Quasi-Parallel Approach to Optimization

Eugene Kindler*
Jiří Weinberger**
*Department of Informatics and Computers, Ostrava University
30. dubna 22, CZ–701 03 Ostrava, Czech Republic
ekindler@centrum.cz
** TIMING Praha, Na Poříčí 12, CZ-110 00 Praha 1, Czech Republic
timing@timing.cz, www.timing.cz

Abstract
The paper is oriented to a non-standard method of optimizing various systems by means of object-oriented simulation. The substance of the method consists in modeling parallel development of several model variants so that they tend - within an evolutional environment - to the optimum. Each of the variants has its own simulated time and during that time it develops, communicates with the other variants and – being stimulated by them – it modifies its own parameters. The variants that develop in a parallel manner but in different time flows can be realistically interpreted in a "quasi-parallel" manner within a mono processor system; that enables to reproduce the computing; certain obstacles related to the quasi-parallelism can be surmounted. The programming technology, system metaphor and application are described. In the project management field the method renders it possible to estimate the real value of a project as an alternative of compound real option approach.
Keywords: anticipatory systems, optimization, simulation, project risk management, real options, quasi-parallel handling, co-routines.

1 Introduction – Frequent Cases of Professional Anticipation

An anticipation system in a weak sense has a formal model of its own and uses it to get data of its possible future states; according to these data the system can modify its reactions to its instantaneous state and input so that its development aims at certain objectives [1]. Nowadays, simulation models implemented on digital computers often represent such formal models; the contemporary advantage of computers is their speed and the capacity of their memory, which allow to apply models (a) consisting of many thousands of elements forming very complex structures, and (b) reflecting millions of more or less interacting events per second.

Let us focus to professional domains. The objectives of anticipatory systems form a large spectrum, beginning from security against deadlocks (in rare cases) over sums of time intervals (e.g. dead times in production or transport systems) to more or less continuous operation satisfying certain demands (in frequent cases of technology, ecology, services, financial worlds and other domains). The evaluation of such a more or less continuous operation can be simply measured by means of time integration of some values that change during the system's simulated existence. Often the integrated values are non-negative, like queue length, energy consumption, material or financial input, energy and/or material production, etc. By dividing by the length of the time, during which the integration was performed, the integrals serve for computing mean values. The objective of system optimization is to find the values of parameters of the studied system so that for these values certain given mean values are optimal; for some mean values the optimum is defined as minimum, for other mean values the optimum is defined as maximum.

Let us consider two non-negative functions $f(t)$ and $g(t)$, . Let us consider their time integrals $F(\tau2)$ and $G(\tau2)$ computed over time interval $I=(\tau1, \tau2)$. Let us suppose $F(\tau2)$ is greater than $G(\tau2)$. From that relation, one cannot derive any consequence for a similar relation for the integrals $F(\tau3)$ and $G(\tau3)$ computed over time interval $K=(\tau1, \tau3)$, where $\tau3$ is rather greater that $\tau2$. In what follows, let the term "GL-ordering" represent the binary relations for comparing real numbers, like "greater than", "less than", "greater or equal to" or "less or equal to". And let us define that the expression "GL-relation switches (at time T)" means, that while it is observed between the same pair of integrals it was "greater (or equal)" before T and "less (or equal) after T" or vice versa. The just mentioned abstract consideration can be interpreted into the universe of anticipatory systems as follows. Let two systems Φ and Γ be observed during the time interval I of their existence and let in each of them be a variable depending upon time so that in Φ it behaves like f and in Γ it behaves like g. Then, from the pure mathematical point of view, no consequence for a GL-relation between their integrals $F(\tau3)$ and $G(\tau3)$ can be derived from a similar relation observed for their integrals $F(\tau2)$ and $G(\tau2)$. Nevertheless, if for example $F(\tau2)$ is observed as much greater than $G(\tau2)$ then one could anticipate that later, i.e. at time $\tau3$, the integral $F(\tau3)$ could also be greater than the integral

