, each modeling a specific task.

Further, we outlined adapters between microscopic and macroscopic models and their use for simulating different parts of the traffic systems at a different level of detail. Finally, we showed that component approach in connection with remoting adapters provide for an easy adaptation of the model for a distributed environment, possibly with some load-balancing mechanism.

The experience gained from this research and the still ongoing research of structural components will be used to develop an experimental multi-adjustable road traffic model.

Acknowledgment

This work was supported by the Grant Agency of the Czech Republic (project number 201/08/0266).

References

[1] E. Gamma, et al. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA, USA, 1994.

[2] A. Hamalainen. *Studies of Traffic Situations Using Cellular Automata.* PhD dissertation, Helsinki University of Technology, Espoo, Finland, 2006.

[3] D. Hartman. *Switching Scalability of Hybrid Model for Complex Traffic Simulation.* PhD dissertation, University of West Bohemia, Plzen, Czech Republic, 2008.

[4] B. Meyer. *Design by Contract*. In Advances in Object-Oriented Software Engineering, Prentice Hall, Englewood Cliffs, NJ, USA, 1991.

[5] J. J. Olstam, A. Tapani. *Comparison of Car-Following Models.* Swedish National Road and Transport Research Institute, Linkoping, Sweden, 2004.

[6] T. Potužak. *Distributed Traffic Simulation and the Reduction of Inter-Process Communication Using Traffic Flow Characteristics Transfer*. In Proceedings of the Tenth International Conference on Computer Modelling and Simulation, Cambridge, UK, 2008.

[7] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. Second Edition. Addison-Wesley, Reading, MA, USA, 2002.

[8] Y. Zhang. *Scalability of Car-Following and Lane-Changing Models in Microscopic Traffic Simulation Systems.* Master's thesis, Louisiana State University, Baton Rouge, LA, USA 2004.

Submitted: March, 2010
Accepted: December 10, 2010

Model-based Safety Monitoring of Produkt Flow Paths

Gustavo Quiros*, Reiner Jorewitz, Ullrich Epple

RWTH Aachen University, Chair of Process Control Engineering, Turmstrase 46, 52064 Aachen, Germany;
g.quiros@plt.rwth-aachen.de

Abstract. Product flow paths are the routes that products take while flowing through a plant. The development of systems that ensure and monitor the correct and safe transport of material is usually plant-specific and based on informal knowledge, and is time-consuming and error-prone. In this work we seek a synthesis solution for the task of monitoring the safety of product flow paths in processing plants. A formal model of the plant is used, which defines a simplified plant representation that considers the possibility of flow through its elements. Based on this model, we present a formalisation of the safety of a product flow path at a given plant state. This formulation may be used as a guideline for automating the construction of systems which perform safety monitoring of product flow paths. An outline for the design of such a system following a decentralisation scheme is also presented.

Introduction

A basic and essential operation performed by processing plants is the movement of material, i.e. products, between plant elements. This movement or flow is physically constrained by the structure of the plant itself and is caused either by gravity or by the operation of active devices such as pumps. We denote the routes that products take while flowing through a plant as *product flow paths*. Understanding this concept is of great importance when developing process automation systems. For instance, the operation of plants with flexible structures consisting of multiple and alternate product flow paths requires adequate working states of plant elements like valves in order to restrict the flow of material to a desired path, as well as to ensure the safety of the flow operation by avoiding undesired and potentially hazardous situations such as leaks (when the product flow diverges from the intended path and reaches unexpected plant locations) and unintended mixtures (when another product enters an active flow path unexpectedly).

For example, Figure 1 shows a diagram of a simple filling station consisting of four tanks, two pumps, and two product input nozzles. Each of the four tanks may be filled by from any of the two product inputs by the corresponding pump. The product flow path shown in blue corresponds to one such filling operation. For this product flow path, the diverging path shown in red represents a potential leak to another tank. Likewise, the joining path shown in green represents a potential unintended mixture. These situations are avoided by closing valves V4 and V7 respectively, whenever this product flow path is in operation.

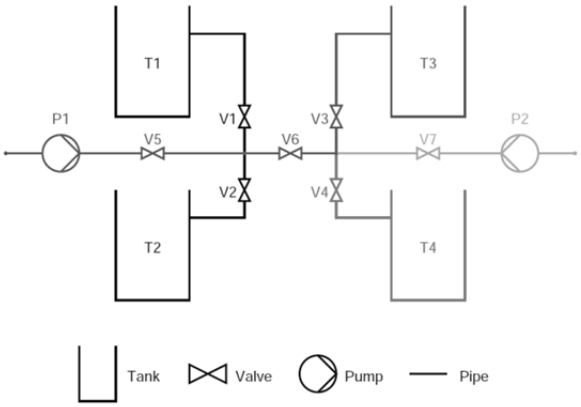


Figure 1. A tank filling station with flexible structure. Each of the four tanks may be filled by from any of the two product inputs by the corresponding pump. The product flow path shown in blue corresponds to one such filling operation. For this product flow path, the diverging path shown in red represents a potential leak to another tank. Likewise, the joining path shown in green represents a potential unintended mixture.

Process control systems usually fulfil the important task of ensuring and monitoring the correct and safe transport of material in processing plants, as they are designed and implemented with these requirements in mind.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 27-36 | doi: 10.11128/sne.21.tn.10043

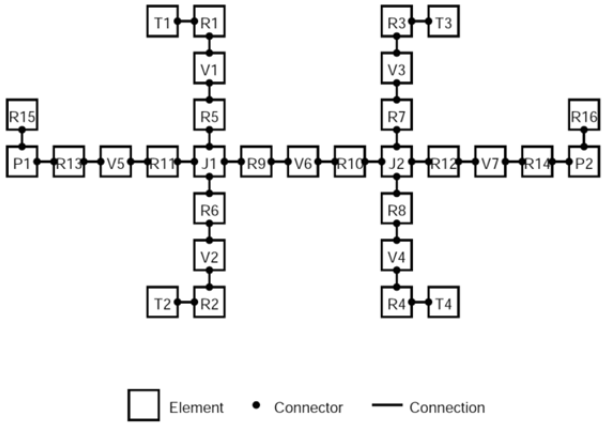


Figure 2. A graphical representation of the structural model of the filling station of Figure 1. The tanks (T#), pumps (P#), valves (V#), pipes (R#) and pipe joins (J#) of the plant are represented as elements uniformly, having an adequate set of connectors which are connected with those of other elements in a one-to-one fashion.

However, the development of these plant-specific solutions is based on informal knowledge and is time-consuming and errorprone, especially in the case of large and complex plants which follow flexible designs, e.g. multi-purpose plants in the pharmaceutical industry. Furthermore, this engineering phase must be repeated as soon as the plant structure itself changes. This serves as a motivation for studying automated approaches that may partly or completely replace this engineering work, ensuring its correctness at the same time.

The automation of automation [10, 11] offers a technique for reducing the engineering effort of the development of process control systems, as well as improving their quality, by automating the construction of such systems (in part or in whole) through the application of engineering rules. Analogously, in this work we seek a synthesis solution for the task of monitoring the safety of product flow paths.

In order to apply an automatic synthesis approach to systems which perform tasks regarding product flow paths, an adequate representation of the plant is required which describes its structure and the behaviour of its elements unambiguously. With the goals of enabling and supporting automation-of-automation approaches for such systems in general, we have developed a formal model, based on the RIVA model presented in [3, 4], which satisfies these requirements by defining a simplified plant representation that considers the possibility of flow through its components.

This model, which we denote as *flow allowance model*, is generic enough to represent practically any type of plant and plant device, as it only requires knowledge about the plant's structure and the general flow-related behaviour of its elements. With the help of this model, it becomes possible to formally define the structure, type, state and safety of product flow paths, and use these formalisms as the base of automated solutions. Additionally, in many cases this model may be automatically created from machine-readable plant representations such as CAEX [1], thereby simplifying the application of this approach further.

Figure 2 shows a graphical representation of the structural model which corresponds to the plant of Figure 1, where the tanks, pumps, valves, pipes and pipe joins of the plant are all represented as elements uniformly, having an adequate set of connectors which are connected with those of other elements in a one-to-one fashion.

The Route Control programming package for SIMATIC PCS 7 [12] enables the automatic routing of products in flexible plants, as well as the administration, control and monitoring of these routes. However, it requires an engineering phase where partial routes are to be defined and configured manually using a traditional approach. In turn, we seek to develop a simple and fully automatic technique for flow path safety monitoring based on our abstract model of plant structure and plant working state.

The Multilevel Flow Modelling presented in [5] is a methodology for modelling goals and functions of complex processing plants, with the intention of aiding in the development of diagnostics and control systems. It considers not only the flow of mass but also that of energy, and considers for each case source, sink, storage, balance, transport and barrier elements. However, an explicit treatment of flow paths as entities which may identified in a plant is not given.

In [9], a technique for the automatic discovery of product flow paths is presented, and [8] outlines a mechanism for the automatic assurance of product flow paths which is inspired by a similar approach used in railway locking [6]. In this paper, we present a formalisation of the safety of a product flow path at a given plant state based on our model of flow allowance. This formulation may then be used as a guideline for automating the construction of systems which perform safety monitoring of product flow paths.

An outline for the design of such a system following the decentralisation scheme used in [9, 8] is also presented.

1 Abstract Plant Model

In this section we give a formal definition of our abstract plant model, which encompasses the structure and flow allowance of the plant, as well as the product flow paths of the plant.

Definition 1: Plant Structure. A plant is a tuple

$$(T, E, C, \tau, \varepsilon, \circ)$$

where

- T is a set of element types,
- E is a set of plant elements,
- C is a set of connectors,
- $\tau : E \rightarrow T$ is a function such that $\tau(e)$ is the type of element e for every $e \in E$,
- $\varepsilon : C \rightarrow E$ is a function such that $\varepsilon(c)$ is the element of connector c for every $c \in C$,
- $\circ \subseteq C \times C$ is the connection relation, which is
  - irreflexive: $c_1 \circ c_2 \Rightarrow c_1 \neq c_2$
  - symmetric: $c_1 \circ c_2 \Leftrightarrow c_2 \circ c_1$
  - functional: $(c_1 \circ c_2 \wedge c_1 \circ c_2) \Rightarrow c_2 = c_2$

  for any $c_1, c_2, c_2 \in C$.

For every $t \in T$, the set E_t of elements of type t is defined such that

$$e \in E_t \Leftrightarrow \tau(e) = t$$

For every $e \in E$, the set C_e of connectors of element e is defined such that

$$c \in C_e \Leftrightarrow \varepsilon(c) = e$$

Similarly to the approach presented in [9, 8], a plant is formally represented by a tuple $(T, E, C, \tau, \varepsilon, \circ)$ with sets of element types T, elements E and product connectors C. The function τ maps every element to its type, and the function ε maps every connector to the element which owns it. Finally, the binary relation $\circ$ represents the interconnection of element connectors as is found in the physical plant. The plant is hereby modelled by a special kind of graph: the elements of the plant are represented by graph nodes (E), and rather than connecting the nodes directly, the edges of the graph ($\circ$) link so-called connectors (C), which are in turn embedded in the element nodes (ε) as shown in Figure 2.

This corresponds on the one hand to typical plant-engineering representations like CAEX [1], and on the other hand allows for attributing the connectors with flow allowances in the following.

Having this formal representation of the structure of a plant, we wish to model the flow of products through this plant structure. As discussed in [9], we follow an approach for representing product flow which is *plant-oriented* (it considers characteristics of the plant itself such as connection structure and working state of the plant components, rather than the actual physical properties of the material) and *passive* (it considers the *possibility* of flow through the plant, rather than the *causes* of flow or the *actual* flow). We denote this possibility of flow through the plant as *flow allowance*, which represents a *necessary* condition for actual product flow. Therefore, the absence of flow allowance guarantees the absence of product flow in the plant, and we use this reasoning when defining the structure and safety of product flow paths.

The definition of flow allowance may be introduced by an analogy: an element is similar to a room with multiple and rather sophisticated doors, which correspond to its connectors. A door may be used for entering or exiting the room exclusively, or may be used simultaneously as an entrance and as an exit. Furthermore, some doors may be opened and closed, and this may be done for the entrance and exit aspects independently. In a similar way, product may flow into and out of an element through its connectors. The labelling of a connector as an entrance or an exit, as well as its ability to open and close, is described by the *flow allowance behaviour* of the connector, which is defined in the following.

Definition 2: Flow Allowance Behaviour. Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

the *flow allowance behaviour function* of the plant is a mapping

$$\beta : \quad C \rightarrow \{0,1,{}^{0}_{1}\} \times \{0,1,{}^{0}_{1}\}$$

For a given $c \in C$ where $\beta(c) = (i, o)$, we denote i as the input flow allowance behaviour of c and o as the output flow allowance behaviour of c.

The *input flow allowance behaviour function*

$$\beta^I : \quad C \rightarrow \{0,1,{}^{0}_{1}\}$$

is defined such that

$$\beta^I(c) = i \Leftrightarrow \beta(c) = (i, o) \quad \text{for any o} \in \{0,1,{}^{0}_{1}\}$$

The *output flow allowance behaviour function*

$$\beta^O : \quad C \rightarrow \{0,1,\tfrac{0}{1}\}$$

is defined such that

$$\beta^O(c) = o \Leftrightarrow \beta(c) = (i,o) \text{ for any i} \in \{0,1,\tfrac{0}{1}\}.$$

The terms *input* and *output* in this definition respectively refer to the flow which *enters* and *leaves* the corresponding element $\varepsilon(c)$ through the connector c, as shown in Figure 3.

The flow allowance behaviour of a connector is expressed in *ternary logic*. The ternary value 0 corresponds to the Boolean value 0 and represents a constant inhibition of flow. In turn, the ternary value 1 corresponds to the Boolean value 1 and represents constant allowance of flow.

Finally, the ternary value $\tfrac{0}{1}$ corresponds to both Boolean values, and represents a switchable behaviour which may either inhibit or permit flow at a given plant state. This allows us to model the flow allowance behaviour of almost any type of plant element; Figure 4 shows some examples of this.

As the flow allowance behaviour of a connector describes the possible flow allowance configurations of a connector at any given time, the composition of the flow allowance behaviour of the connectors of an element e describe the possible flow allowance configurations of e, and this compositional approach may be further used to describe the possible flow allowance configurations of the entire plant.

These configurations correspond to the *flow allowance states* of the plant, which is defined using the function λ which maps each ternary logic value to the set of corresponding Boolean values:

$$\lambda(0) = \{0\}, \lambda(1) = \{1\}, \lambda(\tfrac{0}{1}) = \{0,1\}.$$

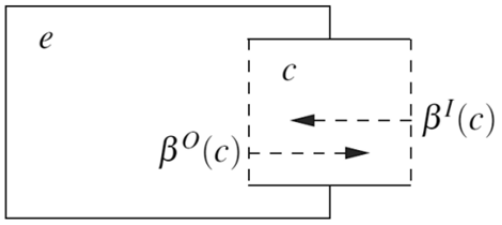


Figure 3. Flow allowance behaviour of a connector c of an element e: $\beta^I(c)$ refers to the flow which enters e through c, and $\beta^O(c)$ refers to the flow which leaves e through c.

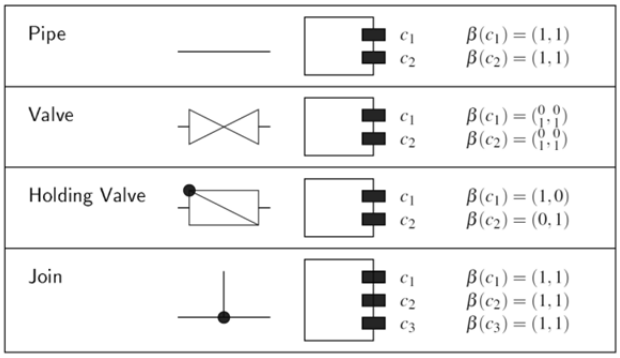


Figure 4. Modelling of various types of plant elements. A pipe is an element with two connectors which always allow flow in both directions. A valve is a 2-connector element whose connectors may allow or inhibit flow in both directions. A holding valve is a 2-connector element where one connector always allows incoming flow only while the other always allows outgoing flow only. Finally, a 3-way pipe join is a 3-connector element whose connectors always allow flow in both directions.

Definition 3: Flow Allowance State. Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

and its flow allowance behaviour function β, the set of *flow allowance states* of the plant is the set of mappings $S \subseteq [C \rightarrow \{0,1\}\times\{0,1\}]$ such that for every $\sigma \in S$ and every $c \in C$ it holds that

$$\sigma(c) = (i,o) \Rightarrow [i \in \lambda(\beta^I(c)) \wedge o \in \lambda(\beta^O(c))].$$

For a given $\sigma \in S$ and $c \in C$ where $\sigma(c) = (i,o)$, we denote i as the *input flow allowance state* c at σ, and o as the *output flow allowance state* of c at σ.

For a given $\sigma \in S$, the input flow allowance state function $\sigma^I : C \rightarrow \{0,1\}$ is defined for any $o \in \{0,1\}$ such that

$$\sigma^I(c) = i \Leftrightarrow \sigma(c) = (i,o)$$

For a given $\sigma \in S$, the output flow allowance state function $\sigma O : C \rightarrow \{0,1\}$ is defined for any $i \in \{0,1\}$ such that

$$\sigma^O(c) = o \Leftrightarrow \sigma(c) = (i,o)$$

The flow allowance state of a connector is expressed in Boolean logic, where 0 represents the inhibition of flow and 1 represents the allowance of flow. Also, the flow allowance states of the plant are determined by the flow allowance behaviour function β as expected. For behaviours with values of 0 or 1, the corresponding values of the flow allowance states are fixed and fully determined by the model, as in the case of static plant elements like pipes or tanks.

However, when a flow allowance behaviour has the value 01 , the actual value of a state may be either 0 or 1. At a given time during the operation of the plant, the physical state of the corresponding plant element determines the actual value of the flow allowance state. As these values are commonly available to a process control system, e.g. from the acknowledgement signals of controllable valves, we may assume that flow allowance states are known when developing algorithms which are based on this model.

Based on this formulation of the input and output flow allowance at every connector in the plant, we may now describe the allowance of flow of material among neighbouring plant elements by means of a binary relation over the set E.

Definition 4: Flow Allowance Relation. Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

and its flow allowance behaviour function β, the *flow allowance relation* $\rightarrow \subseteq E \times E$ is defined such that $e_1 \rightarrow e_2$ if and only if there exist connectors $c_1 \in C_{e_2}$ and $c_2 \in C_{e_2}$ such that $c_1 \circ c_2$, $\beta^O(c_1) \neq 0$ and $\beta^I(c_2) \neq 0$.

Definition 5 Flow Allowance Relation at a State. Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

and a flow allowance state $\sigma \in S$, the *flow allowance relation* $\overset{\sigma}{\rightarrow} \subseteq E \times E$ *at* σ is defined such that $e_1 \overset{\sigma}{\rightarrow} e_2$ if and only if there exist connectors $c_1 \in C_{e_2}$ and $c_2 \in C_{e_2}$ such that $c_1 \circ c_2$, $\sigma^O(c_1) = 1$ and $\sigma^I(c_2) = 1$.

The meaning of these flow allowance relations is an intuitive one: $e_1 \rightarrow e_2$ whenever it may be possible for a product to flow directly from e_1 to e_2 according to the flow allowance behaviour of the intermediate connectors, which in turn occurs whenever there exists a flow allowance state $\sigma \in S$ that permits such a flow; consequently, $e_1 \overset{\sigma}{\rightarrow} e_2$, whenever the flow allowance state σ permits a direct flow from e_1 to e_2. In both cases, we say that there exists a flow step from e_1 to e_2.

As the flow allowance relations describe individual flow steps in a plant, a natural extension of this concept is to chain several flow steps together in order to obtain a flow path. Indeed, this is the basic idea behind our definition of a product flow path: a flow path is a finite sequence of neighbouring plant elements which may be used by a product to flow from an initial element to a final element, and where each pair of consecutive elements conforms a flow step. This is formalised in the following definition, where the notation X^+ denotes the set of all non-empty sequences of elements of set X.