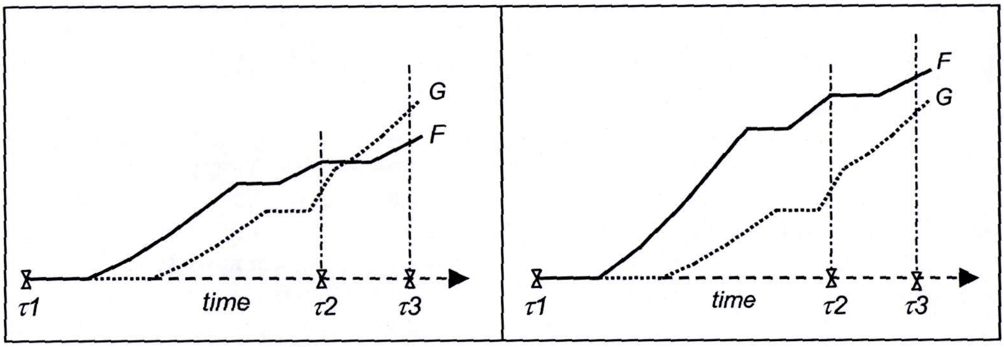

Figure 1: Dynamic relations between time integrals and its anticipation for time $\tau3$

$G(\tau3)$. A simple example is in Fig. 1, where two integrals F and G developing during time are illustrated. Of course, some assumption on stationary or even ergodic must be adopted, however, we are not going into any details in this kind of statistical reasoning.

Let us note that it is the first stimulus for discovering the main idea applied in the method described in the present paper. Naturally, this idea is still very poor, non-intelligent, it contradicts elementary mathematical reasoning, but, as it will be explained, it is opened for large improving. Now, let us use this idea for formulating the first metaphor.

2 First Metaphor of Software for Optimizing

When a person (or a team) $\Phi*$ applies a conventional simulation for getting information on a system intended for realization, such a person or a team is an anticipatory system. $\Phi*$ uses the simulation model to generate data on the possible operation (existence, behavior) of the designed system and – informed by that data that he (or the team) can change the design.

Let us apply the principles derived in the previous section and, according to them, let us introduce another anticipatorysystem S, having the following properties:

(a) it contains three persons (or teams) $\Phi*$, $\Gamma*$ and Ψ and two systems Φ and Γ,
(b) both Φ and Γ have an entity x that – in general – in Φ develops in a different way than in Γ,
(c) the integrals of x over interval $(\tau 1, \tau 2)$ be denoted $F(\tau 2)$ in Φ and $G(\tau 2)$ in Γ, (suppose $\tau 1$ be given as a constant)
(d) people $\Phi*$ and $\Gamma*$ are interested in GL-relation between $F(\tau 2)$ and $G(\tau 2)$,
(e) Ψ is a certain observer or judge in S; at time $\tau 2$ he (or the team) observes the GL-relation between $F(\tau 2)$ and $G(\tau 2)$ and anticipates that the same GL-relation should hold at a later time $\tau 3$.

Ψ takes place in case S is a certain primitive, poor and non-intelligent anticipatory system. Let μ be a formal model of such a system S. Then it should follow a statement "GL-relation between $F(\tau 2)$ and $G(\tau 2)$ does not switch when $\tau 2$ is enlarged" and can be classified as simple, trivial and often false. But let us change S so that Ψ observes simulation models ϕ and γ of the systems Φ and Γ. The new version of S is a primitive anticipatory system, too, but it offers a certain special aspect that could be characterized as *anticipatory system of second order*: each of the simulation models ϕ and γ causes S to be an anticipatory system that may anticipate the behavior of Φ or Γ, but using μ for guessing on the GL-relation between $F(\tau 3)$ and $G(\tau 3)$ (now based on GL-relation between $F(\tau 2)$ and $G(\tau 2)$ observed at the models ϕ and γ) is a formal model (though primitive) that anticipates on models ϕ and γ – see Fig. 2.