Definition 6: Flow Paths. Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

and its flow allowance relation $\rightarrow \subseteq E \times E$, the *set of flow paths* $P \subset E^+$ of the plant is defined such that

$$e_1 e_2 \ldots e_n \in P$$

where $n \geq 1$, if and only if the following hold:

- $e_i \rightarrow e_{i+1}$ for every $i \in [1, n-1]$,
- $i \neq j \Rightarrow e_i \neq e_j$ for every $i, j \in [1, n]$.

For a flow allowance state $\sigma \in S$ of the plant, the set $P_\sigma \subseteq P$ of open flow paths at σ is defined such that $e_1 e_2 \ldots e_n \in P_\sigma$ if and only if $e_i \overset{\sigma}{\rightarrow} e_{i+1}$ for every $i \in [1, n-1]$.

The function $\kappa : P \rightarrow 2^E$ maps every flow path $p = e_1 e_2 \ldots e_n$ to the set $\{e_1, e_2, \ldots, e_n\}$ of elements it contains.

A flow path may be also seen as simple path in a *flow allowance graph* with a set of nodes E and a set of edges $\rightarrow$. Furthermore, an open flow path at a flow allowance state σ is a flow path in the plant which additionally follows the flow allowance relation $\overset{\sigma}{\rightarrow}$, which may be seen as a simple path in a *flow allowance graph at* σ with a set of nodes E and a set of edges $\overset{\sigma}{\rightarrow}$. Figure 5 depicts a flow path in graphical model representation, and Figure 6 shows a corresponding open flow path.

Apart from having to follow the flow allowance relation $\rightarrow$, flow paths must also be free from any repeated elements. We have included this restriction for several reasons. First, disallowing repeated elements in a flow path causes flow paths to be free of cycles, and therefore, guarantees that flow paths have a finite length (given that the plant is itself finite). This simplifies the representation of flow path data and the algorithms which work with flow paths. Second, allowing repeated elements would permit paths of the form $\ldots e_1 e_2 \ldots e_2 e_1 \ldots$, which would model simultaneous bidirectional flow between the elements e_1 and e_2. As the intention of this work is to model physical product flow, we must rule out any form of simultaneous bidirectional flow from our model.

A flow path where no element appears twice represents a form of 'forward only' flow, which best describes the actual flow of products in processing plants.

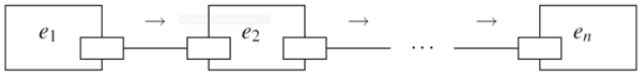


Figure 5. A flow path $p = e_1\, e_2 ....e_n$.

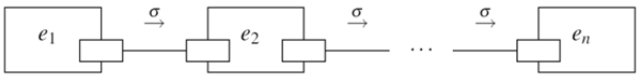


Figure 6. An open flow path $e_1\, e_2 ....e_n$ at a flow allowance state σ.

Finally, our intention is to define the concept of a product flow path in a way which is useful to describe and specify the spatial bounds of general product flow operations in a plant. We feel that the model of simple flow paths presented here which describes product flow between two plant locations captures the essence of these operations.

More complex operations involving recirculation (cycles), forking and joining of flows may be also described using our model by using multiple flow paths. Therefore, we achieve a model which is accurate and comfortable to work with while at the same time expressive enough to handle both simple and complex plant designs.

2 Product Flow Path Safety

The notion of safety in processing plants is a very broad and important part of the corresponding engineering field [7]. Based on our flow allowance model, we formulate a definition of safety of a product flow path at a given plant state with the intention of identifying general scenarios which correspond to undesired and potentially hazardous situations that may arise during the usage of a product flow path. Though this formulation is sufficient for most applications, refinements are possible and may be necessary in special cases.

We identify two subclasses of plant elements, namely *sources* and *sinks*. The former are elements which yield product that may flow to other points in the plant, such as tanks or input nozzles; the latter are elements which consume product which flows from other points in the plant, such as tanks or output nozzles.

A similar classification which describes the general function of an element with respect to product flow may be found in [5].

Definition 7: Sources and Sinks. Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

the set $E^{\uparrow} \subseteq E$ is the *set of sources* of the plant, and the set $E^{\downarrow} \subseteq E$ is the *set of sinks* of the plant.

Usually, product flow paths will begin at a source element and end at a sink element, although we do not require this when asserting their safety. The basic principle behind the definition of product flow path safety is the avoidance of leaks and unintended mixtures, in a similar way to how a train interlocking system avoids derailments (similar to leaks) and collisions (similar to mixtures) whenever a train travels through a given track segment [6]: derailments occur when a train (or a part of it) deviates from its intended course; collisions occur when an approaching train enters a track segment already in use by another train.

With the help of this analogy, we may characterise these two situations for the case of processing plants as follows:

- *Leaks.* A leak occurs when the flow of a product diverges from the intended path *p*, and in our model, when there exists a diverging open flow path *p'* which begins at an element in *p* and which allows flow to a sink element.
- *Unintended mixtures.* An unintended mixture occurs when another material is able to flow into, and mix with, the product flowing through a flow path *p*, and this occurs when there exists a joining open flow path *p'* which begins at a source element and which allows flow to an element in *p*.

The conditions for the occurrence of these situations may be determined in a step-wise manner, as shown in Figure 7. In a flow path $p = e_1\, e_2 ....e_n$, the flow step represented by $e_1 \rightarrow e_2$ is safe if flow may occur from e_1 to e_2 exclusively. Additional outgoing flow from e_1 and additional incoming flow to e_2 are violations of this principle, as they represent leaks and mixtures respectively. Thus, we may determine the safety of a flow step by verifying the impossibility of these additional flows. The safety of a flow path may now be determined by the application of this rule to each flow step in the path, as shown in Figure 8.

A given intermediate element e_i in a flow path p is both start and end of a flow step, and should be therefore free from both leaks and mixtures for p to be safe. On the other hand, the initial element of a p need only be free from leaks, and the final element of a p need only be free from mixtures for p to be safe. This follows from the determination of safety based on the point of view of a flow step as shown in Figure 7, and additionally has the nice property of allowing product to flow into and out of a flow path correctly, that is, via its end points, while considering this safe.

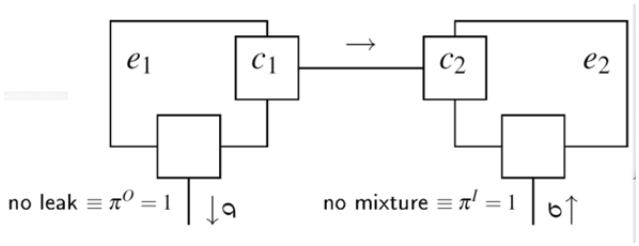


Figure 7. Determination of the safety of a flow step at a flow allowance state σ. A flow step from e_1 to e_2 is safe if there exists no diverging open flow path which begins at e_1 and which reaches a sink, and if there exists no joining open flow path which begins at a source and which reaches e_1. This is determined by the value of the function π^o at e_1 and π^I at e_2.

In order to formalise the absence of leaks and unintended mixtures at an element e with respect to a flow path p and a flow allowance state σ, we present the following definition of *flow protection functions*. As these and some other subsequent functions presented here denote Boolean values, we take the liberty of defining them in terms of first-order logic formulae.

Definition 8: Flow Protection. Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

and a flow allowance state $\sigma \in S$, the *input flow protection function* $\pi^I : E \times P \times S \rightarrow \{0,1\}$ is defined as

$$\pi^I(e, p, \sigma) = \neg[\exists e_1 \ldots e_n e \in P_\sigma : (\{e_1, \ldots, e_n\} \cap \kappa(p) = 0) \wedge e_1 \in E^\uparrow]$$

Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

and a flow allowance state $\sigma \in S$, the *output flow protection function* $\pi^O : E \times P \times S \rightarrow \{0,1\}$ is defined as

$$\pi^O(e, p, \sigma) = \neg[\exists e e_1 \ldots e_n \in P_\sigma : (\{e_1, \ldots, e_n\} \cap \kappa(p) = 0) \wedge e_n \in E^\downarrow]$$

Notice that the negations cause these functions to yield the value 0 when the potentially hazardous situations are present, and 1 when they are absent. Furthermore, for a diverging path to be considered a leak, and for a joining path to be considered a mixture, these paths must not have any element in common with the path p whose safety is being determined, excluding the element e. This causes paths which leave and rejoin p to be excluded from this condition, as well as paths which have a common section with p. This in turn limits the detection of leaks and mixtures to strictly diverging and strictly joining paths.

We may now present the flow step safety function which determines if the flow step represented by $e_1 \rightarrow e_2$ from flow path p is free of leaking deviations from e1 to additional sinks (with the help of the output flow protection function π^O), as well as free of incoming flow from additional sources to e_2 (with the help of the input flow protection function π^I) at a state σ, following the idea from Figure 7.

Definition 9: Flow Step Safety. Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

and a flow allowance state $\sigma \in S$, the *flow step safety function* $\delta : E \times E \times P \times S \rightarrow \{0,1\}$ is defined as

$$\delta(e_1, e_2, p, \sigma) = \left[\pi^O(e_1, p, \sigma) \wedge \pi^I(e_2, p, \sigma)\right].$$

Finally, we may define the safety of a product flow path p at a given flow allowance state σ following the technique shown in Figure 8.

Definition 10: Flow Path Safety. Given a plant

$$(T, E, C, \tau, \varepsilon, \circ)$$

and a flow allowance state $\sigma \in S$, the flow path safety function $\alpha : P \times S \rightarrow \{0,1\}$ is defined as

$$\alpha(p, \sigma) = \alpha'(p, p, \sigma)$$

with help of the auxiliary function α': $P \times P \times S \rightarrow \{0,1\}$ defined as

$$\alpha'(e, p, \sigma) = 1$$
$$\alpha'(e_1 e_2 \ldots e_n, p, \sigma) = = \delta(e_1, e_2, p, \sigma) \wedge \alpha'(e_2 \ldots e_n, p, \sigma)$$

The successive application of the flow step safety function δ to each flow step in the flow path p is achieved by means of a recursive function which defines the safety of p inductively over the structure of p.

In this manner, a simple and unambiguous way of determining the safety of a product flow path based on our abstract plant model is obtained.

3 Decentralised Safety Monitoring of Product Flow Paths

As discussed in [9, 8], the decentralisation of process control systems is advantageous for many reasons (more flexibility, scalability and maintainability than centralised or monolithic systems, more robust handling of errors and service interruptions, easier modifications and upgrades of system components, computation independence thanks to locality, support for dynamic adaptability to new contexts, support for system synthesis for specific cases, etc.). A system for product flow path safety monitoring which operates in a decentralised fashion may be developed using the model-based definition of product flow path safety presented in this paper. This section gives a general description of the composition and operation of such a system.

We follow the decentralised component-based scheme presented in [9, 8]. Every plant element $e \in E$ is assigned a component of the system which controls and monitors the element, and which has connection ports for every connector $c \in C_e$. These ports are interconnected through bidirectional communication links in accordance with the relation $\circ$. Thus, the structure of the decentralised system is an analogy of the plant layout. Each component interacts with each of its neighbouring components by sending and receiving messages. A realisation of this scheme may be accomplished using IEC 61131-3 function blocks [2] for the components, which are common in process control systems. Furthermore, such a system may be automatically constructed from a flow allowance model of the plant by instantiating, parametrising and linking component blocks. This offers a simple and effective technique for synthesising systems such as the one outlined in this paper.

According to Definition 10, the function α is inductively defined over the structure of the flow path, meaning that the safety of a flow path p is determined incrementally in terms of each flow step of p, which in turn corresponds to the recursive invocations of α'. Therefore, we may use this same 'calling' scheme for the definition of the messaging scheme of the components.

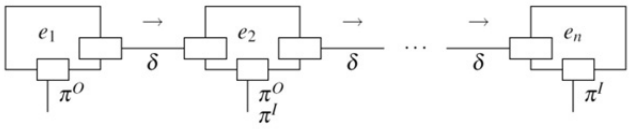


Figure 8. Determination of the safety of a flow path at a flow allowance state σ. A flow path $e_1 e_2 \ldots e_n$ is safe if every flow step it contains is also safe: every element except the final element e_n must be free of leaks, and every element except the initial element e_1 must be free of mixtures.

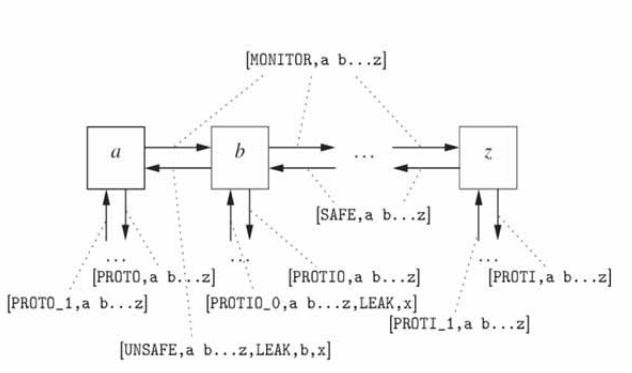


Figure 9. Decentralised, component-based safety monitoring of a flow path. Every element of the plant is represented by a component, and the components are interconnected with each other in the same way as their corresponding elements by means of bidirectional communication links. Messages are sent along the component connections, carrying out the evaluation of the safety functions.

Figure 9 shows the safety monitoring of a flow path *ab....z* by means of adecentralised system. Beginning with the component that corresponds to the first element in the path, a MONITOR message is issued along the components in the path, in correspondence to the evaluation of the function α. The path p is explicitly sent as part of the message, whereas the element e and the flow allowance state σ are given by the context of the execution of every component. The evaluation of the function δ is achieved as follows: if the element which corresponds to the component is not the final element of the path, the function πO is evaluated by issuing PROTO messages in every diverging direction; likewise, if the element which corresponds to the componentis not the initial element of the path, the function π^I is evaluated by issuing PROTI messages in every joining direction.

These messages cause a flow path search to occur, according to the formulation of Definition 8 and similar to the analysis of flow paths presented in [9]. The responses to these messages are either PROTO_1 (resp. PROTI_1), meaning that no offending flow path was found and therefore the value of π^O (resp. π^I) is 1, or they are (resp. PROTI_0) meaning that an offending flow path was PROTO_0 found and therefore the value of π^O (resp. π^I) is 0. Additionally, the offending path is also sent in the message for reporting purposes. In order to reduce the number of messages sent, PROTO and PROTI messages are combined into a single PROTIO which performs the evaluation of both protection functions simultaneously.

After having received a MONITOR message and having sent the corresponding MONITOR, PROTO and PROTI messages, a component waits for a response to every message sent. When all responses have been obtained, the component may issue back a response to the original MONITOR message. In this case, two possible messages may be issued according to the responses received: SAFE corresponds to the case where α has the value 1, and UNSAFE corresponds to the case where α has the value 0. In the latter case, additional information regarding the reason forthis determination may be included. This chain of messages eventually reaches the component which corresponds to the initial element of the path, which may then determine the safety value of α for the entire path p.

4 Summary

A basic and essential operation performed by processing plants is the movement of material, i.e. products, between plant elements. Product flow paths are the routes through the plant that products may use in order to flow from an initial element to a final element in the plant. Based on an abstract plant model which represents the structure of the plant and the possibility of product flow through its elements, a formalisation of the structure of a product flow path has been presented. Additionally, based on this same model, a definition of the safety of a product flowpath at a given plant state has been given. These results may be used for specifying and automating the tasks of monitoring and assuring the safety of product flow paths in processing plants.

The definition of a product flow path is based on a plant structure model, which defines the elements of the plant and their interconnections by means of embedded connectors, and on a flow allowance model, which defines the ways in which product may flow into and out of every element through each corresponding connector. A product flow path is then defined as a sequence of neighbouring plant elements whose connectors may allow flow in the direction of the flow path at some plant state. Additionally, a flow path is said to be open at a given plant state if this flow can occur at this state.

The safety of a given product flow path is determined in a step-wise manner: a product flow path is safe if every flow step it contains is also safe. A flow step in turn is safe if product may flow exclusively from its first elementto its second element, that is, if its first element is free from leaking deviations to product sinks, and if its second element is free from incoming mixtures from product sources. When applying this safety criterion to every flow step in a flow path, a simple formulation for the safety of a flow path is obtained which guarantees that a product flow path is free from product leaks and unintended product mixtures. Furthermore, it is applicable to any plant for which an abstract plant model is provided.

An outline of a system for product flow path safety monitoring has been presented, which operates in a decentralised manner and according to our model-based definition of product flow path safety. It follows a decentralised, component-based functional abstraction, where every plant element is assigned a component of the system which controls and monitors the element, and which has a connection port for each connector of the element. These ports are interconnected by means of bidirectional communication links, and in analogy to the structure of the plant.

Each component interacts with each of its neighbouring components by sending and receiving messages. These messages correspond to the evaluation of the safety functions as defined in this paper, thus achieving the automaticdetermination of the safety of a product flow path in a decentralised manner. Furthermore, the automatic synthesis of such a system from an abstract plant model may be implemented in a straightforward way by instantiating, parametrising and linking component blocks.

References

[1] DKE, Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE. DIN V 44366:2004-12: *Specification for Representation of process control engineering requests in P&I Diagrams and for data exchange between P&ID tools and PCE–CAE tools.* DIN Deutsches Institut fur Normung e.V., Berlin, 2004.

[2] IEC, International Electrotechnical Commission. IEC 61131-3 Ed. 1.0 en:1993: *Programmable controllers — Part 3: Programming languages.* International Electrotechnical Commission, Geneva, Switzerland, 1993.

[3] R. Jorewitz, U. Epple, A. Munnemann, R. Bockler, W. Wille, R. Schmitz. *Automation of performance monitoring in an industrial environment*. In PCIC Europe 2005, 2nd European Conference on Electrical andInstrumentation Applications in the Petroleum and Chemical Industry, pages 116–120, Basel, Switzerland, Oct. 26-28 2005.

[4] R. Jorewitz, G. Quiros, U. Epple. *Modelling of multiple, semantically-coupled flowgraphs.* In SOLI 2008, IEEE International Conference on Service Operations and Logistics, and Informatics, pages 965–971, Beijing, China, October 12-15, 2008.

[5] M. Lind. *Modeling goals and functions of complex industrial plants.* Applied Artificial Intelligence, 8(2):259–283, 1994.

[6] J. Pachl. *Railway Operation and Control*. VTD Rail Publishing, Mountlake Terrace, USA, 2004.

[7] M. Polke, editor. *Process Control Engineering*. VCH Verlagsgesellschaft, Weinheim, Germany, 1994.

[8] G. Quiros, U. Epple. *From railroads to processing plants: Decentralised automatic assurance of product flow paths.* In EKA 2008, Entwurf komplexer Automatisierungssysteme: Beschreibungsmittel, Methoden, Werkzeuge und Anwendungen, Magdeburg, Germany, 15-17 April, 2008.

[9] G. Quiros, M. Mertens, U. Epple. Function blocks for decentralised analysis of product flow paths. In ETFA 2008, 13th IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany, 15-18 September, 2008.

[10] T. Schmidberger, A. Fay, R. Drath. Automatisiertes Engineering von Prozessleitsystem-Funkionen. Atp - Automatisierungstechnische Praxis, 02/2005, 2005.

[11] S. Schmitz, U. Epple. On rule based automation of automation. In Breitenecker F., Troch I. (Hrsg): 5th MATHMOD 2006: 5th Vienna Symposium on Mathematical Modelling. ARGESIM - Verlag, Vienna, February 2006.

[12] Siemens AG. *Route Control V7.0 SP1 Programming and Operating Manual.* Siemens AG, Automation and Drives, 2007.

Submitted: March, 2010
Accepted: December 5, 2010

Simulation based Planning and Optimisation in Supply Chains: Application in ECLIPS Project

Galina Merkuryeva, Yuri Merkuryev*

Riga Technical University, Department of Modelling and Simulation,1. Kalku Street, Riga LV-1658, Latvia;
merkur@itl.rtu.lv

Abstract. The paper presents simulation-based methodology to solving multi-echelon supply chain planning and optimisation problems. It is aimed to analyse an efficiency of a specific planning policy over the product life cycle within the entire supply chain and to optimise the cyclic planning policy at the product maturity phase. Software prototypes and applications are described in the paper. The presented research is funded by the ECLIPS Specific Targeted Research Project of the European Commission 'Extended Collaborative Integrated Life Cycle Supply Chain Planning System'.

Introduction

For the last decade, supply chain planning has become a critical factor in the success and profitability of an enterprise, especially given a global competition, rapidly changing markets and increasing customer expectations.

Supply chain planning can be defined as a process of coordinating and integrating key logistics activities, i.e. inventory management, production planning, warehousing, etc., from the procurement of raw materials through production to distribution of finished products to the end-customer with the goal to minimise total supply chain cost and maximise customer service level [2].

To manage supply chains, two different approaches are used [5] in practice. For years, researchers and practitioners have primarily investigated so called single echelon approach, where a stage or facility in the supply chain is managed. Recently, however, increasing attention has been placed on the performance, design, and analysis of the supply chain as a whole that allows optimising the global supply chain performance. Indeed, almost every product is produced in a chain of successive processes (either in different companies or different departments within the same company). A multi-echelon environment considers multiple processes (e.g. purchasing, production, picking and transportation) and multiple stock points (buffer or storage).

A variety of planning policies, which are grouped in non-cyclic and cyclic ones, can be used within a multiechelon approach. In cyclic planning, fixed processing (i.e. order, production or delivery) interval lengths are applied to all items, while non-cyclic planning assumes that interval lengths can vary over the planning horizon.

In practice, cyclic policies are more preferable for a multi-product and a multi-stock case, as they easier to control, and reducing administrative costs could reduce higher inventory costs [1]. However, when a customer demand is variable and uncertain, e.g. at the product introduction or end-of-life phases, flexibility in spacing of planning periods can result in lower total costs for the non-cyclic policy.

Simulation technology provides an experimental approach [5] to supply chain analysis and optimisation that allows the analyst easily to: 1) introduce into the multi-echelon cyclic planning procedure variability of demand, lot sizes and processes lead times; 2) model processes that contain nonlinearities, combinatorial relationships and uncertainties; 3) take into account constraints at different echelons of the supply chain. Moreover, by building a virtual reality out of small components and not requiring a rigid structure of the analytical model, a simulation model provides the great flexibility that allows in the planning procedure: 1) validate different assumptions and planning decisions; 2) estimate consequences of planning decisions in time and by echelon; 3) perform a sensitivity analysis of parameters that influence optimality of the cyclic schedule; 4) define optimal planning parameter for each of supply chain nodes during the product maturity phase; and 5) analyse stability of the optimal production schedule under conditions of uncertain demand and finite capacity.

The ECLIPS Specific Targeted Research Project [13] is addressing both academic and business state-of-the-art in supply chain planning and management. Its abbreviation stands for 'Extended Collaborative Integrated Life Cycle Supply Chain Planning System'.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 37-44 | doi: 10.11128/sne.21.tn.10045

The key research is multi-echelon supply chain planning for industries with batch and semi-batch processes on a tactical level spanning the full life cycle of the product. The project is aimed at minimization of total inventories through the whole supply chain, taking into account a product lifecycle, from its introduction into market, through a maturity phase, and finally to an end-of-life phase. In order to achieve this goal, simulation is used intensively in the ECLIPS project [6]. From one hand, it supports supply chain management processes (e.g., optimization and decision making), thus providing conditions for minimization of inventories. From another hand, simulation provides a platform for testing algorithms and tools, being developed within the project. The benefits of the project developments have been proved in practice in the environments of two industrial partners. The results, however, could be exploited to a wide range of industries, e.g., in manufacturing, wholesale and retail sale, and transport.

The project scope in the paper focuses on development of simulation-based methodology and tools for optimizing multi-echelon cyclic planning solutions for products at the maturity phase, and analysing cyclic and noncyclic planning policies over the product life cycle in order to prove in practice efficiency of a cyclic schedule or to switch from a cyclic planning policy to a non-cyclic one.

1 Simulation-based Multi-echelon Cyclic Planning and Optimization

Application of the MILP (Mixed Integer Linear Problem) analytical model in multi-echelon supply chain planning and optimisation is limited by assumptions of a constant demand, fixed set-up costs and lead times. These assumptions significantly decrease the complexity of the problem, but still are considered very useful for mature products [3]. In this context, simulation-based planning and optimization techniques are more flexible and do not require a rigid structure of the analytical model. They allow estimating consequences of different planning policies and decisions in time and by echelon; analysing stability of an optimal production schedule received from the MILP analytical model, and define optimal parameters of a multi-level cyclic schedule under conditions of uncertain demand and finite capacity [5].

In the paper, multi-echelon cyclic planning and optimisation at the product maturity phase is based on integration of analytical and simulation techniques [7]. Analytical formulas are used to obtain initial planning decisions under conditions of stochastic demand and constant or stochastic lead time. Simulation techniques extend these conditions to backlogging and capacity constraints. In this case, the multi-echelon cyclic planning problem is formulated as a simulation-based optimisation problem that is aimed to determine optimal parameters of cyclic schedules at different supply chain echelons.

1.1 Network Conceptual Model

The following are main assumptions that define the scope of a network simulation model: (1) Demand is considered to be uncertain, while predicting the demand mean value, its variations are estimated by a standard deviation of the demand per period; (2) Lead times of the processes are considered to be variable and/or stochastic; (3)

Lot sizes of the products are variable; (4) Capacities are limited; (5) Demand is considered to be independent only for customised products; (5) Backorders are delivered in full; (6) Fixed production and ordering costs, and linear inventory holding costs are assumed; (7) Planning is performed for a finite planning horizon.

A network simulation model [7] itself is built as process oriented model with a one-directional flow of goods. It is presented by two types of atomic elements: stock points and processes that are graphically represented by triangles and rectangles, correspondingly (see Figure 1). Any process with a stock point connected with a directed arc defines a stage. A set of stages that belong to the same network level creates an echelon. Input parameters, decision variables and constraints are assigned to atomic elements. The supply chain generic network is constructed from basic sub-networks, such as linear, convergent and divergent. The replenishment and delivery logic for each sub-network is defined.

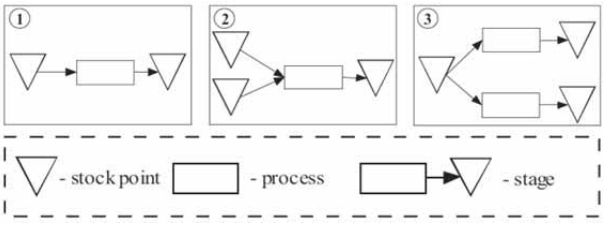


Figure 1. Basic sub-networks of the supply chain.

Average total cost of a cyclic schedule that includes a sum of set-up, ordering and inventory costs is defined as the main network model performance measure. However, in order to avoid unconstrained minimization of the total cost and satisfy customer service requirements, the average order fill rate is introduced as additional performance measure to be analysed in simulation optimisation experiments. It is defined as the percentage of endcustomer's orders filled from the available inventory. As controllable variables, lengths of replenishment cycles and order-up-to-levels for stock points are defined in the network model. These variables determine the reorder period and quantity to be ordered or produced for each mature product in the network.

The main idea of a cyclic schedule is to use fixed order intervals at each stage or echelon while synchronizing these cycles in a multi-echelon supply chain to keep cycle inventory and order costs low. For that, additional cyclical replenishment constraints that define cyclic policy, e.g. power-of-two policy, are introduced.

1.2 Simulation Environment

The simulation environment for cyclic planning and optimisation is built in the ProModel simulation software [4]. It provides automatic generation of the simulation model of a generic network described in the Excel format by using the ProModel ActiveX technology; as well as definition of an initial point for simulation optimization using analytical calculus, and realization of the simulation-based optimization algorithm to find optimal parameters of a multi-echelon cyclic schedule and optimise network simulation model performance measures. Automation capability allows the program to automatically generate simulation models from external applications by using VBA programming language. The ActiveX-based VBA program developed in MS Excel consists of subroutines that provide ProModel operational control and allows accessing the model information, i.e. loading a blank simulation model; definition a title of the model, a path to a graphical library, an animation speed, the simulation length and number of replications; creating entities, locations of stock points and processes, path networks used to establish links between a stock and process points; creating arrays, variables, functions and procedures; and definition of entities arrival schedule, sequence of processes and their operational logic. The simulation environment for cyclic planning and optimisation includes [7] the following components presented in Figure 2:

1. Database component built in the Excel format that contains network and dataset subcomponents. The dataset subcomponent includes basic data about products, costs, capacities, time steps or period in the planning horizon and customer demand.
2. Procedural component by using analytical calculus generates cyclic schedules for different products and contains lot sizing procedures workable under conditions of time-varying demand.
3. Process component where the network is built up and simulated, cyclic schedules are modelled, inventory levels are controlled, and the network performance measures are estimated
4. Optimisation component to find optimal parameters of a multi-echelon cyclic schedule and optimize network simulation model performance measures.

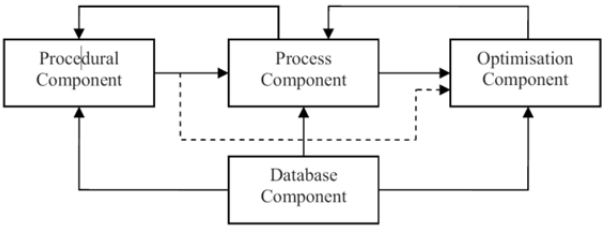


Figure 2. Basic components of simulation environment

Architecture of simulation-based environment, components structure and data exchange processes between these components are described in [8].

1.3 Simulation Optimisation

Within simulation-optimisation component the network simulation model is used in traditional way with a simulation optimiser in negative feedback. Variables controlled in the simulation model, i.e. lengths of replenishment cycles and order-up-to-levels for stock points, define multiple decision variables to be optimised in the problem.

The number of decision variables increases with the number of stock points. As a result, a large number of decision variables in practice could make conducting iterative optimisation experiments difficult. Moreover, two objective functions such the average total costs and the average order fill rate are associated to the network simulation model. As a result, optimisation of multi-echelon cyclic solutions leads to the multi-objective combinatorial optimisation problem. To solve the problem, the simulation-based optimisation algorithm based on the cooperative search of the multiobjective genetic

algorithm (GA) and response surface-based method (RSM) is introduced [9]. While a GA is well suited to solve combinatorial problems and is used to guide the search towards the Pareto-optimal front, RSM-based linear search is appropriate to improve GA solutions based on the local search approach. Let us note that metric scales of decision variables have a very different range of possible values. During simulation experiments, 'order-up-to-levels' type variables are calibrated with a discrete step size.

Multi-objective genetic algorithm [12] is used to find optimal parameters of cyclic schedules, i.e. cycles and order-up-to levels, in each echelon of the supply chain. Starting with the initialisation of an initial population, the following steps are performed per loop iteration. First, the initial population of the pre-defined size is randomly generated and chromosomes are encoded with respect to power-of-two synchronisation policy. Afterwards, fitness values are assigned to population members using Pareto-ranking approach and discrete-event simulation model. Next, penalty function is applied to infeasible solutions in current population. In order to maintain a diverse population and prevent premature convergence, crowding distances of all chromosomes are calculated. The next step represents the mating selection, where individuals are chosen by means of crowded tournament selection.

Finally, after crossover and mutation the new population is replaced by the union of the best parents and mating pool individuals. The user-interface of the developed genetic algorithm is implemented in MS Excel using ActiveX controls.

RSM-based linear search is used [9] to improve cyclic planning solutions of the genetic algorithm by adjusting order-up-to levels that could result in decreasing the total cost and/or increasing the end-customers fill rate. The algorithm is based on local approximation of the simulation response surface by a regression type meta-model in a small region of independent factors and integrates linear search techniques for optimising stock points' orderup to levels. Finally, the Pareto-optimal front initially generated by the GA is updated including solutions found in RSM-based linear search procedure. Solutions received are reordered according to their fitness values in the increasing sequence.

Numerical example that illustrates simulation optimisation algorithm is given in section 4 of the paper.

2 Simulation-based Comparative Analysis of Cyclic and non-cyclic Policies

Evaluation of the difference between performance measures of cyclic and non-cyclic planning policies in supply chains gives possibilities to determine the most efficient planning policy at the product life cycle different phases, and provides a control mechanism for switching from one planning policy to another one. Simulation is defined as the most suitable technique to reveal significant parameters affecting the difference between costs of cyclic and non-cyclic schedules and to investigate the optimality gap [10] between performances of cyclic and non-cyclic planning policies in conditions of demand variability and uncertainty for switching to cyclic planning.

2.1 Simulation-based Methodology for Comparative Analysis

The following main factors that influence the difference between the cost of the cyclic policy and the cost of the non-cyclic policy are analysed in literature [1, 10]: coefficient of demand variation (CODVAR); capacity utilization; and number of planning periods. It is shown in that the coefficient of demand variation is the key factor affecting an additional cost of a cyclic schedule.

In general, the optimality gap is defined as a percentage or ratio measure to investigate how close a solution is to optimum. To measure the gap between performances of planning policies, usually the difference in their costs is expressed as percentage. For this purpose, ACCS performance measure (i.e. an Additional Cost of a Cyclic Schedule) that describes the gap between cyclic and non-cyclic solutions is used:

$$ACCS = \frac{Cyclic\ Solution\ Cost - Noncyclic\ Solution\ Cost}{NoncyclicSolutionCost}$$

Simulation-based scheme for comparative analysis of planning alternatives over the product life cycle is introduced [11] and presented in Figure 3. It allows estimating the difference between the total costs of cyclic and non-cyclic policies, analysing an additional cost of a cyclic schedule and making a decision about application of an appropriate policy. As input data, parameters of non-cyclic and cyclic policies are determined using either analytical calculus or simulation optimisation techniques.

Here, cycles and order-up-to levels are used as parameters of a cyclic planning policy, while a non-cyclic policy is defined by reorder points and order quantities per each supply chain echelon. For a cyclic planning policy the optimal parameters received from simulation optimisation component are used. Supply chain simulator model behaviour of alternative planning policies, and correspondent performance measures, i.e. the total costs mean values and correspondent ACCS values, are received from simulation experiments. Cost comparison for planning alternatives requires a careful analysis to ensure that the differences being observed are attributable to actual differences in their performances and not to statistical variations. This is done by analysing steady-state behaviour of the network simulation model, performing multiple simulation replications for each planning policy and comparing average results received from replications.

To determine the most efficient planning policy at a specific phase of the product life cycle, simulation-based switching algorithm is developed that contains the following phases: cost comparison for planning alternatives based on testing statistical hypotheses in the first phase, and ACCS analysis based on a set of supply chain parameters in the second phase. Cost comparison for alternative policies is based on estimation of the difference between their total costs mean values through simulation experiments by using the Paired-t confidence interval method. It is aimed to discover if two mean values are significantly different. Confidence level is defined at least at 95%. Two statistical hypotheses, i.e. the null hypothesis H0 and an alternative hypothesis H1, for making these comparisons are intro duced and tested. The null hypothesis supposes that there is no a significant difference between total costs mean values for two policies.

Let μ_{cyclic} and $\mu_{non\text{-}cyclic}$ define the true mean value of total costs for cyclic and non-cyclic policy, correspondingly, and $\mu_{cyclic} - \mu_{non\text{-}cyclic}$ or $\mu_{(cyclic\text{-}Non\text{-}cyclic)}$ defines the difference between mean values of total costs for two policies. In Paired-t notation these statistical hypotheses are formulated as follows: H0: $\mu_{(cyclic\text{-}Non\text{-}cyclic)} = 0$ and H1: $\mu_{(cyclic\text{-}Non\text{-}cyclic)} \neq 0$. While testing statistical hypotheses, it is supposed that simulation observations are independent, normally distributed and a number of observations received for two policies are equal. Based on testing of statistical hypotheses H0 and H1, the following conclusions are made.

If the Paired-t confidence interval excludes zero with a probability 1-7, then μ_{cyclic} is significantly different from the $\mu_{non\text{-}cyclic}$ with 7 significance level (Figure 4, position (b) and position (c)). In case of $\mu_{cyclic} < \mu_{non\text{-}cyclic}$ (Figure 4, position (c)) the cyclic planning policy outperforms non-cyclic one. Otherwise, if the Paired-t confidence interval includes zero (Figure 4, position (a)) with a probability 1-7, the null hypothesis H0 is failed to reject, and there is no a significant difference between the mean costs for two policies, i.e. μ_{cyclic} is not significantly different from the $\mu_{non\text{-}cyclic}$ with 7 significance level. In this case, the final decision is based on the ACCS analysis.

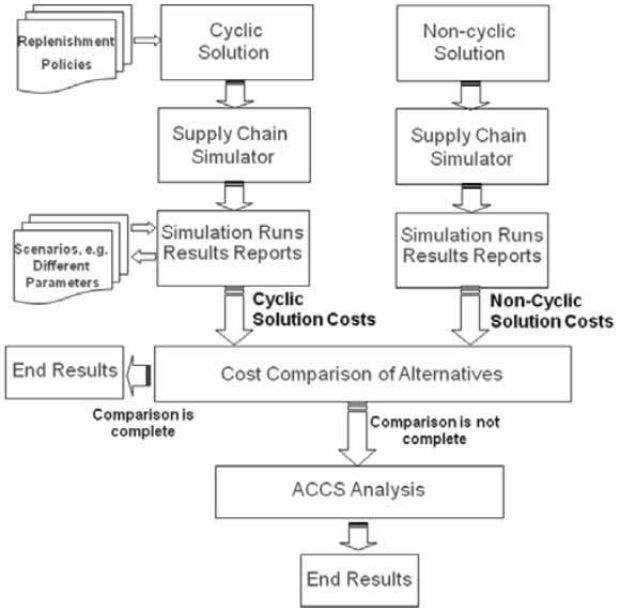


Figure 3. Comparative analysis of planning alternatives through simulation experiments

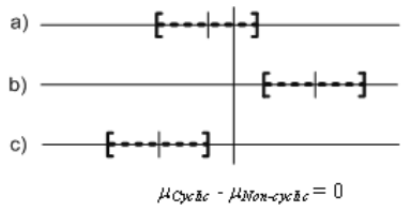


Figure 4. Hypotheses testing using Paired-t confidence intervals.

An additional cost of a cyclic schedule is estimated by the mean ACCS value. The width of the ACSS confidence interval is used to indicate accuracy of the ACCS estimate.

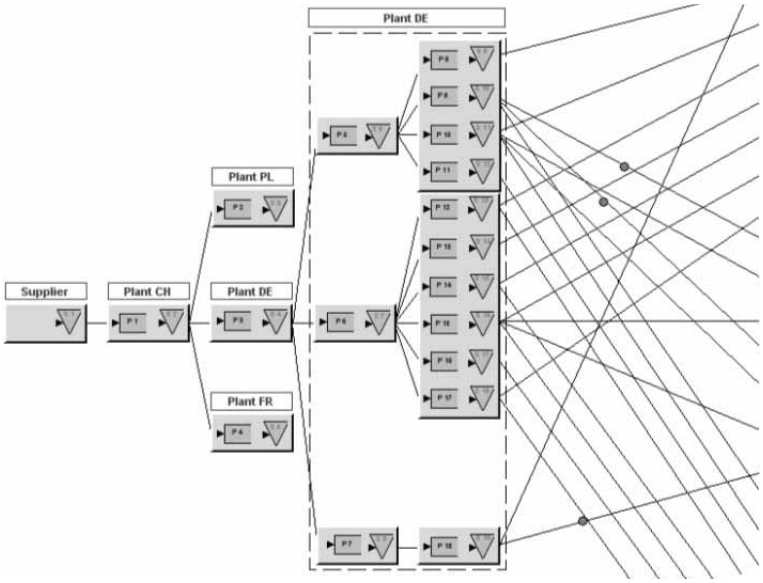


Figure 5. Simulation model screenshot

The mean ACCS value received from simulation experiments is compared with the critical one (or the maximum allowed) by using IF-THEN production rules. ACCS critical values are fixed by an application expert, refined within simulation-based analysis and used as a threshold for final decision making.