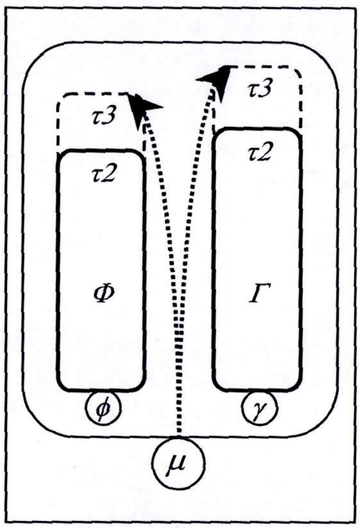

Figure 2: Scheme of anticipatory system of

3 The Role of Steady States

As we mentioned, μ is very primitive model and often offers more seduction than anticipation. A question arises whether the quality of μ could be improved.

The simulation people speak about *steady-state* of the studied systems, say that its testing is usually quite an easy task, since a "non steady state used to be very different, however, for rigorously studying the problem we refer e. g. to [2]. There were mathematical experiments to define what term steady-state means, but their applications demonstrate that each of the definition follows idealized and simplified cases and that it is not possible to convert the definition into a robust algorithm that would decide on a commercially viewed system (i.e. without essential simplification) when or whether it has or has not entered its steady state. Expressed in the language oriented to applications (i.e. in a language that is not exact), steady state is the phase when the studied system behaves "in an ordinary manner". A frequent and general situation, in which a real system exists out of its steady state, is that after its start from scratch; for example, when a hospital, a factory, a transport system or that of supply chain was just built and starts operating, it is not in steady state, as its resources and transport lines are idle and queues are empty, so that no capacity overflowing and dead times spent by waiting in queues menace; observing a system inhering in such an initial phase can tell no serious information on its longer existence in future, on its operating in "a normal situation", professionally said in "the steady state". The phase before a system enters its steady state is called *start phase*. If one has to determine optimal parameters of such a system (e.g. number of resources, capacity of stores, geometrical configuration etc.) he must count with steady state and not with some exceptional situations like that of start phase.

The starting phase and steady state is mapped in the corresponding simulation model. It is constructed, debugged and run but – in general – its run does not start from its steady state but before it occurs in its start phase. Therefore the view applied for observing real systems is interpreted also to the simulation models, namely in the most frequent case of its application, when they are applied for generating information how different variants of the designed system would operate; the objective of determining the optimum variant is related to the system operating in steady state and so the computing way to the optimum configuration should evaluate only the simulation models performing within their steady state.

Note that the steady state is not a state that lasts without any change. In commercial applications, one could assume the following law: if Φ and Γ are two systems with numerical entity x, the necessary condition for judging that both of them are in steady state at time $\tau 1$ is that for any $\tau 2$ greater than $\tau 1$ the GL-relation between the integrals $F(\tau 2)$ and $G(\tau 2)$ (defined as in the preceding section) are the same (in other words: when $\tau 2$ grows up from $\tau 1$ the GL-relation between the integrals $F(\tau 2)$ and $G(\tau 2)$ does not switch). It would be also possible to state that starting from $\tau 1$, the GL-relation can be anticipated in the same manner as in the first metaphor.

The same argumentation can be transferred from the observed systems Φ and Γ to their simulation models ϕ and γ .And that can be the first starting point for the improving the idea-seduction and for its further application for optimizing.

4 First Improving the Method

Let V be a set of data structures. The data structures that are elements of V may represent parameters of a parametric system S: assigning the components of an element

$y \in V$ as parameters of S turns this parametric system to a (non-parametric) system, which will be denoted $S(y)$ and called *variant* of S. Let W be a set of numerical entities defined for S. If $w \in W$ and $y \in V$ then $w(y,t)$ represents the value of w in variant $S(y)$ at time t and $INT(w,y,t1,t2)$ represents integral of $w(y,t)$ from $t=t1$ to $t=t2$. Let $x \in V$, $y \in V$, $w \in W$ and R be a GL-relation; then we use the sentence "x dominates y for entity w over time interval $(t1,t2)$" for $R(INT(w,x,t1,t2),INT(w,y,t1,t2))$, and denote it as $DOM(R,x,y,w,t1,t2)$. Intuitively, it represents a sentence that variant $S(x)$ carries better behavior of the integral of w over $(t1,t2)$ than variant $S(y)$, where GL-relation R expresses what means "better behavior". $C(x,y,t1,t2)$ be a Boolean function composed of dominatings $DOM(R,x,y,w,t1,t2)$, where $x \in V$, $y \in V$, $w \in W$ and R is a GL-relation. Such a C is called *criterion*. $y \in V$ is called *optimum* of S relating to V, W and $(t1,t2)$ according to criterion C, if $C(y,x,t1,t2)$ is true for any $x \in V$. In such a case, $S(y)$ is called optimum variant of S. Note that y does not need being unique but discussing about it is not the subject of the present paper (such a situation roots in commercial base of the optimizing and can be an image either of a bad formulation of the criterion C or of a real situation that allows a free choice of several variants viewed as equivalent). It is reasonable not to apply $t1$ belonging to the start phase of any variant of S.