2.2 Software Prototype

The developed software prototype that allows analysing efficiency of planning policies and determining the switching from one planning policy to another is developed using ProModel, MS Excel and VBA integration possibilities. It includes the following main blocks:

- *Modelling & Simulation* module controls the input of initial parameters of the simulation model and planning policies; initialize the simulation model run within ProModel Software and export the output data from the simulation model to the MS Excel format. The network simulation model itself consists of three sub modules that simulate two alternative planning policies and estimate ACCS performance measure.
- *Switching Module* recognizes the switching moment from non-cyclic to cyclic policy by performing two types of analysis, i.e. Cost Comparison of planning alternatives by using Paired-t confidence interval method, and following ACCS analysis on a set of parameters.
- *Advanced Analysis* on a Parameter Set performs sensitivity analysis of parameters influences ACCS values; What-If Analysis and off-line gap investigation.

3 Application

The application itself is aimed to find an optimal cyclic plan of a chemical product, i.e. liquid based raisin, in order to minimise inventory holding, ordering and production costs, and maximise end-customers fill rate. As a test bed, the chemical manufacturing supply chain is used. The main operations occurred in the supply chain network are the following. In the plant CH (see, Figure 5), the raw material is converted to the liquid based raisin.

It is then either sourced to direct customers or shipped to the plant DE, where other components are added to make different products. From that plant, the end-products are shipped to different customers. The ProModel-based network simulation model is generated automatically using a simulation-based environment described in Section 2.2.

The end-customer demand is normally distributed; and cycles are defined according to the power-of-two policy. Cycles are presented in weeks as follows, 7, 14, 28, 56, where 56 days is the maximal cycle that corresponds to one full turn of a planning wheel. Initial stocks are equal to order-up-to levels plus average demand multiplied by cycle delays. Stock point 1 has infinite on hand stock and is not controlled by any policy. Backorders are delivered in full.

Simulation run length is equal to 224 periods. This allows modelling of four full turns of the planning wheel, i.e. 4*56 periods. Number of simulation replications is equal to 5. The GA is executed with the following parameters: the population size is 40; crossover and mutation probabilities are 0.5 and 0.1, correspondingly; a tournament size is equal to 2. The GA works with 66 decision variables assigned to network stock points. Initial values of order-up-to levels are calculated analytically. When the number of generations with a stagnant nondomination set is equal to 3, the GA is terminated.

Figure 6 shows solutions received from the final population that includes three non-dominated solutions with performance average measures 1) total cost = 787,431, fill rate = 100.00; 2) total cost = 766,669, fill rate = 98.88; and total cost = 752,300, fill rate = 93.76.

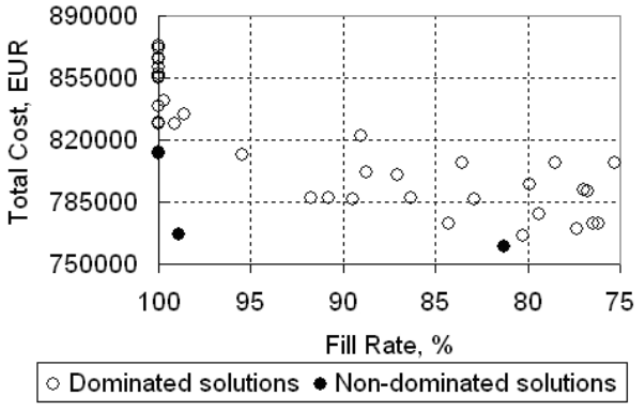


Figure 6. Final GA population mapped in the objective space.

RSM-based linear search algorithm is used to adjust order-up-to levels of three non-dominated solutions received with the GA while fixing stock points' cycles. The updated Pareto-optimal front contains three non-dominated solutions found by the GA, where the second solution is improved by the RSM-based linear search algorithm with the average total cost and average fill rate equal to EUR 756,178 and 98.88%, respectively.

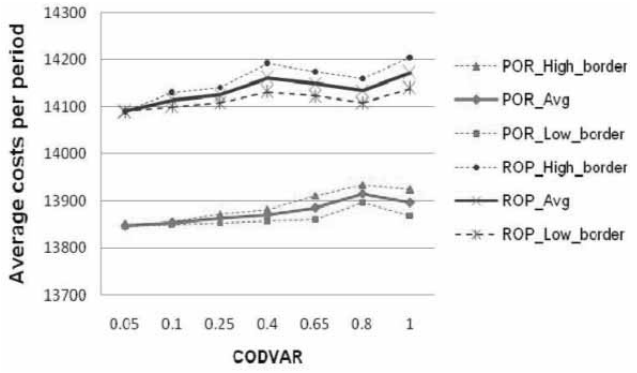


Figure 7. The average cost values per period as function of CODVAR.

The results of simulation-based analysis of performances of multi-echelon cyclic and non-cyclic planning policies are presented in Figure 7 and show that the average total cost per period and its confidence interval increase as demand variability CODVAR. Here, CODVAR coefficient is limited by 1.

Here, process lead times are assumed to be constant, and confidence intervals are estimated with 95% of confidence. The difference between total costs average values for cyclic and non-cyclic policies always stays negative and leads to conclusion that in this case the cyclic solution is more preferable then non-cyclic one.

4 Conclusions

The paper describes simulation-based methodology and tools for analysis and optimisation of planning policies over product life cycle within the entire supply chain. Simulation optimisation is used to define the optimal parameters of cyclic planning policies for mature products by integrating the multi-objective genetic algorithm and RSM-based linear search.

Simulation-based comparison analysis provides estimating the difference between the total costs of cyclic and non-cyclic policies, analysing an additional cost of a cyclic schedule and allows determining the most efficient planning policy at the product life cycle different phases, providing a control mechanism for switching from one planning policy to another one and finally improving the product life cycle management.

The application described and presented results demonstrate efficiency of the proposed methodology.

References

[1] G. M. Cambell, V.A. and Mabert. *Cyclical schedules for capacitated lot sizing with dynamic demands*. Management Science, 37 - 4 (1991), 409‑427.

[2] M. Christopher. *Logistics and Supply Chain Management: 3rd ed.* Prentice Hall, 2005.

[3] M. Gregoire, E. Jacquet-Lagreze, G. Merkuryeva, C. Rebai, S. Timmermans, C. Verlhac, V. De Vulpillieres. *Cyclic planning in a multi-echelon network. Livre des resumes*. In :Proc. Conference scientifique conjointe en Recherche Operationnelle et Aide a la Decision FRANCORO V / ROADEF2007, 2007, Grenoble, 221- 222.

[4] C. Harrell, B. K. Ghosh, R. O. Bowden. *Simulation Using Promodel: 2nd ed*. McGraw-Hill Higher Education, 2007.

[5] Y. Merkuryev, G. Merkuryeva, B. Desmet, E. Jacquet-Lagreze. *Integrating Analytical and Simulation Techniques in Multi-Echelon Cyclic Planning*. In: Proc. First Asia International Conference on Modelling and Simulation, Asia Modelling Symposium 2007, Prince of Songkla University, Phuket, Thailand, Eds.: Al-Dabas, D., Zobel, R., Abraham A. and Turner, S., IEEE Computer Society, 2007, 460‑464.

[6] Y. Merkuryev, G. Merkuryeva, J. Hatem, B. Desmet. *Supply Chain Simulation in the ECLIPS Project*. In: Proc. 2nd Asia International Conference on Modelling and Simulation, AMS 2008, Kuala Lumpur, Malaysia, Eds.: Al-Dabass, D., Turner, S., Tan, G. and Abraham, A., IEEE Computer Society, 2008, 684-690.

[7] G. Merkuryeva, Y. Merkuryev, L. Napalkova. *Simulation-Based Environment for Multi-Echelon Cyclic Planning and Optimisation*. In: Proc. International Mediterranean Modelling Multiconference I3M 2007, Bergegi, Eds.: Bruzzone, A., Longo, F., Merkuryev, Y., and Piera M.A., University of Genoa, 2007, 318-325.

[8] G. Merkuryeva, L. Napalkova. *Development of simulation-based environment for multi-echelon cyclic planning and optimization.* In: Proc. 6th EUROSIM Congress on Modelling and Simulation, EUROSIM 2007, Ljubljana, Slovenia, 2007, Eds.: Zupan_i_, B., Karba, R. and Blaži_, S., EUROSIM/SLOSIM, 2007, ID 452.

[9] G. Merkuryeva, L. Napalkova . *Two-Phase Simulation Optimisation Procedure with Applications to Multi-Echelon Cyclic Planning*. In: Proc. 20th European Modelling and Simulation Symposium EMSS- 2008, Campora San Giovanni, Italy, Eds.: Bruzzone, A., Longo, F., Piera, M.A., Aguilar, R. and Fridman C., 2008, 51-58.

[10] G. Merkuryeva, S. Timmermans, O. Vecherinska. *Evaluating the 'optimality gap' between cyclic and non-cyclic planning policies in supply chains*. In: Proc. International Conference on Production Engineering - 2006, Ed.: Chlebusa, E., Oficyna Wydawnicza Politechniki Wroclawskiej, Wroclaw, 2006, 155‑162.

[11] G. Merkuryeva, O. Vecherinska. *Simulation-Based Approach for Comparison of (s, Q) and (R, S) Replenishment Policies Utilization Efficiency in Multi-echelon Supply Chains*. In: Proc. 10th International Conference on Computer Modelling and Simulation EUROSIM/UKsim-2008, Eds.: Al-Dabas, D., Orsoni, A., Abraham, A. and Zobel, R., IEEE Computer Society, 2008, 434-440.

[12] L. Napalkova, G. Merkuryeva. *Theoretical Framework of Multi-Objective Simulation-Based Genetic Algorithm for Supply Chain Cyclic Planning and Optimisation*. In: Proc. 10th International Conference on Computer Modelling and Simulation EUROSIM/UKsim-2008, Eds.: Al-Dabas, D., Orsoni, A., Abraham, A., Zobel, R., IEEE Computer Society, 2008, 467-474.

[13] www.eclipsproject.com/portal/ ECLIPS web site.

Submitted: March, 2010
Accepted: December 10, 2010

A Probability Model for TCP Slow Start

Asmir Gogic*, Nermin Suljanovic, Amer Hasanovic, Aljo Mujcic

[1] University of Tuzla, Faculty of Electrical Engineering Franjevacka 2, 75000 Tuzla, Bosnia and Herzegovina;
* asmir.gogic@untz.ba

Abstract.
This paper presents one approach to modeling of TCP connection during the slow start phase. Such modeling can be used for TCP connection analysis with reduced computation complexity compared to the packet-level simulators. Proposed model is validated by comparing the results obtained from ns-2 simulations.

Introduction

Application protocols mainly used on the Internet, such as HTTP or SMTP, use TCP protocol for reliable transport. TCP connection performance analysis can be carried out in two different ways. First one is simulation of TCP connection at the packet-level. This approach leads to accurate results but also requires long simulation time. Alternative approach is to model the TCP behaviour analytically, significantly reducing simulation time while simultaneously keeping accuracy at an acceptable level.

This paper reviews models available in the literature and proposes an alternative analytical TCP model during the slow-start phase. TCP behavior can be described with different models. Packet-level models are the most accurate since they employ full TCP stack implementation. However, when analyzing large-scale networks or simple networks with high throughputs, packet- level simulators are impractical due to long simulation time. An alternative approach is to use mathematical abstractions to model the TCP behavior. One way to abstract the TCP behavior through mathematical tools is to use differential equations. Another approach is based on probability analysis. In this approach, statistical formulas are used to describe TCP behaviour in different stages. Aggregating these models, the full TCP behavior could be obtained. In this paper, a probability model for the slow-start TCP stage is derived. The derived model is validated by comparing the results with the packet-level simulation tool ns-2 [7].

Finally, future directions for employment of probability models of other TCP stages are given.

1 TCP Protocol

TCP is a reliable connection-oriented transport protocol for packet-switched networks. Reliability is achieved by employing acknowledgements (ACKs) [2]. Using ACKs and sequence numbers, the transmitter keeps the track of packets that are successfully delivered to the receiver. TCP operates in different stages: Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery and Timeout. Transition between stages is determined by packet loss or acknowledgement of predefined number of packets. The window size determines the maximum number of packets that a transmitter may send before receiving the first acknowledgement.

In the Slow Start phase, window size is incremented with every received ACK. Time interval, from departure of the first packet to the last packet in a window, represents round. The window size varies with the rate of the packet loss in the network. Hence, the packet loss probability increases with the number of sent packets due to the congestion in the network. Generally, window size w_i grows in rounds and can be expressed as:

$$w_i = w_{i-1} + \frac{1}{b} w_{i-1} = w_{i-1} \cdot \left(1 + \frac{1}{b}\right) = \gamma \cdot w_{i-1} = w_0 \cdot \gamma^{i-1} \qquad (1)$$

where b is number of packets acknowledge by one ACK, and i is the round number and w_0 is the initial window size. Total number of packets sent in slow start including the round i is

$$ssdata_i = w_0 + \gamma \cdot w_0 + \gamma^2 + \cdots + \gamma^{i-1} \cdot w_0 = w_0 \cdot \frac{\gamma^i - 1}{\gamma - 1} \qquad (2)$$

When a packet loss occurs during the slow start stage, there are two mechanisms to detect it. The first mechanism detects packet loss by using timeouts (TO). The second mechanism detects packet loss upon receiving three duplicate ACKs.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 45-48 | doi: 10.11128/sne.21.sn.10047

2 Approaches for Modelling the Packet Switched Networks

There are several approaches for modeling packet - switched networks. The most accurate network models are *packet-level models* that keep track of individual packets. These models are implemented in network simulators, such as ns-2 [7]. The main drawback of packet-level model is a large computational effort required to keep track of each virtual packet in large scale simulations.

Analytical models are based on idea to model the TCP network mathematically in order to reduce complexity involved in simulations [1, 6]. These models do not consider the network dynamics. They assume that round-trip time and loss probability are constant and there are no interactions between TCP flows.

Fluid models are intended for simulation of packet networks. They overcome network scalability problem by keeping track of average quantities for relevant network parameters [5, 4]. Additionally, they use assumption that the bit rates are piecewise constant.

Hybrid models use continuous time state variables with discrete time events [3]. Hybrid simulations require significantly less computational resources than packet level simulators. However, solution of hybrid equations is still necessary in order to simulate networks.

3 Probability Model for TCP Slow-start Stage

There are two approaches for creating probability models of TCP behaviour. The source centric model assumes that packets leave the source with a certain loss probability [1]. The assumption taken by the second model is that the network generates loss probabilities for each packet [5]. Thus, the arrival process is represented by Poisson random process. In this paper we use the first approach with the further assumptions: the packet losses in two successive rounds are not correlated, packet loss occurs only in the forward direction; packet losses are independent of window size.

Model detects packet losses by triple duplicates and TOs. Based on the source-centric model [1,6], we derived the expected value of number of packets sent in the slow start phase as a function of packet loss probability p, the initial size of congestion window w_i and the number of packets acknowledged by one ACK. We further distinguish two boundary cases related to the position where packet was lost. The first case is characterized by the packet loss occurring at the beginning of the round (Figure 1). For the second case, the packet loss occurs at the end of the round (Figure 2).

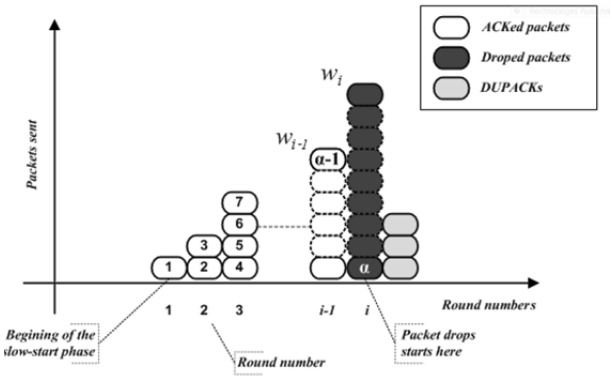


Figure 1. Slow-start phase, packet drop occurred at beginning of the round i.

Figure 1 shows the number of acknowledge packets α-1. Taking into consideration equation (2), we can write

$$ssdata_i = \alpha - 1 = w_0 \cdot \frac{\gamma^{i-1} - 1}{\gamma - 1} \tag{3}$$

where α is a random variable. If the probability of packet loss is p, probability mass function *pmf* of having k-1 acknowledged packets is $(1\text{-}p)^{k-1}p$, the expected value of the discrete random variable α is

$$E[\alpha] = \sum_{k=1}^{\infty} (1-p)^{k-1} \cdot p \cdot k = \frac{1}{p} \tag{4}$$

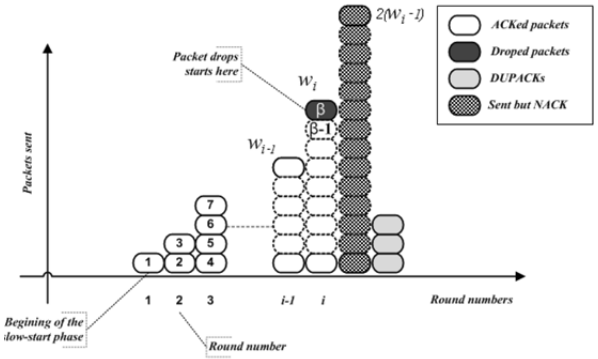


Figure 2. Slow-start phase, packet drop occurred at end of the round i.

Using the same principle for Figure 2, the number of acknowledge packets can be expressed as

$$ssdata_2 = \beta - 1 = w_0 \cdot \frac{\gamma^{i-1} - 1}{\gamma - 1} - 1 \quad (5)$$

where β is random variable, and it is clear that $\beta > \alpha$ for the same round. Expected value of β is the same as expected value of α.

The exact number of acknowledged packets sent in slow start phase lies in the interval [$ssdata_1$, $ssdata_2$]. Width of this interval is $ssdata_2$ - $ssdata_1$ = β - α = w_i-1. Hence, in general case, the number of acknowledged packet in the slow start phase equals

$$ssdata_{ACK} = \alpha + w_i - 1 + \delta_i = w_0 \cdot \frac{\gamma^i - 1}{\gamma - 1} + w_i - \kappa_i \quad (6)$$

where δ_i and κ_i are discrete random variables with values in the interval [O, w_{i-1}].

If we assume uniform distribution for these two variables, the expected values for these variables are equal and can be calculated as:

$$E[\kappa_i] = E[\delta_i] = \sum_{\delta_i=0}^{w_i-1} \delta_i \frac{1}{w_i} = \frac{w_i - 1}{2} \approx \frac{E[W] - 1}{2} \quad (7)$$

where w_i is window size in round i. Since w_i is a random variable we are going to use its expected value

Using relations (4), (6) and (7) we can deduce expected value of window size:

$$E[W] = \frac{1-p}{p} \cdot (\gamma - 1) + w_0 \quad (8)$$

The expected number of acknowledged packets in slow start phase, therefore, is:

$$E[ssdata_{ACK}] = \frac{1-p}{2 \cdot p} \cdot (\gamma + 1) + \frac{1}{2}(w_0 - 1) \quad (9)$$

Using the same approach we can calculate the total number of packets sent in slow start phase as:

$$E[ssdata] = \frac{1-p}{2 \cdot p} \cdot (3 \cdot \gamma + 1) + \frac{1}{2}(3 \cdot w_0 - 1) \quad (10)$$

4 Throughput in Slow Start Phase

Mean value of the throughput the receiver is experiencing during the slow start phase can be expressed as:

$$R = \frac{E[ssdata_{ACK}]}{E[i] \cdot t_{link} + E[ssdata_{ACK}] \cdot R_{link} + t_{ADT}} \quad (11)$$

where $E[i]$ is expected round number value in which packet drop occurred, t_{link} is link delay, R_{link} is link speed t_{ADT} average packet loss detection time.

Probability of packet loss p_R occurring in round i is

$$p_R(i) = (1-p)^{w_0 \cdot \frac{\gamma^{i-1}-1}{\gamma-1}} \cdot p \cdot \sum_{k=0}^{w_0 \cdot \gamma^{i-1}-1} (1-p)^k \quad (12)$$

Hence, expected round number value $E[i]$ in which packet drop occurs is

$$E[i] = \lim_{N \to \infty} \sum_{i=1}^{N} i \cdot p_R(i) = \lim_{N \to \infty} \sum_{i=1}^{N} (1-p)^{w_0 \cdot \frac{\gamma^{i-1}-1}{\gamma-1}} \quad (13)$$

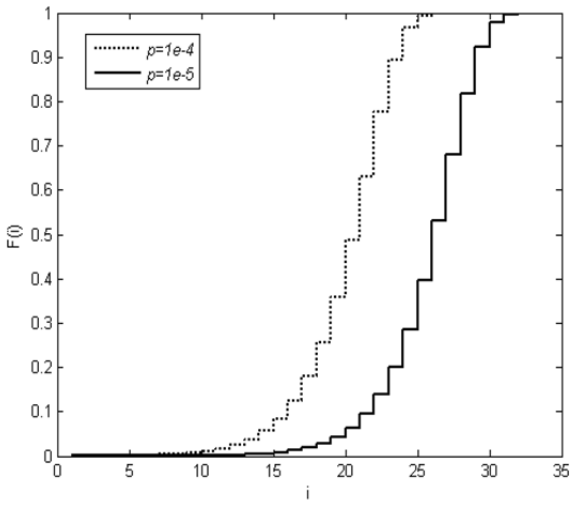


Figure 3. Cumulative distribution function for discrete random variable *i*.

For value, for which CDF shown in Figure 3 is higher than 0.9999995, the limit in equation (13) can be omitted and $E[i]$ can be calculated from the finite sum.

Average packet loss detection time interval t_{ADT} is mean time for detection of packet loss using timeouts or three duplicated ACK:

$$t_{ADT} = t_{TO} \cdot p_{TO} + t_{TD} \cdot p_{TD} \tag{14}$$

where p_{TO}, p_{TD} are probabilities of packet loss detection with timeout and three duplicated ACK, t_{TO} timeout interval and t_{TD} = RTT.

$$p_{TO} = p^{w_i} \cdot \left(1 + \frac{1-p}{p} \cdot (w_i - 1) + \frac{1}{2}\left(\frac{1-p}{p}\right)^2 \cdot (w_i - 1) \cdot (w_i - 1)\right) \tag{15}$$

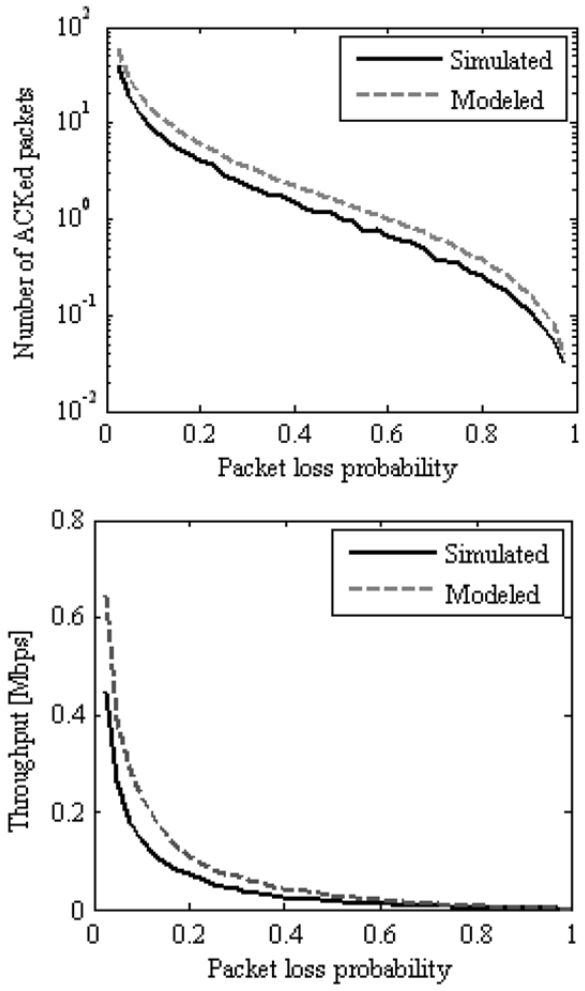


Figure 4. Simulation results: ns2 parameters: link speed 10Mbps, link delay 25ms, packet size 1024B, number of packet acknowledge by one ACK set to 1.

5 Simulation Results

Simulations were conducted in the packet level simulator ns2 [7], for a simple network topology containing only two nods. Figure 4 shows that the results obtained by probability analysis give appropriate upper boundary compared to the simulated results. Furthermore, it can be concluded that the error introduced by formula (9) grows as packet loss probability decreases. This leads to the conclusion that for lower probabilities more accurate model needs to be derived

6 Conclusion

In this paper, a probability model for the slow-start TCP stage is derived. The derived model is validated by comparing the results with the packet-level simulation tool ns-2. The test-case used for analyses was a simple to- pology consisting of two TCP nodes. Our analysis is confirmed by a large number of simulations. Similar ana- lysis should be performed for the other stages of TCP, and aggregated into a single model so that the full TCP behaviour could be analysed

References

[1] N. Cardwell, S. Savage, T. Anderson. *Modeling TCP latency*. In Proc. INFOCOM, Mar. 2000, pp. 1742-1751.

[2] V. Jacobson. *Congestion avoidance and control*. In IEEE/ACM Trans. on Networking, volume 18, 4, pages 314-329, 1988.

[3] J. Lee, S. Bohacek, J. P. Hespanha, K. Obraczka. *Modeling Data Communication Networks Using Hybrid Systems.* In IEEE!ACM Trans. on Networking, volume 15, pages 630-643, 2007.

[4] Y. Liu, L. F. Presti, V. Misra, D. Towsley. *Fluid models and solutions for large-scale ip networks*. In ACM SIGMETRICS, 2003.

[5] V. Misra, W. Gong, D. Towsley. *Stochastic Differential Equation Modeling and Analysis of TCP-Win- dowsize Behavior*. In Proceeding of PERFORMANCE, Istanbul, Turkey, 1999.

[6] V. Padhye, D. Towsley, J. Kurose. *Modeling TCP throughput: A simplc model and its empirical validation.* In SIGCOMM '98, September 1998.

[7] The ns Manual (Formerly ns Notes and Documentation) Oct.2000 [Online]. Available: http://www.isi.edu/nsnam/ns/ns-documentation.html, The VINT Project.

Accepted MATHMOD 2009
Submitted: August, 2010
Revised: February, 2011

A Soccer Simulation for Comparison of Game Strategies

Martin Bruckner[1,2 *], Nikolas Popper[2]

[1] Inst. For Analysis and Scientific Computing, Vienna University of Technology, Wiedner Haupstraße 8-10, 1040 Vienna, Austria

[2] dwh Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria; *martin.bruckner@dwh.at

Abstract. Our goal was to develop a simple simulator to show capabilities of simulation methods and to introduce a simple game for understanding how various strategies within a rather complex system of invididuals can be tested. A simple soccer game was chosen, which was motivated by the 2008 European Football Championship in Austria. With aspects form Cellular Automata and Agent Based Modelling it was possible to show, that changes in the model are very simple to implement by changing the transitional function, in our case by adding or omiting individual rules. So different strategies but also other changes in the system can be easily tested and compared.

Introduction

'The Goal' of this work is to define a Model of a soccer game. There are - not surprisingly - 22 players, including two goalkeepers, one ball and two goals. Foul play or offside will be ignored. Out balls are avoided by introduction of a 'board' where the ball is rebounded. Rules could easily be added, but were ignored to test the reactions of the system in a first stage. The model was introduced to test the modeling of different strategies of team line up and was influenced by hosting the 2008 UEFA European Football Championship in Austria and Switzerland. The model is realized as a JAVA Applet.

1 Cellular Automata Theory

As a first approach a cellular automaton was planned to implement the simulation. A cellular automaton (plural: cellular automata) is a discrete mathematical model to simulate problems for example in theoretical biology and microstructure modeling. The automaton consists of a discrete lattice of identical cells; each is in one of a finite number of states. The dimension of the grid is finite. Time is also discrete, and the state of a cell at time t is depending on the states of a finite number of cells (the neighborhood of the cell) at time $t-1$.

The neighbourhood is classed as either Moore or Von Neumann neighborhood. The Moore neighborhood comprises the eight cells surrounding the central cell on a two-dimensional area, during Von Neumann comprises the four cells orthogonally surrounding the cell. In the transition function it is precisely defined how the transition from one state to the next state takes place. The transitions of the states are made for all cells with the same transition function simultaneously.

2 Advantages/Disadvantages of the System – Introducing Agents

An advantage of the system is, that changes in the model are very simple to implement by changing the transitional function. In our case this can be done by adding or omitting individual rules. For this reason different strategies can be tested and compared. But individual rules mean, that for the various players different beaviour should be possible to implement. This is not basically possible with a cellular automaton. The virtual manager sould also be able to compare the advantages or disadvantages of 4-4-2 vs. 3-4-3 with his – limited – pool of players. By doing so it can be shown, that some strategies only work with players like Ronaldo and not with Prödl and so the strategies can be adjusted to the pool of players, as different players have different possibilities (see Section 3).

Regarding to this it appears that not a cellular automaton but some kind of mixture between cellular automton and an agent based system is appropriate. A great advantage of this simulation method is that this relatively simple system, often with only a few states, has the capability to simulate very complex systems and nevertheless it is possible to introduce different behaviour for different 'players'. As a common problem in using such systems the 'simultaneous application of the cells' is scarcely feasible.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 49-52 | doi: 10.11128/sne.21.en.10049

The state of the cells is also depending on the processing sequence of the cells.

3 The User Interface

Figure 1 shows the interface and the visualisation of the simulation, which is implemented as a Java applet. The screenshot depicts the model during the situation of a pass between German forwarders. Informations about the players are shown on the left side and can be achieved by clicking on the player. Informations like name, age, size or number help to identify the various players. More important is the possibility to change two variables for each player. The first possible change can be done on 'playing ability', the second change can be done on 'pass force'. This change can be done on every different player separately.

The user can change the overall system in the succession of computing of the different cells, shown in the lower left part of the picture.

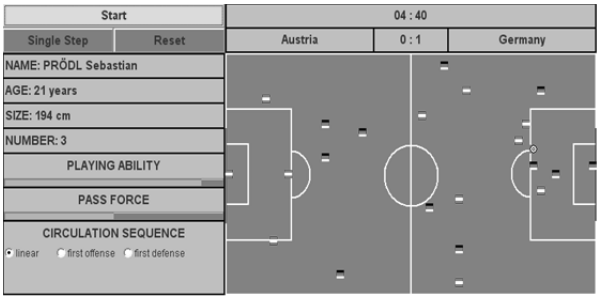


Figure 1. Screenshot – Situation of a pass.

4 Implementation

In order to realize the simulation a combination of a cellular automaton and an agent-based model was used. In contrast to other types of modeling in an agent-based model many small units (agents) have decisions or options for action. The system behavior results from the behavior of individual agents and will not be provided at the system level. Both modelling methods are 'bottom up' methods, so changes on system behaviour can be implemented easily by changing rules of the cells or agents in the system.

The cellular automaton is constructed by definition of the size of the automaton, the used neighbourhood definition, the different kind of states a cell can be in and the applied rules.

Cell space. the size of the playing field is defined by 50x29 cells, which is approximately the relation for an international playing field.

Neighborhood. Moore neighborhood (8-neighboring Cells) as we have tried to improve the influence of other players, which are near to the player holding the ball.

Set of states. empty, AUT player with ball, AUT players without ball, BRD player with ball, BRD player without ball, ball

Transitional function: Is defined in the form of rules

In order to limit the movement of the player the target field has been divided into three parts and each player is allocated by one part. The only adjustment in the APPLET is the processing sequence from the individual cells.

Linear: The cells pass through, as usual, from top left to bottom right.

First Offense. The playing field will be passed through twice in a linear process. In the first pass the cells of the attacking team will be processed, in the second step the ones of the defending team.

First defense. Analog to the first offense; the only difference is the reverse order.

The described characteristics of each player (name, age, size, number) as well as the playing ability and strength of pass will be stored in additional text files.

5 The Model

As mentioned, the basic structure of the model is 50x29 cells. At any pass, each non-empty cell is classified (team related, ball possession, attacking or defending team, goalie or field players) and, depending on the type of the cell, the matching routine, with the rule for this cell, is called. These rules evaluate the adjacent fields and allocate each field a positive natural number which indicates the unease of the player to go on this field. After evaluation of all the adjacent fields the cell with the lowest value will selected as a new site for the player. In case two or more cells have identical numbers a random process decides how the next step looks like.

In order to compensate the problems with the parallel processing of cells, the result will be stored in a second field, also 50x29 cells.

Thus, the move by more than one field of a player is avoided. This may occur when a player has moved from a line in the underlying line and will process again. Going forward, the whole game will be simulated step by step until the end. This, while also possible, shooting and kicking to a friendly player is a combination of a random process and the distance to the gate or the border area of the player. (Remember, each player is in one of the three areas.)

The direction of the shot or pass is calculated by an algorithm, in which the positions of the team mates and the opponents are taken into account. After the completion of a cycle, after all cells are processed, it is checked whether a defendable player is next to the ball carrier. If so, there will be a 'duel' in a random process, depending on the strength of the players which decides who is in possession of the ball.

In Figure 2 the situation of the German team is in possession of the ball. The player holding the ball is indicated by a small blue square. The red stop button indicates that the model is running. The buttons 'Single Step' and 'Reset' are unavailable in the running modus and therefore grayed out. The simulation can be stopped at any time to run the simulation in the single step mode or to reset the simulation run.

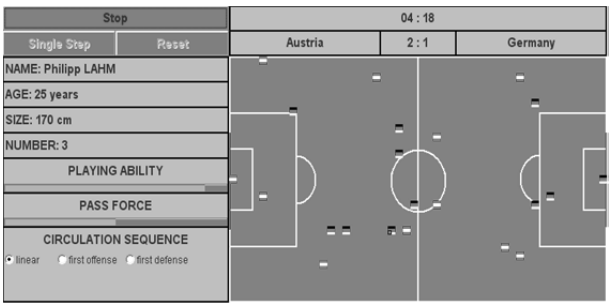


Figure 2. Screenshot – German Player be in possession of the ball.

6 Example of a Rule

The most important part of the model is the possibility to define and to integrate various rules. As mentioned at the beginning the rules can be adopted or changed more or less easily, which is one of the main advantages of the modelling technique.

In the following some examples of the mentioned 'rules' for the different types of cells (Offence with ball - Offence without ball - Offence Goal-Keeper – Defence - Defence Goal-Keeper - Player (only when Ball is free)) will be considered in more detail.

This is only one example of the whole set of rules, which have been implemented in the system. These examples for the integrated rules present how rules are described and how they influence the behaviour of the whole system.

Rule for Offence with ball

The higher the value is the less suited the cell is for the next move:

```
(STEP 1) If the cell is already occupied, it makes
no sense to consider the cell for the next step.

If (cell is not empty)
{
        Value = max.
} Else {
        Value = 0
}

(STEP 2) To constrain the range of each player on
his third each cell will be controlled.

If (cell still in my third)
{
        Value = Value + 0
} Else {
        Value = Value + max.
}

(STEP 3) To reach a balanced distribution of play-
ers in their third, it is undesirable to approach
to other teammates.

For (all teammates with distance < 7)
{
        Switch (approach to other teammates)
        {
                Case lower distance:    temp = temp
        + 6
                Case same distance:
                temp = temp + 3
                Case greater distance:  temp = temp
        + 0
        }
}

Value = Value + (temp/ teammates with distance <
7)

(STEP 4) This rule compares the strength of the
attacker with his opponent's strenth. If the at-
tacker is very strong he tends to get a more, per-
haps weaker, opponent to approach.
```

```
For (all opposing players)
{
   Switch (distance Enemy player)
   {
      Case 1 cell:
        Value = Value + (5 x force Opponents x (1
                  - force Attackers))

      Case 2 cell:
        Value = Value + (3 x force Opponents x (1
                  - force Attackers))

      Case 3 cell:
        Value = Value + (2 x force Opponents x (1
                  - force Attackers))

      Case 4 cell:
        Value = Value + (1 x force Opponents x (1
                  - force Attackers))
      Default:
        Value = 0
   }
}

(STEP 5) One of the most important rules is that
the Offence comes closer to the goal.

   Switch (come closer to goal)
   {
      Case lower distance:
        Value = Value + 0
      Case same distance:
        Value = Value + 2
      Case greater distance:
        Value = Value + 4
   }
```

The whole set of rules depending on additive system description (there are three different areas) and the individual abilities (playing ability, passing force) defines the behaviour of the system.

7 Experiment

In the experiment a number of successive games with different players'-strengths were realized and are shown in Table 1. The first 20 games were performed with the same skill and pass level (average 0.7 and 13.6) for both teams. The results of these games are shown in the table in the second column.

Afterwards 20 games with a reduced average strength (average 0.5) of the German team where performed. The strength of passport remained unchanged. These results are showen in the third column.

Game Nr.	Final outcome Equal force	Final outcome Advantage AUT
1	4:3	5:4
2	4:4	6:4
3	10:6	6:4
4	8:5	6:1
5	6:6	5:4
6	5:6	5:2
7	4:4	10:2
8	2:7	4:5
9	4:6	4:5
10	3:4	4:3
11	3:4	6:2
12	5:5	2:5
13	4:4	9:4
14	6:5	3:8
15	5:4	5:6
16	3:6	7:3
17	2:10	6:2
18	5:8	6:8
19	7:6	7:2
20	8:4	10:3
Won AUT	7	14
Won GER	8	6
Draw	5	0

Table 1. Result Table of Games.

As we can easily see the behaviour of the system (the result of the game) is influenced directly by the ability of the players. This is only one example of changing the parameters and sets of the simulation. In other experiments it can be analyzed f.e. how integrating one superstar (in teams with the same skills for the other players) would affect the results. Another point which could be analyzed is that the shown results in Table 1 are higher than we would expect in international games. This could be changed on one hand by variation of skills between defeners and other players, but also by redefining the areas of the place.

References

[1] S. Wolfram. *Cellular Automata.* 1983, http://www.stephenwolfram.com/publications/articles/ca/83-cellular/1/text.html

[2] W. Golder. *Beziehungskiste aka Party-Planer* .http://www.golder.de/psion/bkiste/main.html

[3] Wikipedia. *Zellulärer Automat* http://de.wikipedia.org/wiki/Zellulärer_Automat

[4] Wikipedia. *Agentenbasierte Modelle* http://de.wikipedia.org/wiki/ Agentenbasierte_Modelle

Submitted: August, 2010
Revised: February, 2011
Accepted: March3, 2011

A Physical Modelling Approach to ARGESIM Benchmark C11 'SCARA Robot' using Dymola

Boris Malinowsky[1], Oliver Höftberger[1], Florian Miksch[2 *]

[1] Inst. for Analysis and Scientific Computing, Vienna Univ. of Technology, Wiedner Haupstraße 8-10, 1040 Vienna
[2] dwh Simulation Services, Neustiftgasse 57-59, 1070, Vienna, Austria; * *Florian.Miksch@@drahtwarenhandlung.at*

Simulator. Dymola is an object oriented environment for simulation based on MODELICA language. Due to the object orientation, defined components can be fully reused. Equations can be stated in neutral form (DAEs), and algorithms are used to derive efficient equation code (compiled from C language) for simulation runs. This allows acausal modeling to be used without consideration of computational order. For simulation, Dymola uses the implicit DASSL solver.

Modelling: In the definition of the benchmark implicit equations describe the problem. Graphical elements in Dymola's model editor are parameterized through the component dialogs. A few equations given in the problem definition are directly written in textual form into the model. Those textual additions might be edited at a later point from the components dialog as well. The editing of the graphical forms automatically creates and adjusts the corresponding model text descriptions. Connect statements are created for the model by drawing connector lines in the editor. Libraries support modeling, here the *Modelica. Mechanics.MultiBody* Robot drives, control and collision avoidance control can be added by using standard library components.