If the moment t_{st}, in which each of the variants of V is in steady state, were known then it would be possible to continue simulation for each of the variants up to a certain time t_{st}, during it to integrate the values of W, to use the given criterion C for a stepwise process of partial ordering of V – beginning with some elements of V. As V is often enormously great, it would be suitable to start with some subset U of V, to perform some steps to derive other variants of V from those already tested and ranged into the partially ordered subset, and to make it so that the derived variants would be hopeful for being better than the optimum one(s) heretofore existing in V. Such steps could be considered as models of a certain sort of adaptation, or – better – each variant v_i of V could be considered as governed by a certain virtual expert ε_i so that the community of N experts would "live, think and communicate": during the communication, any expert could be inspired by his colleagues so that he would modify his variant and possibly start to think on another one, on which he could hope to be better than the best variants owned by his colleagues.

5 Another Metaphor

The community of the experts represents another metaphor. In general, the less successful experts may be inspired by their more successful colleagues so, that a less successful expert refuses his variant – either as the whole or only in some of its parameters – and creates a new variant in that he imitates one or several of its colleagues. For example, such an expert would recognize the parameters $\{\sigma_1, \sigma_2, \ldots, \sigma_R\}$ of a variant v_j owned by his more successful colleague and take one or more of them as the corresponding parameters of his own new variant, i.e. of a variant on that he would hope to be still better than v_j (of course such a hope could be dud). Or an expert could take such parameters of several colleagues over, or he could even use them as inputs for computing new values like average one etc. Another step could consist in random deviating of the parameters taken over, or extrapolating a trace that already improved a variant.

Nevertheless, there is an essential obstacle for direct application of this metaphor: t_{st} is not known and it is difficult to suppose that is will be computed even for certain special domains of the studied systems. Note that t_{st} must hold as a time for steady state for all the variants, i.e. – beside the starting ones – for those derived during the improving of the elaborated ones.

6 Further Improving the Method

Determining t_{st} is a difficult problem even for one variant, especially for the systems generated from scratch. This fact causes that about t_{st} certain prognoses, hypotheses and approximations are formulated. The same concerns simulation models of such systems. Among the prognoses on t_{st}, there are the following extreme ones: one of them can be called *optimistic one* (telling the steady state will come at a rather small time t_{opt}) and the other can be called *pessimistic one* (telling the steady state will come at a rather distant time t_{pes}).

Let us define both the terms exactly. The statement that a prognosis is optimistic means that every other prognosis on the time t_{st} supposes it being greater than t_{opt}. And, inversely, the pessimistic prognosis is that with the greatest approximation of t_{st}. In fact, t_{st} occurs somewhere between t_{opt} and t_{pes}.

One may expect that the pessimistic prognosis is the most secure: when one bases the comparing of the variants and deriving new variants on the integrals over time interval $(t_{pes},T+ t_{pes})$ he may be sure that he comes to the best solution. But the computing time for running the simulation models up to (simulated) time $T+ t_{pes}$ is long, sometimes enormous, especially when we know that in such a case many variants must be simulated (note that in applications the pessimistic prognosis may be even of some order greater than the reality of t_{st}; and note that T is in the most cases known, in certain commercial cases even prescribed by law).