A-Task: Mechanical Modelling Approach. The SCARA mechanic parts were put into own model component and instanced once for the robot, parameterized accordingly. The same was done for the motors and control logic. Listing 1 shows the structure of the generated textual model description, which is flattened to be called from the DASSL solver.

```
model scaraMechanics
  Inner Modelica.Mechanics.MultiBody.World…….
  Modelica.Mechanics.MultiBody.Parts. …….
  Modelica.Mechanics.MultiBody.Joints.
    ActuatedRevolute actuatedRevolute a;[...]
equation
  connect(world.frame_b, b0.frame_a) a;
  connect(b0.frame_b, actuatedRevolute.frame_a) …
end scaraMechanics;
```

Listing 1. Textual model generated by Dymola.

In Figure 1, the whole layout of the SCARA robot is shown. On the left side in the middle of the figure, the control and collision avoidance component is located. Default control is applied later in Task b, collision avoidance in Task c. The component outputs the maximum voltage to apply for the servo motors of link 1 and 2, and the destination angle position of the two robot joints 1 and 2. Two inputs are used for the current angle positions; its values are dependent on the movement of the rotational drives. The third input is the current vertical position of the tool tip (the joint distance in z-direction). On the right side in the middle of the figure, the object of the mechanical structure (mechanics component) is located.

Figure 2 shows the mechanics component, modeling the robot arm. The link and joint objects are instanced from predefined parts of the *Modelica.Mechanics. MultiBody* library. The component contains 3 inputs, 2 rotational angle connectors for the absolute rotational angle (joint 1 and 2), and 1 translational angle connector for the absolute positioning (joint 3).

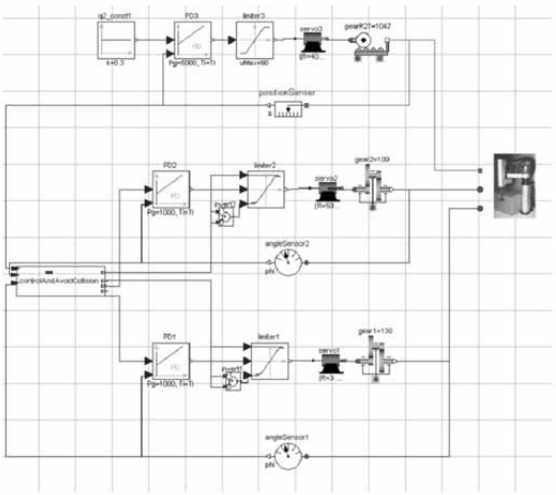


Figure 1. Layout of the model in the graphical editor of Dymola.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 53-54 | doi: 10.11128/sne.21.bn11.10050

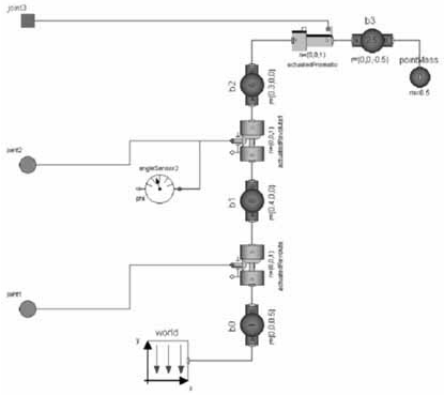


Figure 2. The modeled SCARA mechanic component using Modelica MultiBody parts.

B-Task: Simulation of a point-to-point movement. The robot contains 3 servo motors to drive the joints that are controlled by a single axis PD-controller each. The PD-controller is a modified standard component from the MODELICA library. A limiter block limits the voltage to a positive and negative maximum while the motor limits the maximum armature current in its electric/mechanic transformer (EMF, component limited EMF) using an if-condition. Figure 3 shows simulation results, start position is 0 for all joints (q1, q2, q3 equals 0), end position is 2 rad for q1 and q2, and 0.3m for q3.

The standard DAE solver DASSL worked efficiently. Dymola offers also other DAE solvers like LSODAR, a multi-step solver for stiff and non-stiff DAEs with root finder, and DOPRI45, a Runge-Kutta method with local extrapolation. While DOPRI45 was significantly slower thatn DASSL (factor 2.8), the LSODAR solver was slightly faster – factor 0.8. It must be expected, that LSODAR is faster if state events as the possible collision (Task C) must be recognized, but also without state events the algorithm has its benefits.

C-Task: Collision avoidance feature. For detecting a collision of the robot tool tip with an obstacle, a collision detector component is modeled. It controls maximum voltage and angular target positions. Collision avoidance can be enabled (for this task) or disabled (for Task b). A graphical structure setup in the component is used to calculate the tool tip position xtip (green graph in Figure 3). A logical network with boolean expressions checks for the critical position.

In case of moving into a critical position (possible collision in the future), the a boolean condition becomes true, it freezes tool tip position and stops angular movement by setting the temporary target position for motor 1 and 2 to current position.

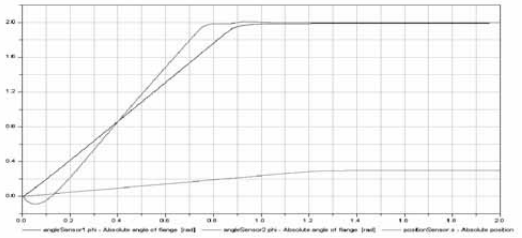


Figure 3. Joint positions without collision.

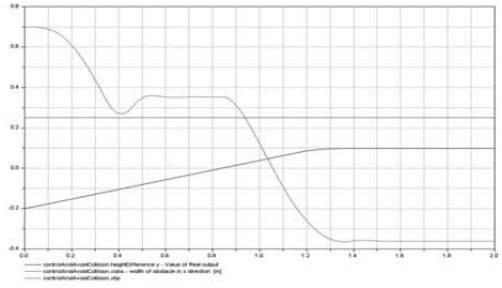


Figure 4. Tool-tip position and obstacle (in x-direction), and difference over time with collision avoidance.

It also switches output signal from regular maximum voltage to maximum emergency voltage. If the tool tip raised above obstacle height, targets and voltages are set back to regular. These actions are synchronized with the DAE solver (state event finding) by means of IF or WHILE constructs in DYMOLA.

The simulation results are shown in Figure 4, with simulation time 2 seconds, and distances and lengths drawn in meters.

Dymola also provides the functionality to animate and render graphical objects representing physical components during simulation runs (Figure 5).

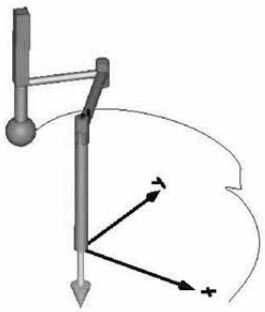


Figure 6. Animation view with tool-tip track (blue line) after collision avoiding

Resumé. Dymola provides appropriate predefined components in the MODELICA mechanics library and the standard library to build up the SACRA Robot and to implement the requested behavior as well as collision avoidance feature. Making use of plotting and rendering functionality offers the possibility to visualize the simulation and results very well.

Received: December 2009
Revised: July 20, 2010
Accepted: November 30, 2010

A Scipt Language – supported Approach to ARGESIM Benchmark C14 'Supply Chain' in Enterprise Dynamics

Philipp Gutwenger, Do Duc Hoa, Wolfgang Puffitsch, Shabnam Tauböck *

[1] Inst. for Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, *Austria; shabnam.tauboeck@tuwien.ac.at*

Simulator. Enterprise Dynamics is a simulation environment for discrete systems. It employs an event-oriented approach and supports 2D and 3D animations. Models consist of 'atoms' which can be configured with the 4D Script language. Atoms can inherit attributes from 'mother' atoms and can thus form a hierarchy. Communication between atoms takes place via channels.

Modelling: Figure 1 shows how a *factory* (including delays to distributors) was modeled. Figure 2 shows a *distributor*. The complete model consists of 4 factories, 4 distributors and a group of wholesalers.

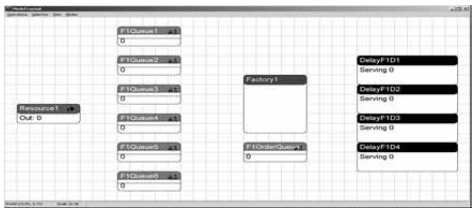


Figure 1. Factory Modelling.

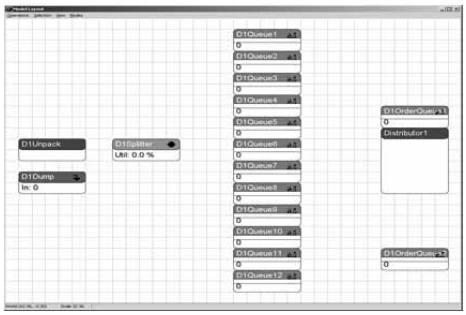


Figure 2. Distributor Modelling.

Factory. Products are spawned in a generic source and are distributed to a random product queue of the factory. An order is a special type of atom, containing information about the type and number of the ordered product and the distributor ordering it.

If such an order enters through the respective queue, the factory checks the content of the queue for that type of product and delivers the products (packed into the order) through the appropriate delay element or returns the order to the distributor if unavailable.

The following code (written in 4D script) is located in the *onEntered handler* and builds the core of the factory. Checking for available products and possibly postponing them is done here. Routing to the appropriate distributor is determined by the attribute *DistributorNumber* of the order automatically. The function *moveatom* was for sending the atoms.

```
if ((att([ProductCount],i) >
    content(atomByName(concat([F1Queue],
    string(att([ProductNumber],i))),
    Model))),
 do ( {* postpone order *}
  case (att([DistributorNumber],i),
   moveatom(i,atombyName(
     [D1OrderQueue2], Model)),
   moveatom(i,atombyName(
     [D2OrderQueue2], Model)),
   moveatom(i,atombyName(
     [D3OrderQueue2], Model)),
   moveatom(i,atombyName(
     [D4OrderQueue2], Model))
  ),
  return
 ),
 do ( {* set table for packing *}
  cell(att([ProductNumber],i)+1-6,
       att([ProductNumber],i),c) :=
    att([ProductCount],i),
  finishquant :=
    +(content(i),
      sum(nrows(c),cell(count,curcolref,c))),
  setloc(0,0,zsize(c),i)
```

Distributor. In the distributors, orders are unpacked and products are stored in the appropriate queue. Upon request from the group of wholesalers, the distributor checks its queues and delivers the product if available. Every 24 hours, a special order is sent to the distributors. This order causes the distributor to pass its orders to the factories along the implemented strategy.

The 4D-language code describing the handling of the atoms in distributor follows same principles as the code for actions in the factories. Attributes record and control unfulfilled requests; two different attributes are needed in order to implement the different control strategies properly.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 55-56 | doi: 10.11128/sne.21.bn14.10051

Group of wholesalers. The group of wholesalers is a source which produces orders for random products and sends it to a random distributor.

Orders. All information is passed inside the orders, which are derived from the standard Container atom with additional attributes for holding the appropriate information. This has the advantage, that for the factories and the distributors, standard atoms could be used too (albeit extended to some degree).

A-Task: Simple Order Strategy. As Figure 3 shows, the stock of Distributor 1 grows steadily (backwards history timescale) with the simple order strategy. This is caused by the fact that on average, less than 2 products are ordered each day from the group of wholesalers.

Table 1 shows the number of delivered products *D*, the total costs *T* and the costs per product *R*. 100 runs with the experimentation tool in Enterprise Dynamics were made to compute the statistics.

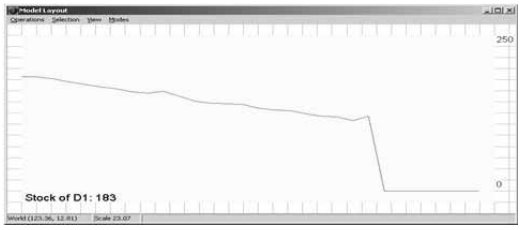


Figure **3.** Stock with simple order strategy (backwards time scale)**.**

	Average	Deviation	Minimum	Maximum
D	225,50	13,79	194,00	256,00
T	€ 34.759	€ 1.496	€ 30.748	€ 37.766
R	€ 154,41	€ 4,69	€ 138,72	€ 170,56

Table 1. Simple order strategy costs**.**

B-Task: On Demand Order Strategy. Figure 4 shows that the on demand strategy yields a constant stock for Distributor 1 – the expected result, if the same number of products is ordered and sold. The costs for the on demand order strategy are only slightly lower than for the simple order strategy, as seen in Table 2, which might be explained with the lower costs for stock keeping.

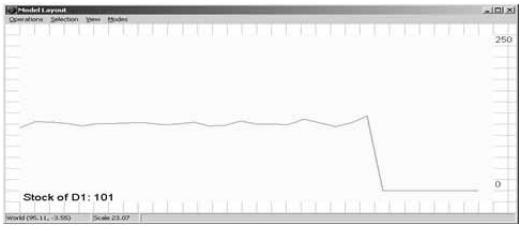


Figure 4. Stock with on demand order strategy, (backwards time scale)**.**

	Average	Deviation	Minimum	Maximum
D	228,24	13,36	195,00	257,00
T	€ 33.653	€ 1.501	€ 29.302	€ 36.790
R	€ 147,66	€ 5,65	€ 135,65	€ 161,60

Table 2. On demand order strategy costs.

C-Task: Minimal Supply Time Strategy. For this strategy it was necessary to extend the model, because the distributors can order from any factory with this strategy, and they have to check the stock of a factory before they send the orders.

The stock stays constant with this strategy, as with the on demand order strategy, as seen in Figure 5. Due to the lower cost for transportation, the costs with this strategy are significantly lower than with the other strategies, as Table 3 shows.

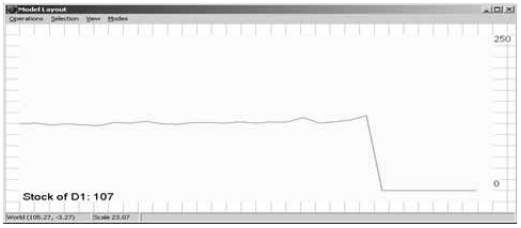


Figure 5. Stock with minimal supply time strategy, (backwards time scale)**.**

	Average	Deviation	Minimum	Maximum
D	229,17	12,88	201,00	262,00
T	€ 27.237	€ 1.201	€ 23.978	€ 30.362
R	€ 119,00	€ 4,36	€ 109,16	€ 132,36

Table 3. Minimal delay order strategy costs**.**

Summary. Enterprise Dynamics is an object-oriented simulation system well suited for process flow models and similar applications. Supply chain models require additionally a flow of orders, usually in the opposite direction than the classical physical entity flow. Because of its programming capabilities with 4D script language, various approaches for control of the process flow can be used. The approach used in this solution makes use of order objects which queue in the various object stations to be handled different.

Received: November 2009
Revised: July 1, 2010
Accepted: November 30, 2010

A Quantization-based ODE Approximation and HPP-LGCA Approach to ARGESIM Benchmark C17 'SIR-type Epidemic' in a DEVS Environment based on MATLAB

Tobias Schwatinski*, Thorsten Pawletta

Wismar University of Applied Sciences, Dept. of Mechanical and Process Engineering,
Res. Group Computational Engineering and Automation (CEA),
PB 1210, 23952 Wismar, Germany; *tobias.schwatinski@hs-wismar.de*

Simulator. MATLAB is a widely used programming environment for numerical computations extended by different toolboxes. Using MATLAB's object oriented programming features a Parallel-DEVS simulation toolbox (PDEVStbx.) for discrete event oriented modeling and simulation is implemented [5]. The toolbox is based on the Discrete Event System Specification (DEVS) formalism by Zeigler and Chow [1, 2]. To support also continuous and hybrid system modeling the PDEVStbx is extended by specific components according to the Quantized-State-Systems (QSS) approach introduced in [3].

Modelling. The DEVS formalism provides a comprehensive framework for modeling and simulation based on systems theory. There are two basic system types used for modeling called atomic DEVS and coupled DEVS. The dynamic behaviour is specified in atomic DEVS. Both basic system types can be composed in coupled DEVS to define systems in a modular, hierarchical manner. The Classic DEVS formalism has been advanced several times. One extension is the Parallel DEVS formalism [1, 2]. A Parallel atomic DEVS is specified as follows:

$$\text{PDEVS} = \{X, Y, S, \delta_{ext}, \delta_{int}, \delta_{conf}, \lambda, ta\} \quad (1)$$

where X is the set of input values, S the set of sequential states, Y is the set of output values, δ_{ext} is the external transition function, δ_{int} is the internal transition function, δ_{conf} is the confluent function, λ is the output function and ta is the time advance function.

According to Figure 1 every atomic DEVS has an internal state $s \in S$. By the mean of the time advance function ta(s) the time step till the next internal event is calculated on the basis of the internal state s.

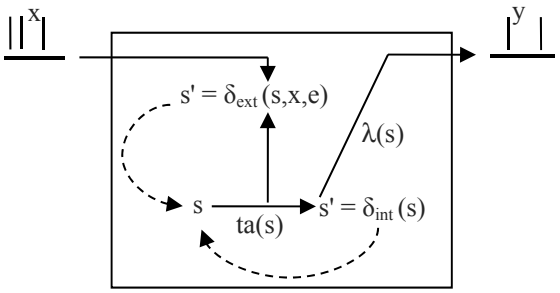


Figure 1. Dynamic behaviour of an atomic DEVS [1].

After having expired this time period the output function $\lambda(s)$ is carried out and as a result all output events $y \in Y$ at the output port are calculated on the basis of the internal state s. In the following the internal transition function $\delta_{int}(s)$ is carried out, which calculates the next state $s' \in S$ on the basis of the current state s. If external events $x \in X$ occur at the input port the external transition function $\delta_{ext}(s,x,e)$ will be executed. This function calculates the next state $s' \in S$ on the basis of the current state s and the elapsed time e since the last event and the current external events $x \in X$.

At the end the time advance for the next internal event is calculated by the time advance function ta(s). If external and internal events occur simultaneously the confluent function $\delta_{conf}(s,x,e)$ is used instead of the internal or external transition function for calculating the next state s'. A coupled DEVS is specified as follows:

$$\text{DEVN} = \{X, Y, D, \{M_d \mid d \in D\}, Z_{i,d}\} \quad (2)$$

where X is the set of input values, Y is the set of output values, D is the set of the component names, M_d is the DEVS system of component name $d \in D$ and $Z_{i,d}$ defines the coupling relations of a coupled DEVS.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 21(1), 2011, 57-60 | doi: 10.11128/sne.21.bn17.10053

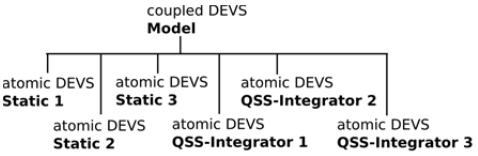
A modular, hierarchical DEVS model is executed by assigning a simulator to each atomic DEVS system and a coordinator to each coupled DEVS system. They control the evaluation of the system functions and the analysis of the coupling relations of the DEVS systems assigned to them. Simulators and coordinators communicate by messages. A special root coordinator manages the current simulation time, starts and stops a simulation run.

Modeling with QSS. The Quantized-State-Systems (QSS) method [3] deals with approximating differential equations, replacing the time discretization by a quantization of state variables. Thus a continuous system $\dot{\vec{x}} = f(\vec{x}(t), \vec{u}(t))$ is transformed to $\dot{\vec{x}} = f(\vec{q}(t), \vec{u}(t))$ where $\vec{x}(t)$ and $\vec{q}(t)$ are component-wise related by hysteretic quantization functions. During simulation the smallest time step h is calculated, such that $x(t+h) = x(t) \pm \Delta Q$ where ΔQ is the quantization level, t is the current simulation time and x is any quantized state value. It is important to point out that such an integration algorithm supports an adaptive step size control. QSS has many advantages in comparison with ordinary time-based ODE solvers. According to [3], the QSS method is always stable without using implicit formulae at all.