In the opposite way, the optimistic hypothesis saves computing time but can lead to erroneous information on the optimum: since t_{st} may be greater than t_{opt}, the integrating over time interval $(t_{opt},T+ t_{opt})$ can proceed before attempting the steady state and so the integrated values may be distant from the typical ones and the optimum values computed under that regime may differ from those expected as optimum in the ordinary operation of the designed system.

7 Completing Metaphor

Nevertheless, the metaphor mentioned in section 5 can be adapted so that it offers the best of both the extremes, namely that it can serve as a model of computing, which is secure like computing based on t_{pes} and which exploits computer time like computing based on t_{opt}.

The metaphor can be described as a session of the experts $\varepsilon_1, \varepsilon_2,\ldots, \varepsilon_N$, mentioned in section 5. Nevertheless, they do not worry about t_{st}, but they only agree on the values of

t_{opt} and t_{pes}. Every expert ε_i of the session has a variant y_i that he understands as a hypothesis on the optimum behavior of the studied system S. Each of the experts has its own computer and starts to simulate S respecting his own variant. All experts simulate until reaching t_{opt} and then start to collect the integrated values, which demands to continue the simulation until reaching $t_{opt}+T$ where T>0 is a certain value declared for the wholes session as the duration of integration (to determine T is generally not a problem). Let $t_{opt}+T$ be identified as t_1.Then the experts start to communicate, so that each of them makes his colleagues acquainted with his variant y_i and with the results of the simulation experiments obtained by it. So one expert ε_k gets to know that according to the simulation experiment his variant y_k is the worst of all ones figuring in the session, and he decides to learn from his more successful colleagues to formulate a variant that – may be – should be a more successful. He learns similarly as mentioned in section 5. The newly formulated variant replaces y_k.

Then the session continues so that each of the experts simulated his variant until t_2= t_1+T. Expert ε_k should start from $t=0$, while the other ones can simply continue. Then the communication among the experts takes place similarly as after reaching t_1 and with a similar conclusion: an expert discovers his variant the worst and – being enlightened by his colleagues – he formulates a new variant, with which he starts from scratch in the next step (performing simulation from time zero up to t_3= t_2+T) while his colleagues need only to continue the simulation from t_2 to t_3.

So the alternating simulation-communication goes on until to t_n which is equal or greater to $t_{pes}+T$. Although the value of t_{st} is not known it had to be "met" at a step in simulating from t_p to t_{p+1} and beginning from that step the changes of the variants could approach to the optimum one.

8 Implementation of the Metaphor

The last metaphor can be slightly improved in details. The result is as follows.

If an approximation of t_{opt} and t_{pes} are at disposal one determinates a time interval (t_{opt}, τ) where $\tau > t_{pes}$ and divides it into Q subintervals of the same length, which he then declares as T (see Fig. 3). Then the computer model M of the session of experts based on the time-table and manipulating with the given parametric system S is activated. When the model is at its end it should give a good approximation to the optimum variant of S.

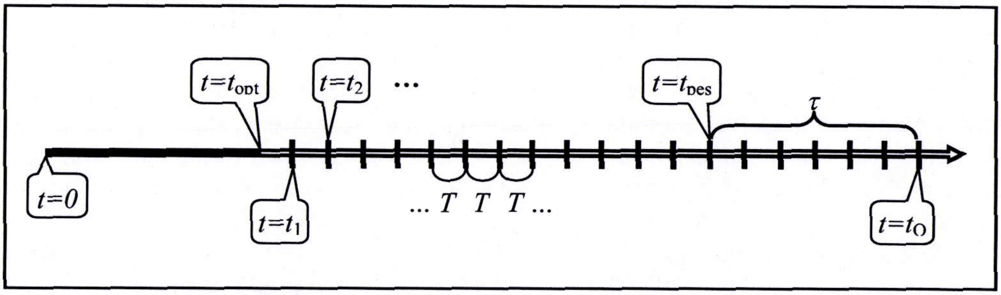

Figure 3: Using the interval between both extreme hypotheses

For the implementation, let us emphasize the statement that appeared already in relation with the first metaphor: a person or a team using model M is a certain anticipatory system of second order, because M is a model of a system containing elements (the experts) who are themselves anticipatory systems, because they use their own models, namely simulation models of the variants they think of. Among other, a consequence of that is a parallel existence of different flows of simulated time (an example is the state of M when the successful experts should continue their simulation models relating to a certain simulated time greater than t_{opt} while the expert forced to refuse his variant has to start simulation of a new variant from time equal zero). That nesting of anticipatory systems is projected in the implementation – one has to manipulate M that manipulates other models nested in it, in general each with its proper simulated time. The habitual simulation programming tools like simulation languages and packages cannot be used, because they allow handling only one Newtonian time flow during a simulation experiment.