To solve the ODEs of benchmark C17 with the QSS method in a Parallel-DEVS simulation environment it is necessary to specify two basic atomic DEVS models according to definition (1) called Integrator and Static. Both basic atomic systems have a generalized DEVS specification [3]. Each Integrator handles and quantizes a specific continuous state quantity of an ODE system and each Static system calculates the derivative for one Integrator. Thus the structure of a QSS model looks quite similar to a control based block model like e.g. in MATLAB/Simulink.

Figure 2 shows the model structure of the example C17 using the QSS approach and the PDEVStbx. The atomic system types Static and Integrator are implemented as basic classes in the PDEVStbx. Each atomic DEVS component is incarnated from its basic class definition. Static objects have to be parameterized with a specific derivative formulae and Integrator objects have to be parameterized with the quantization level ΔQ and possibly with a hysteresis width ε.

Modeling with HPP-LGCA-cells using PDEVS. The behaviour of a single HPP lattice gas cellular automata cell can also be modelled based upon the common atomic PDEVS specification. In addition such atomic PDEVS components are specified in a coupled DEVS in accordance to the required domain.



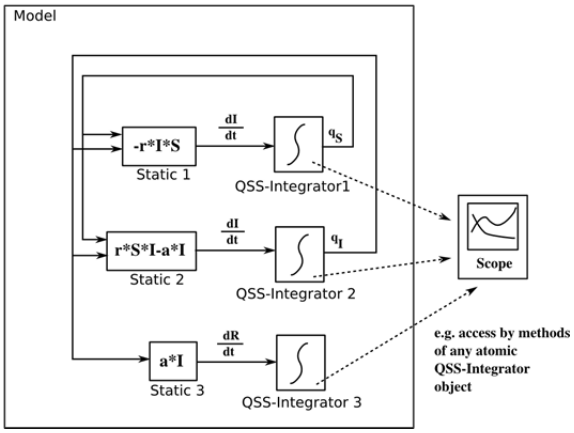


Figure 2. Composition tree and structure of the C17 model.

Therefore a HPP-LGCA atomic PDEVS component can be specified as follows:

- four input and output ports to bordering atomic DEVS used for the exchange of particles

X = {(p,v) | p∈InPorts, v∈Xp},
where: InPorts ∈ {I1, I2, I3, I4}
$X_{I1} = X_{I2} = X_{I3} = X_{I4} \in \{1, 2, 3\}$

Y = {(p,v) | p∈OutPorts, v∈Yp},
where: OutPorts ∈ {O1, O2, O3, O4}
$Y_{O1} = Y_{O2} = Y_{O3} = Y_{O4} \in \{1, 2, 3\}$

- two state variables

S = (sigma, storage) = ({1, ∞} x [Vector 1x4]),

where *sigma* is the time till the occurrence of the next internal event and *storage* is a vector which four elements present the occupancy of the HPP-LGCA cell with particles. In this context 0 indicates an empty field without any particle (1, 2 and 3 indicates the occupancy by a susceptible, infected or recovered particle).

- the internal transition function

  $\mathbf{s' = \delta_{int}(s),}$

where the new state s' is set to *storage* = [0, 0, 0, 0] and *sigma* = ∞.

- the external transition function

  $\mathbf{s' = \delta_{ext}(s, x, e),}$

  where at first *storage* is set to [0, 0, 0, 0] and afterwards the new state s' is composed by *storage* that is filled up with any incoming particles from the input x and *sigma* that is set to 1.

- the time advance function

  $\mathbf{\Delta t = ta\ (s),}$

  where the time period Δt till the next internal event is calculated from the current state s. In detail Δt is set to *sigma*.

- the output function

  $\mathbf{y = \lambda(s),}$

  where propagation and collision rules are implemented. As a result *y* contains the set of leaving particles distributed over the output ports.

- the confluent function

$\mathbf{\delta_{conf}(s, x, e) = \delta_{ext}(s, x, e),}$

which simply calls the external transition function δ_{ext}.

By the mean of HPP-LGCA-cells implemented as atomic PDEVS and composed in a coupled DEVS it is possible to analyze different vaccination strategies regarding *space* and *time* which are discussed in Task b.

A-Task: QSS, CA and ODE Simulation. To compare the results the given system of ODEs has been solved with MATLAB's ODE45 solver using the initial values and parameters shown in table 1.

S(t=0) = S0	16000
I(t=0) = I0	100
R(t=0) = R0	0
Infection rate r	0.6/10000
Recovery rate a	0.2

Table 1. Initial values and parameters for Task a.

Beside the simple QSS-method there exist different extensions which improve the abilities and computation speed. One improvement of QSS is to change hysteretic quantization to first order quantization.

In the ordinary QSS approach the output trajectory of an Integrator has to be constant within the quantization level. In contrast to QSS, the advanced QSS2 method allows a linear growth of the output trajectory within the quantization level [3]. In analogy to the QSS Integrator and Static system types QSS2 specific Integrator and Static system types are implemented in the PDEVStbx. Figures 3 and 4 show the simulation results for both QSS-methods, the PDEVS implementation of a HPP-LGCA approach and the standard MATLAB ODE45 solver. It is important to point out that the QSS solutions are computed using ordinary **PDEVS** simulators and coordinators.

The simulation results based on the QSS and QSS2 method are of similar qualitative and quantitative nature in comparison with ODE45. However, the QSS solution (b) in Figure 3 with a quantization level of ΔQ=10 causes higher numerical costs than the QSS2 solution (d) in Figure 4 with a smaller quantization level of ΔQ=5. The reason for this is the modified quantization function within the QSS2 method. The simulation result from the HPP-LGCA implementation differs from the ODE45 solution to the effect that the epidemic spreads slower. However, the qualitative nature of the solution is similar to ODE45.

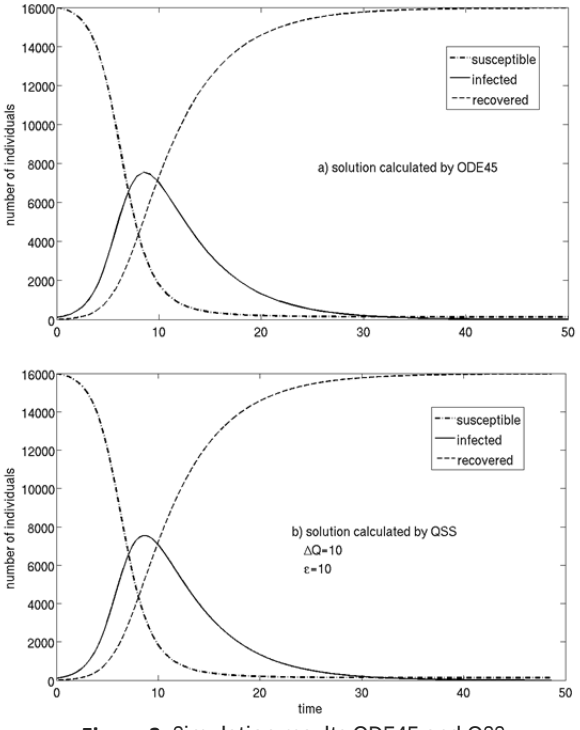


Figure 3. Simulation results ODE45 and QSS.

B-Task: Vaccination Strategies in CAs. The distribution of different particles over the cellular automata can be easily realized by setting the *storage* vector of every atomic HPP-LGCA cell during the generation of a coupled DEVS model. Figure 5 shows exemplarily simulation results of three possible vaccination strategies.

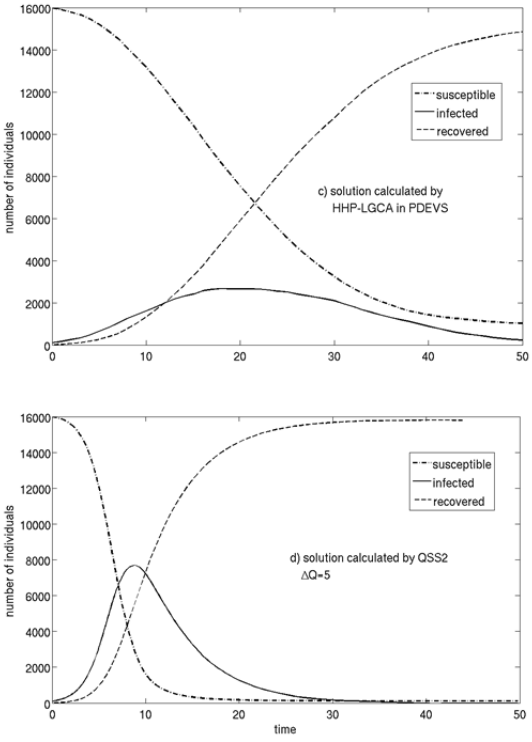


Figure 4. Simulation results HPP-LGCA and QSS2.

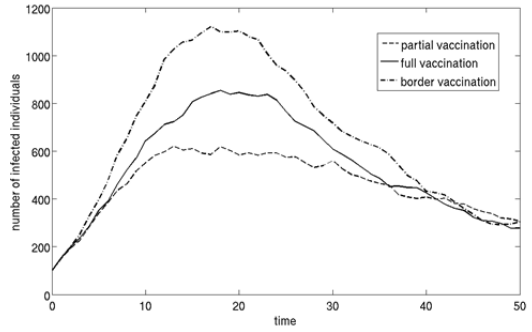


Figure 5. Simulation results of vaccination strategies.

At every strategy all infected particles are randomly distributed at the left half of the domain. Following Figure 5 the maximum amount of infected particles reaches the highest value for the border vaccination strategy where vaccinated particles are randomly distributed at the border of the whole domain. It is followed by the full vaccination strategy where vaccinated particles are randomly distributed over the whole domain.

The smallest amount of infected particles is reached for the partial vaccination strategy where vaccinated particles are randomly distributed together with infected particles at the left half of the domain.

Résumé. The benchmark shows on the one hand that both QSS approaches deliver the same results as MATLAB's ODE45 solver. Moreover the QSS approach offers additional methods for solving stiff-systems. These methods use future derivatives similar to implicit time integration methods. Because the QSS approach can be seamless integrated in a discrete event simulation environment like the PDEVStbx., its application is particularly useful in hybrid system simulation.

On the other hand the benchmark shows that even CAs can be modelled on the basis of the common PDEVS specification. Hence, it is possible to simulate models regarding to time and space. However, the generation of the domain is difficult and the simulation quite slow. The reason for this is the fact that PDEVS coordinator algorithms focus only on the coupling relations between individual atomic DEVS and not on their position on a domain or their neighbourhood. This leads to ineffectiveness and difficulties if the model is composed of many thousand square lattices which are only coupled with its four nearest neighbours. These drawbacks can be overcome by using the Cell-DEVS formalism [4] which allows the definition of cell spaces based on the DEVS formalism and CA models.

References

[1] B. P. Zeigler, H. Prähofer, T. G. Kim. *Theory of Modeling and Simulation,* Second Edition, Academic Press 2000.

[2] A. C.-H. Chow. *Parallel DEVS: A parallel, hierarchical, modular modelling formalism and its distributed simulator* , Transactions of the Soc. For Modeling and Simulation Int., Vol. 13 No 2, June 1996, 55-67.

[3] F. E. Cellier, E. Kofman. *Continuous System Simulation*, Springer 2006.

[4] G. A. Wainer. *Discrete-event modeling and simulation: a practitioner's approach*, CRC Press 2009.

[5] T. Schwatinski, T. Pawletta. *An Advanced Simulation Approach for Parallel DEVS with Ports*. In: Proc. of Spring Simulation Multiconference 2010, Orlando/Florida, USA, April 11-15, 2010.

Received: January 2011
Revised: March 6, 2011
Accepted: March 30, 2011

SNE Simulation News

EUROSIM Data and Quick Info

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013
www.eurosim2013.info

Contents

Simulation Notes Europe SNE is the official membership journal of EUROSIM and distributed / available to members of the EUROSIM Societies as part of the membership benefits. **SNE** is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With **Online SNE** the publisher **ARGESIM** follows the **Open Access** strategy for basic **SNE** contributions. Since 2011 **Online SNE** contributions are identified by DOI 10.11128/sne.xx.nnnnn. for better web availability and indexing.
Print SNE, high-resolution **Online SNE**, and additional SNE contributions are available for subscription via membership in a **EUROSIM** society.

This *EUROSIM Data & Quick Info* compiles data from EUROSIM societies and groups: addresses, weblinks, officers of societies with function and email, to be published regularly in **SNE** issues.

SNE Editorial Office /ARGESIM

→ *www.sne-journal.org, www.eurosim.info*
office@sne-journal.org
Felix Breitenecker, *eic@sne-journal.orgt*
Anna Mathe, *Anna.Mathe@tuwien.ac.at, office@sne-journal.org*
Nikolas Popper, *Niki.Popper@drahtwarenhandlung.at*

If you have any information, announcement, etc. you want to see published, please contact a member of the editorial board in your socitey or the editorial office.

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 thirteen full members and one observer member:

ASIM	Arbeitsgemeinschaft Simulation *Austria, Germany, Switzerland*
CEA-SMSG	Spanish Modelling and Simulation Group *Spain*
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*
LSS	Latvian Simulation Society *Latvia*
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*
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	Khalid Al.Begain *kbegain@glam.ac.uk*
Past president	Mikuláš Alexík (CSSS), *alexik@frtk.fri.utc.sk*
Secretary	Borut Zupančič (SLOSIM) *borut.zupancic@fe.uni-lj.si*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
SNE Repres.	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

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

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

→ *www.sne-journal.org, menu SNE*
✉ *office@sne-journal.org*

EuroSim Congress. EUROSIM is running the triennial conference series EUROSIM Congress. The congress is organised by one of the EUROSIM societies. EUROSIM 2013 will be organised by UKSIM in Cardiff, Wales, UK, September 10-13, 2013.

Chairs / Team EUROSIM 2010
Khalid Al.Begain, kbegain@glam.ac.uk
Richard Zobel, r.zobel@ntlworld.com
David Al-Dabass, david.al-dabass@ntu.ac.uk
Alessandra Orsoni, *A.Orsoni@kingston.ac.uk*
Richard Cant, *richard.cant@ntu.ac.uk*

→ *www.eurosim2013.com*

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 Officers

President	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*
Vice presidents	Sigrid Wenzel, *s.wenzel@uni-kassel.de*
	T. Pawletta, *pawel@mb.hs-wismar.de*
Secretary	Anna Mathe, *anna.mathe@tuwien.ac.at*
Treasurer	I. 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, *felix.breitenecker@tuwien.ac.at*
Universities / Research Inst.	S. Wenzel, *s.wenzel@uni-kassel.de*
	W. Wiechert, *W.Wiechert@fz-juelich.de*
	J. Haase, *Joachim.Haase@eas.iis.fraunhofer.de*
	Katharina Nöh, *k.noeh@fz-juelich.de*
Industry	S. Wenzel, *s.wenzel@uni-kassel.de*
	K. Panreck, *Klaus.Panreck@hella.com*
Conferences	Klaus Panreck *Klaus.Panreck@hella.com*
	A. Gnauck, *albrecht.gnauck@tu-cottbus.de*
Publications	Th. Pawletta, *pawel@mb.hs-wismar.de*
	Christina Deatcu, *christina.deatcu@hs-wismar.de*
	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
Repr. EUROSIM	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
	N. Popper, *niki.popper@drahtwarenhandlung.at*
Education / Teaching	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	N. Popper, *niki.popper@drahtwarenhandlung.at*
	Katharina Nöh, *k.noeh@fz-juelich.de*
International Affairs	H. Szczerbicka, *hsz@sim.uni-hannover.de*
	O. Rose, *Oliver.Rose@tu-dresden.de*
Editorial Board SNE	T. Pawletta, *pawel@mb.hs-wismar.de*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
Web EUROSIM	Anna Mathe, *anna.mathe@tuwien.ac.at*

Last data update March 2011

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

ASIM Working Groups

GMMS	Methods in Modelling and Simulation Th. Pawletta, *pawel@mb.hs-wismar.de*
SUG	Simulation in Environmental Systems J. 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*
EDU	Simulation in Education/Education in Simulation N. Popper, *niki.popper@drahtwarenhandlung.at*
SBW SVS STA	Special Working Groups on Simulation in OR , Traffic Systems, Standardisation, Modelica, Numerical Mathematics, etc.

Last data update December 2011

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*

vdusak@foi.hr

CROSSIM / Vesna Dušak
Faculty of Organization and
Informatics Varaždin, University of Zagreb
Pavlinska 2, HR-42000 Varaždin, Croatia

CROSSIM Officers

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 Božikov, *jbozikov@snz.hr*
Edit. Board SNE	Vesna Dušak, *vdusak@foi.hr*
Web EUROSIM	Jadranka Bozikov, *jbozikov@snz.hr*

Last data update March 2011

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*

snorek@fel.cvut.cz

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

CSSS Officers	
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

Last data update December 2009

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*

y.hamam@esiee.fr

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

FRANCOSIM Officers	
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*

Last data update April 2006

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 with affiliated societies.

→ `www.eurosim.info`

a.w.heemink@its.tudelft.nl

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

DBSS Officers	
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*

Last data update April 2006

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*

javor@eik.bme.hu

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

HSS Officers	
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*

Last data update March 2008

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 257 members.

→ *www.ptsk.man.bialystok.pl*

leon@ibib.waw.pl

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

PSCS Officers	
President	Leon Bobrowski, *leon@ibib.waw.pl*
Vice president	Andrzej Grzyb, Tadeusz Nowicki
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*
PSCS Board Members	R. Bogacz , Z. Strzyzakowski Andrzej Tylikowski

Last data update March 2009

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*

Mario.savastano@uniina.at

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

ISCS Officers	
President	M. Savastano, *mario.savastano@unina.it*
Vice president	F. Maceri, *Franco.Maceri@uniroma2.it*
Repr. EUROSIM	F. Maceri, *Franco.Maceri@uniroma2.it*
Edit. Board SNE	M. Savastano, *mario.savastano@unina.it*

Last data update April 2005

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*

esko.juuso@oulu.fi

✉ SIMS / SIMS/Esko Juuso, Department of Process and Environmental Engineering, 90014 Univ.Oulu, Finland

SIMS Officers	
President	Esko Juuso, *esko.juuso@oulu.fi*
Treasurer	Vadim Engelson, *vadim.engelson@mathcore.com*
Repr. EUROSIM	Esko Juuso, *esko.juuso@oulu.fi*
Edit. Board SNE	Esko Juuso, *esko.juuso@oulu.fi*
Web EUROSIM	Vadim Engelson

Last data update December March 2011

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.

→ www.slosim.si

slosim@fe.uni-lj.si

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

SLOSIM Officers	
President	B. Zupančič, *borut.zupancic@fe.uni-lj.si*
Vice president	Leon Žlajpah, *leon.zlajpah@ijs.si*
Secretary	Vito Logar, *vito.logar@fe.uni-lj.si*
Treasurer	Milan Simčič, *milan.simcic@fe.uni-lj.si*
Repr. EUROSIM	B. Zupančič, *borut.zupancic@fe.uni-lj.si*
Deputy	Rihard Karba, *rihard.karba@fe.uni-lj.si*
Edit. Board SNE	Rihard Karba, *rihard.karba@fe.uni-lj.si*
Web EUROSIM	Vito Logar, *vito.logar@fe.uni-lj.si*

Last data update March 2011

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*

david.al-dabass@ntu.ac.uk

✉ UKSIM / Prof. David Al-Dabass
Computing & Informatics,
Nottingham Trent University
Clifton lane, Nottingham, NG11 8NS
United Kingdom

UKSIM Officers	
President	David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Secretary	A. 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*
Repr. EUROSIM	Richard Zobel, *r.zobel@ntlworld.com*
Edit. Board SNE	Richard Zobel, *r.zobel@ntlworld.com*

Last data update March 2009 (partially)

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*

→ *simulacion@cea-ifac.es*

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

CAE - SMSG Officers	
President	María J. la Fuente, *maria@autom.uva.es*
Repr. EUROSIM	Emilio Jiminez, *emilio.jiminez@unirioja.es*
Edit. Board SNE	Emilio Jiminez, *emilio.jiminez@unirioja.es*

Last data update March 2009

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/*

merkur@itl.rtu.lv

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

LSS Officers	
President	Yuri Merkuryev, *merkur@itl.rtu.lv*
Repr. EUROSIM	Yuri Merkuryev, *merkur@itl.rtu.lv*
Edit. Board SNE	Yuri Merkuryev, *merkur@itl.rtu.lv*

Last data update December 2008

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.

→ *www.ici.ro/romsim/*

sflorin@ici.ro

✉ ROMSIM / Florin Stanciulescu,
National Institute for Research in Informatics, Averescu
Av. 8 – 10, 71316 Bucharest, Romania

ROMSIM Officers	
President	Florin Stanciulescu, *sflorin@ici.ro*
Vice president	Florin Hartescu, *flory@ici.ro* Marius Radulescu, *mradulescu@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*

Last data update March 2009

EUROSIM Federation of European Simulation Societies

EUROSIM 2010 CONGRESS. The Congress was organized by the Czech and Slovak Simulation Society CSSS in cooperation with the Czech Technical University in Prague. More than 235 papers from almost 41 countries were submitted during the rewieving process.

All the papers were revised by two members of the International Programme Committee (IPC). Altogether 185 contributions were accepted by the IPC and then included in the final programme. 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.

Five Invited Plenary Talks and one scientific performance were presented by outstanding speakers and presenters. All of them are well known European experts in the field of modelling and simulation:

- Human Behaviour Modelling as a Challenge for Future Simulation R&D: Methodologies and Case Studies; Agostino G. Bruzzone, University of Genoa (IT)
- Agent-Based Simulations of Transportation Terminals; Antonin Kavicka, University of Pardubice (CZ) , Norbert Adamko, University of Žilina (SK)
- Physical Modelling: Some Experiences from Education and Applications; Borut Zupančič, Univ. of Ljubljana (SL)
- Modelica; Peter Fritzson, University of Linkopinkg (SE)
- Modelling and Simulation in Health Technology Assessment; Felix Breitenecker, Vienna University of Technology (AT)
- Behave Emotions like Transfer Functions? - The Laura-Petrarca Case; LAURA Group, Vienna Univ. of Technology (AT)

Vice Mayor of Prague and Miroslav Snorek address the EUROSIM 2010 participants

Among the participants there were big delegations from **EUROSIM** member societies, especially from CSS as hosting society, from ASIM as neighbouring countries Germany and Austria, and from SLOSIM, the Slovenian Simulation Society.

Delegation from SLOSIM

On occasion of the **EUROSIM** Board Meeting at this congress, also a new constitution of the **EUROSIM** Board took place. The officers of the new board are shown in the following table – the change of board was celebrated on occasion of congress dinner at Prague castle, where M. Alexik forwarded the **EUROSIM** umbrella to K. Al Begain from UKSIM, the organiser of **EUROSIM 2012 CONGRESS.**

President	Khalid Al.Begain *kbegain@glam.ac.uk*
Past president	Mikuláš Alexík (CSSS), *alexik@frtk.fri.utc.sk*
Secretary	Borut Zupančič (SLOSIM) *borut.zupancic@fe.uni-lj.si*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
SNE Repres.	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

Handing over of EUROSIM Umbrella: K. Juslin (SIMS), B. Zupancic (SLOSIM), M. Alexik (Past President), F. Breitenecker (ASIM), A. Orsoni (UK), M. Snorek (CSSS), K. Al Begain (New President)

SNE – Simulation Notes Europe.

For 2011 changes in publication of **SNE** are planned. First, the name will change from **Simulation News Europe** to **Simulation Notes Europe**, and publication will be in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271).

With **Online SNE** the publishers will follow the **Open Access** strategy for basic **SNE** contributions, a demand from a majority of EUROSIM member societies. **Online SNE** in high-resolution quality with colour pages, additional SNE contributions as benchmark sources, and the **SNE Archive** will be implemented for EUROSIM member societies as additional part of their society subscription and are accessible via web for members of the EUROSIM member societies.

EUROSIM Congress 2013 Cardiff

Preparations for this congress have already started. EUROSIM Congress 2013 that will take place in Cardiff City Hall, the webpage is already online for further info: *www.eurosim2013.info*. The proceedings of the Congress will be published by IEEE CPS and IEEE Explore.

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

September 10 - 13, 2013, Cardiff, UK

www.eurosim2013.info

The EUROSIM 2013 Congress will include invited talks, parallel, special and the poster sessions. Proposals are welcome for Keynotes, Tutorials, and Special Sessions!

IMPORTANT DATES

Paper Submission10 May 2013
Notifications 15 June 2013
Camera Ready 1 July 2013
Author Registration 1 July 2013

EUROSIM 2013 VENUE

The Congress will be held in the historic and magnificent City Hall in the heart of Cardiff, the capital city of Wales, the Gala Dinner will be held in the main hall of the National Museum of Wales. Social activities include visits to Cardiff Castle and Caerphilly Castle

ASIM
German Simulation Society
Arbeitsgemeinschaft Simulation

Coming Events.

In 2011, Winterthur in Switzerland is hosting the *ASIM Symposium Simulation Technique* ASIM SST These conferece series orginates from the previous *Annual ASIM Conference* and is now held each 2nd year, in alternation with the conference series ASIM SPL - Simulation in Production and Logistics. ASIM SST focusses on interdisciplinary aspects of modelling and simulation and aims for exchange between different areas of modelling and simulation.

ASIM STS 2011

21th Symposium Simulation Technique

September 7 – 9, 2011, Winterthur, Switzerland

www.asim-gi.org

As in December 2012 the well-known WINTERSIM conference is organised for the first time in Europe (Berlin), ASIM has postponed the autumn 2012 conference ASIM SPL *Simulation in Production and Logistics* to autumn 2013 (**ASIM SPL 2013**). ASIM members are organising contributions and special sessions for WINTERSIM 2012, Berlin.

WINTERSIM 2012

Winter Simulation Conference

December 9-12, 2012, Berlin, Germany

www.wintersim.org

ASIM is co-operating with ARGESIM and EUROSIM in organising the MATHMOD conferences series. The scope of the *MATHMOD* covers theoretic and applied aspects of all types of mathematical modelling. The next MATHMOD conference takes place in begin of 2012.

MATHMOD 2012

6th Vienna Conference on Mathematical Modelling

February 15 – 17, 2012, Vienna, Austria

www.mathmod.at

PSCS – Polish Society for Computer Simulation

General Information. PSCS (The Polish Society for Computer Simulation) 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 257 members. The Board of sixth cadence consisting of the following persons directs the affairs of the PSCS:

- Leon Bobrowski - President
- Andrzej Grzyb - Vice President
- Tadeusz Nowicki - Vice President
- Zenon A. Sosnowski – Treasurer
- Zdzislaw Galkowski - Secretary
- Roman Bogacz
- Zygmunt Strzyzakowski
- Andrzej Tylikowski

→ *www.ptsk.man bialystok.pl*

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

Activities. The main activities of the Polish Society for Computer Simulation are annual conferences known as „PSCS Workshops on Simulation in Research and Development". The PSCS Workshops were organized in: Mielno (1994), Warszawa(1995), Wigry (1996), Jelenia Gora (1997, 1998), Bialystok&Bialowieza (1999), Zakopane – Koscielisko (2000), Gdansk-Sobieszewo (2001), Osieki k/Koszalina (2002), Zakopane (2003), Białystok &Augustow (2004), Sarbinowo Morskie k/Koszalina (2005), Krynica Zdroj (2006), Zakopane (2008), and Bialystok-Bobrowa Dolina(2009).

Past Events. The annual PSCS Workshop on "Simulation in Research and Development" took place on September 7-9, 2010 in Krynica Zdroj, Poland.

The papers of the workshop covered the following areas: simulation in mechanical engineering, simulation in mathematical problems, artificial intelligence and simulation, simulation in transportation, neural nets and simulation, simulation in automation and control, and simulation tools.

Coming Events. Prof. Tadeusz Nowicki will organize the 18th PSCS Workshop on „Simulation in Research and Development" on September 25-28, 2011 in Zakopane, Poland. E-mail: *tadeusz.nowicki@wat.edu.pl*

18TH PSCS WORKSHOP
Simulation in Research and Development
September 25-28, 2011, Zakopane, Poland
www.ptsk.man.bialystok.pl

Publications. The new journal „Simulation in Research and Development", Warszawa, 2010, vol. 1 (in Polish). The price is 100,- PLN.

Z. A. Sosnowski, *z.sosnowski@pb.edu.pl*

SIMS – Scandinavian Simulation Society

General Information. 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 is organised as federation of regional societies. There are FinSim (Finnish Simulation Forum), DKSIM (Dansk Simuleringsforening) and NFA (Norsk Forening for Automatisering). The SIMS Board steers the society:

- Esko Juuso, chairman
- Erik Dahlquist, Brian Elmegaard, Peter Fritzson, Kaj Juslin, Tiina Komulainen, Bernt Lie, Tommy Mølbak
- Vadim Engelson, SIMS coordinator for practical matters

You can contact the chair of the SIMS board,

Esko Juuso, Control Engineering Laboratory, Department of Process and Environmental Engineering, 90014 University of Oulu, Finland

To become a member of SIMS you should join one of the SIMS member organizations, as specified on the SIMS web page

→ *www.scansims.org*

✉ *info@scansims.org*

Past Events

SIMS 51 - The 51st Scandinavian Conference on Simulation and Modelling, was organised by FinSim at The Conference Centre Lasaretti, Oulu, Finland, October 14-15, 2010. The programme was focused on modelling and simulation in energy, environment, mining and metal industry.

The programme consisted of two keynote and 30 regular papers. Both days were opened with keynote presentations. The topics were: (1) Modeling and Control of Hydro Power Systems - an Overview with Trends (Prof. Bernt Lie, Telemark University College, Norway), and (2) Structure of a Property Based Simulator for Minerals and Metallurgical Industry (Senior Technology Adviser, Ph.D., Pertti Lamberg, Outotec Research, Finland).

Energy systems, process industry, water treatment, services and maintenance were important topics throughout the conference. Energy systems were discussed in a wide perspective, including power plants, pyrolysis processes, fludized beds, fuel cells, residential buildings and district heating.

Water treatment session covered both water and wastewater and extended to a biofilm process and a water distribution network. Digester fibre lines were the topics in pulp and paper. Metal industry applications combined various modelling methodologies, including plug flow, CSTR, CFD, equilibrium calculation and parametric modelling, for a AOD process, oxygen blowing, flotation, crystallization and alumina digestion.

Condition monitoring and modelling papers provided links between processes and services. Education was dealt in connection with chemical industry. Advances on simulation tools were presented as well, and there was an exhibition.

A panel discussion on Simulation Methodologies and Software Tools in Future Applications and Education was chaired by Esko Juuso. Panelists were prof. Bernt Lie, prof. Erik Dahlquist, Dr. Kaj Juslin and Dr. Pertti Lamberg.

The audience took actively part on discussions. There were over 40 participants. The proceedings will be available online.
More information → *ntsat.oulu.fi/.*

The ***3rd OpenModelica Annual Workshop*** was organised by The Open Source Modelica Consortium (OSMC) and Linköping University in Linköping, February 8, 2010. OSMC is a non-profit organization supporting the development of the OpenModelica Open-Source implementation of Modelica for industrial and academic usage.

There were 10 presentations, a panel discussion on Future Directions of OpenModelica and Open Source Modelica Consortium Annual Statutory Meeting.
More information → *www.openmodelica.org*

The ***4th MODPROD Workshop on Model-Based Product Development*** was organised by The Center for Model-based Product Development (MODPROD) at Linköping University, February 9-10, 2010. MODIPROD 2010 focused on model-based tools and methods for electronic systems and software. The workshop was organised in two days.

The programme consisted of three keynote and 27 regular papers and a panel discussion. On February 8 there were five tutorials in parallel.
More information → *www.openmodelica.org*

The Swedish National Energy Convention was organised by The Swedish Energy Agency, Stockholm International Fairs and Enercon/Konferensforum (PCO), March 16-17, 2010. It is Sweden's by far largest and most important energy conference.
More information → http://www.energimassan.se/

The ***Control Systems Conference 2010*** was organised by SPCI (The Swedish Association of Pulp and Paper Engineers) and Innventia (one of the world's leading R&D companies in the fields of pulp, paper, the graphics media, packaging and logistics), September 15-17, 2010, Stockholm, Sweden.

The Control Systems conference is held every two years and is the preeminent conference in the area of process measurement and control and systems engineering for the pulp and paper industry. The conference attracts presenters and attendees from all major pulp and paper producing countries, so this will be your best opportunity to network with experts from all around the world and to learn more about important ongoing work in the field.

The conference focusses towards the practical use of new and emerging technologies and on research on the brink of implementation.
More information → www.controlsystems2010.com

Servomøtet was organised by NFA October 20-21, 2010 in Ålesund. The programme consisted of 18 presentations.
More information → *www.nfaplassen.no*

The ***4th OpenModelica Annual Workshop*** was organised by The Open Source Modelica Consortium (OSMC) and Linköping University in Linköping, February 7, 2011. OSMC is a non-profit organization supporting the development of the OpenModelica Open-Source implementation of Modelica for industrial and academic usage. There were 12 presentations, a panel discussion on Future Directions of OpenModelica and Open Source Modelica Consortium Annual Statutory Meeting.
More information → *www.openmodelica.org*

The ***5th MODPROD Workshop on Model-Based Product Development*** was organised by The Center for Model-based Product Development (MODPROD) at Linköping University, February 8-9, 2011. MODIPROD 2011 focused on model-based tools and methods for electronic systems and software. The workshop was organised in two days. The programme consisted of four keynote and 18 regular papers, posters and a panel discussion. On February 8 there were 6 tutorials in parallel.
More information → *www.openmodelica.org*

19th Automation Seminar was organised in Helsinki, March 15-16, 2011. The programme consisted of two keynote, two theme sessions with 12 presentations and 13 regular sessions with 52 presentations, as well as practical demonstrations. The seminar is the major domestic forum for presenting notable results from research, drawing attention to commercial products, systems, and services, exchanging real-life experiences, and – very importantly – bringing together automation professionals, as has been our tradition every two years.

The seminar is directed to researchers, developers and end users both from industry and academia that shares the interest, challenges, and needs for future automation and control systems. The scope, united automation, describes well how widely automation is used and that automation is heavily multidisciplinary science. There was a special track on modeling and simulation, including three sessions with 13 papers.
More information → *www.automaatioseura.fi*

Coming Events

SIMS 2011

52nd Scandinavian Conference on Simulation and Modelling

September 29-30, 2011, Västerås, Sweden

www.scansims.org

The 52nd Scandinavian Conference on Simulation and Modelling, will be organised by Mälardalen University, Västerås, Sweden, September 29-30, 2011.

The purpose of the SIMS conference is to cover broad aspects of modeling and simulation and scientific computation. It will be of interest for model builders, simulator personnel, scientists, engineers, vendors, etc. Special emphasis is on energy and environment applications.

The scientific program will consist of technical sessions with submitted and invited papers, and is open for poster sessions and vendor demonstrations. The focus areas are energy and environment.

Proceedings of the accepted papers will be distributed at the conference. Presented papers will be considered for publication in the EUROSIM scientific journal "Simulation and Modelling - Practise and Theory (SIMPRA)“ published by Elsevier Science.
Especially Ph.D. students are encouraged to con-tribute with papers according to the conference themes. More information → *www.scansims.org ´*

Esko Juuso, *esko.juuso@oulu.fi*

ASIM - Buchreihen / ASIM Book Series

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

I. Troch, F. Breitenecker (eds): *Proceedings MATHMOD 09 - Abstract Volume / Full Papers CD Volume*. Proc. 6th Conference Mathematical Modelling Vienna, February 2009, Vienna; ARGESIM Reports no. 34 & no. 35, ASIM / ARGESIM Vienna, 2009; ISBN 978-3-901608-34-6, ISBN 978-3-901608-35-3

M. Rabe (ed.): *Advances in Simulation for Production and Logistics Applications*. Proc. 13. ASIM-Fachtagung Simulation in Produktion und Logistik, October 2008, Berlin; Fraunhofer IRB-Verlag, Stuttgart, 2008, ISBN 978-3-8167-7798-4.

B. Zupančič, R. Karba, S. Blažič (eds.): *Proceedings EUROSIM 2007 - Abstract Volume / CD Volume*. Proc. 6th EUROSIM Congress on Modelling and Simulation, Sept. 2007, Ljubljana, Slovenia. ARGESIM Report no. 32, ASIM / ARGESIM Vienna, 2007; ISBN 978-3-901608-32-2.

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.

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

Fortschrittsberichte Simulation (FB) - ARGESIM Reports (AR) - Special Monographs, PhD Theses, ASIM Workshop Proceedings

D. Leitner: *Simulation of Arterial Blood Flow with the Lattice Boltzmann Method* ARGESIM Report 16, ASIM / ARGESIM Vienna, 2009; ISBN 978-3-901608-66-7.

Th. Löscher: *Optimisation of Scheduling Problems Based on Timed Petri Nets*. ARGESIM Report 15, ASIM / ARGESIM Vienna, 2009; ISBN 978-3-901608-65-0.

R. Fink: *Untersuchungen zur Parallelverarbeitung mit wissenschaftlich-technischen Berechnungsumgebungen*. ARGESIM Report 12, ASIM / ARGESIM Vienna, 2008; ISBN 978-3-901608-62-9.

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

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

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

C. Deatcu, P. Dünow, T. Pawletta, S. Pawletta (eds.): Proceedings 4. ASIM-Workshop Wismar 2008 – Modellierung, Regelung und Simulation in Automotive und Prozess-automation. ARGESIM Report 31, ASIM / ARGESIM Vienna, ISBN 978-3-901608-31-5, 2008.

J. Wittmann, H.-P. Bader (Hrsg.): *Simulation in Umwelt- und Geowissenschaften - Workshop Dübendorf 2008*. Shaker Verlag, Aachen 2008, ISBN 978-3-8322-7252-4.

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: ASIM / ARGESIM Publisher Vienna - WWW.ASIM-GI.ORG
SCS Publishing House e.V., Erlangen, WWW.SCS-PUBLISHINGHOUSE.DE
Download ASIM Website WWW.ASIM-GI.ORG (partly; for ASIM members),
Bookstores / Buchhandlung, tw. ermäßigter Bezug für ASIM Mitglieder WWW.ASIM-GI.ORG

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013

EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM2013, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics

The EUROSIM 2013 Congress will include invited talks, parallel, special and the poster sessions. The Congress topics of interest include, but are not limited to:

- Intelligent Systems and Applications
- Hybrid and Soft Computing
- Communication Systems and Networks
- Case Studies, Emergent Technologies
- Workflow Modelling and Simulation
- Web-based Simulation
- Security Modelling and Simulation
- Computer Games and Simulation
- Neural Networks, Fuzzy Systems & Evolutionary Computation
- Autonomous Mental Development
- Bioinformatics and Bioengineering
- Circuits, Sensors and Devices
- e-Science and e-Systems
- Image, Speech & Signal Processing
- Human Factors and Social Issues
- Industry, Business, Management
- Virtual Reality, Visualization and Computer Games
- Internet Modelling, Semantic Web and Ontologies
- Computational Finance & Economics
- Systems Intelligence and Intelligence Systems
- Adaptive Dynamic Programming and Reinforcement Learning
- Methodologies, Tools and Operations Research
- Discrete Event /RT Systems
- Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
- Control of Intelligent Systems and Control Intelligence
- Robotics, Cybernetics, Control Engineering, & Manufacturing
- Energy, Power, Transport, Logistics, Harbour, Shipping and Marine Simulation
- Semantic & Data Mining

Congress Venue / Social Events

The Congress will be held in the historic and magnificent City Hall in the heart of Cardiff, the capital city of Wales. The Gala Dinner will be held in the main hall of the National Museum of Wales. Social activities include visits to Cardiff Castle and Caerphilly Castle.

Congress Team: K. Al-Begain, A. Orsoni, R. Zobel, R. Cant, D. Al-Dabass; *kbegain@glam.ac.uk*

Info: www.eurosim2013.info


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