Model M was successfully implemented at computers programmed in a monoprocessor way. It was enabled by using programming language SIMULA [3], [4] that has suitable standard tools for simulation, is object-oriented, process-oriented and block-oriented (and thus permits nesting of models and existence of more simulation models with different time flows in one "supermodel" [5]-[8]) and has *quasi-parallel* sequencing of program components so that the program product and its parts can be viewed as composed of more objects operating in parallel, though they exist at the same monoprocessor system (such quasi-parallel system of components that switch their operation at a monoprocessor system to model a multiprocessor operation is to be preferred also because its run is deterministic and reproducible). SIMULA allowed developing software systems based on the principles mentioned in the present paper and oriented to optimizing [9] and multioptimizing [10] simulation models. The user is expected to complete a description of a model S, an algorithm for computing a Boolean function that tests whether a structure of parameters belong to the set V of permitted structures, and a procedure for evaluation the GL-relations (all three procedures are added as declarations of the procedures with the same names introduced in the software systems as virtual).

9 Applications

The first experiments and experiences were obtained by use of main-frame IBM computers, with which many systems were optimized, covering a large spectrum of scientific and technical domains from steel and machine production [9], [10], over services until neurophysiology of brain [11]. During that development a lot of experiences were obtained, which enabled improving the mentioned software system. Nowadays the software works at PC under Windows and is widely used as a part of a greater software product PMF (Project Management Forcast) [12]. Together with other modules (for risk analyzis etc [13]), the optimization is nested in an interactive dialog which enables describing the project structure in special formal language. The whole software system can be bought at the address of TIMING Praha (see under the title of the paper).

10 Summary

Optimizing a system is a frequent and typical activity that characterizes a person or a (part of a) society as an anticipatory system, namely that in the weak sense. For the last fifty years, computers offer to help the optimizing; the growing operation rate of the computers offer to use simulation for that purpose. That technique can be characterized so that the formal models figuring in the anticipatory systems in weak sense according to their definition become simulation ones.

There are two obstacles related to the optimizing of simulation models. One is the determining of the moment when the models is in a "steady state", i.e. when its initial run enters the state representing a situation that could exist in the simulated system when is would exist. The other problem is that the optimizing needs much computing time when the optimized function has to be computed as a result of a simulation experiment.

The paper offers ideas that are able to surmount both the obstacles. The ideas root in the fact that the optimized function is almost always a time integral of a positive (vector or scalar) function generated during the run of a simulation experiment, i.e. a function that monotonely grows during that run; that enables that prior to the end of simulation experiments the integrals related to different variants can be compared and their configuration can be used to a way to probable optimum. On the time when a given system reaches its steady state, one can formulate many hypotheses based on intuition, on analogies taken from the former case studies, and on mathematical argumentation based on simplified aspects of the concerned system. Among them, there is the "pessimistic" hypothesis (considering the maximum value of the time when steady state could be reached) and the optimistic one (considering the minimum value of such a time). The difference D between both the hypotheses is often enormous. One can be sure that the accumulating the considered data performed after reaching the steady state according to the pessimistic hypothesis gives a true value of the integral. Taking into account that that hypothesis is really pessimistic, one could hope the integration should have start sometimes before. Thus the integration start already at time of reaching optimistic hypothesis and during the phase until reaching the pessimistic one the data that are to be optimized could already more or less approach to their optimum configuration.

When a usual computing technique is at disposal, the described method needs to allow running more than one simulation experiment at the same computer task and the simulated time flows of such experiments have to be mutually independent. Beyond a very small number of exceptions, the existing programming languages and other tools do not allow it in a bearable form. The programming languages that are truly object-oriented, agent-oriented and block-oriented allow that. Good experiences were made with SIMULA [3], [4] which is a language with PC implementation under Windows and under LINUX is fee and efficient and enabled a wide spectrum of applications (see chapter 9). Nevertheless, to understand class and agent nesting is difficult and therefore some metaphors were elaborated to facilitate the understanding (see chapters 2 and 5).

11 Conclusion

The view of the model of communicating experts is near to the view of real discussions of real experts tending to an optimal design of a system. The basic idea of such a session came to existence in the 80-ies of the XX century, i.e. before the genetic algorithms were commonly known. Naturally, the development of the variants handled by the communicating experts has a lot of properties that could be observed in the developing generations of cells, but the metaphor of communicating experts appeared sufficient. Nevertheless, at the present months, the description of model M is transformed as being viewed from the direction of genetic algorithms.

References

1. Dubois Daniel M. (2000) Review of Incursive, Hyperincursive and Anticipatory Systems - Foundation of Anticipation in Electromagnetism. Computing Anticipatory Systems: CASYS'99 – Third International Conference. Edited by Daniel M. Dubois, Published by The American Institute of Physics, AIP Conference Proceedings 517, pp. 3-30.
2. Lee Jong-Suk R., Mc Nikle D. and Pawlikowski K. (2000) Initial Transietnt Period Detection for Steady State Quantile Estimation. Proc. Int. Summer Computer Simulation Conf. Published by Int. Society for Computer Simulation Press, pp. 213.1-6.
3. Dahl O.-J., Myhrhaug B. and Nygaard K. (1968). Common Base Language (1st ed.). Norsk Regnesentralen, Oslo. 1972 (2nd ed.), 1982 (3rd ed.), 1984 (4th ed.).
4. SIMULA Standard (1989). SIMULA a.s., Oslo.
5. Kindler E. (2001) Computer Models of Systems Containing Simulating Elements. Computing Anticipatory Systems. CASYS 2000 – Fourth International Conference. Edited by Daniel M. Dubois, Published by The American Institute of Physics, AIP Conference Proceedings 573, pp. 390-399.
6. Kindler E. (2000) Nesting Simulation of a Container Terminal Operating With its own Simulation Model. JORBEL (Belgian Journal of Operations Research, Statistics and Computer Sciences), 40, pp. 169-181.
7. Berruet P., Coudert T. and Kindler E. (2004) Conveyors With Rollers as Anticipatory Systems: Their Simulation Models. Computing Anticipatory Systems: CASYS 2003 – Sixth International Conference. Edited by Daniel M. Dubois, Published by The American Institute of Physics, AIP Conference Proceedings 718, pp. 582-592.
8. Kindler, E., Coudert T. and Berruet P. (2004) Component-Based Simulation for a Reconfiguration Study of Transitic Systems, SIMULATION, 80, pp.153-163.
9. Weinberger J. and Mojka A. (1983) Optimization of an Industrial Simulation Model by Means of Quasi-Parallel Handling. Ekonomicko-matematický obzor, 19, pp. 179-187.
10. Weinberger J. (1988) Evolutional Approach to Extremization of Vector Criteria of Simulation Models. Acta Universitatis Carolinae Medica, 34, pp.249-257.

11. Faber J. and Weinberger J. (1988) Thalamocortical Reverberation Circuit Simulation Using The Simula Language. Acta Universitatis Caorlinae Medica, 34, pp. 149-248.
12. Weinberger J (2005) Steep Steps to Optimization of Simulation Models. Simulation Almanac 2005. Edited by M. Šnorek, J. Štefan, Publisher by The Czech and Slovak Simulation Society and Czech Technical University, Prague, pp.145-150.
13. Weinberger J. and Kopeček S.(2004) Object Oriented Modeling in Investment Project Feasibility and Profitability Analyse. PRONT 03, konference on PROject managemeNT. Edited by M. Brejcha, Publisher by EVIDA, Pilsen, Czech Republic.