DECISION SUPPORT METHOD FOR EFFICIENT SEQUENTIAL AND PARALLEL SIMULATION: TIME DECOMPOSITION IN MODIFIED CONCEPTUAL MODELS

László Muka
Elassys Consulting Ltd.
Bég utca 3-5.
H-1026 Budapest, Hungary
e-mail: muka.laszlo@elassys.hu

Gábor Lencse
Department of Telecommunications
Budapest University of Technology and Economics
Magyar tudósok körútja 2.
H-1117 Budapest, Hungary
e-mail: lencse@hit.bme.hu

KEYWORDS

efficiency of simulation, time decomposition, modified conceptual model, resolution of conceptual model, execution time, execution-state model, execution-path model, business process, organisational process, information and communication technology, process simulation, parallel simulation, sequential simulation

ABSTRACT

A new method, the time decomposition in modified conceptual models (MCM) is described. There are introduced two approaches to model the function execution time in MCMs, necessary to decomposition: the execution-state approach and the execution-path approach. There are also described transformations necessary to manage the resolution of MCMs. There is also introduced the concept of preliminary simulation to the time decomposition of MCMs in process simulation environment. There is given an example of using decomposition results to support sequential-parallel simulation decisions increasing the efficiency of simulation.

INTRODUCTION

Efficiency of Simulation Projects

The *efficiency* of simulation projects aimed to support the design of Information and Communication Technology (ICT) and Business Process (BP) systems in an organisation is influenced by some key factors:

These projects usually begin with an *unstructured,* “soft” situation because of the compound system of goals and different roles of the participants (Checkland 1989, Sierhuis and Clancey 2002; Warmerdam and Bredveld 2003).

Common analysis of ICT and BP systems may have advantages but in this case we need to have methods with a *double feature* that is appropriate for both types of systems.
(For example, Warmerdam and Bredveld express a clear need to have methods dealing with the *double feature* when describing the BSM (Business Service Management) method (Warmerdam and Bredveld 2003): by the BSM method the current and future BPs and their requirements (service-level requirements) should be matched with the *performance* and availability of ICT systems required to support them, together with the BPs operating the ICT systems.)

In our previous papers (Muka and Lencse 2006, Muka and Lencse 2007; Lencse and Muka 2007) we outlined a set of methods, the modified conceptual model (MCM) approach, the meta-methodology coping with the unstructured aspect and with the double feature of the simulation projects, outlined above.

In this paper we deal with other key factors of *efficiency*: the problem of finding the right *resolution* of a model and also finding the right *simulation execution methods* (sequential and parallel).

Our proposal to answer these questions is the *method of time decomposition in modified conceptual models,* addressing the problem of *model resolution* and supporting the decision on the use of *sequential and parallel simulation* helping in speeding up the simulation.

CONCEPTUAL MODELS

Traditional Conceptual Models

Soft Systems Methodology (SSM) was developed to be used with unstructured problems, defined by Jackson and Keys (Jackson and Keys 1984, Jackson 1991), in Human Activity Systems (HAS) (Checkland 1989, Checkland and Scholes 1990). The outcome of SSM may also be used with information system analysis and design (Curtis 1989).

The traditional process of SSM consists of seven stages: 1. Entering the Problem Situation, 2. Expressing the Problem Situation, 3. Identifying Relevant Systems, 4. Developing Conceptual Models, 5. Comparing Conceptual with Actual, 6. Defining Changes, 7. Implementing Agreed Changes.

The central idea of SSM is the *conceptual model* of HAS. Conceptual models are the *views of what exist* and not models of what exist in the real world.

The main elements of conceptual models are *key activities* representing *subsystems* of the system. The selected set of subsystems with their logical connections is the conceptual model. The set of conceptual models with defined connections among them form a system of conceptual models. A hierarchy of conceptual models can be found when replacing a first-level conceptual model of a subsystem with its detailed conceptual model.

Modified Conceptual Models

In our previous paper (Muka and Lencse 2006), using ideas of Gregory (Gregory 1993) about necessary and sufficient conditions and also “temporal relations” in conceptual models as the starting approach, we have described the *modified conceptual models (MCM)* with extended harden-up elements to SSM conceptual models, focusing on the design of information systems in an organisation.

In an MCM a key activity is performed generally by an Organisational Process (OP) function or by an ICT system function, that is, any function in the organisation can be performed by some relevant organisational process (P subsystem) with its human resources or by some relevant IT system (IT subsystem) with its technical resources.

Thus MCM elements can be P-type or IT-type depending on what they represent, OP or ICT system function.

An important feature of MCM is that any IT element in the model should be connected to a minimum of one P element in order to have its human resource connection.

To the analysis of necessary and sufficient conditions in MCM, there were three defined element types: F, C and A, that is, PF, PC and PA for processes and ITF, ITC and ITA for IT systems.

F (*function*) is an element performing basic function in the system; element C (*condition*) is providing condition function necessary to perform basic function; while A (*agent*) is an *agent* element ensuring the sufficiency (“motivational” condition) for the basic function to be completed.

Virtual time was also introduced into MCM: *virtual time* is a time sequence assigned to an MCM by giving time labels to its elements.

The virtual time of different MCMs may also be *synchronised* through transient edges (logical connections between MCMs) and condition elements.

In virtual time of MCMs the question of *time decomposition* may also be examined.

TIMING IN MODIFIED CONCEPTUAL MODELS

Execution Time in Modified Conceptual models

Time label T given to an MCM element has the meaning that the event of “function is performed” takes place at T. Extending this definition, time label T may also denote another event: the event of “performance of a function is started” takes place at T.

To the analysis we may also choose the more appropriate definition.

Time label of MCM T_{i-1} denotes on earlier point of time then T_i but there is nothing said about the T_i- T_{i-1} interval.
To give an estimation about time intervals, we introduce *the execution time of key activities (elements)* into MCMs.

Now, we describe timing definitions in conceptual models in a more detailed way:

Timing of function (F) elements
T_i may be the time label of the *finishing event* (the event when the function execution is finished) or time label of the *starting event* (the event when the function execution is started.

If ΔT_i denotes the execution time of a function then in the case of a *finishing event* the execution starts at T_i-$\Delta T_{i,}$

Timing of condition (C) elements
The main feature of the timing of a condition is that the condition should be ready by the starting point of the execution.
Setup time is the time interval necessary to prepare the condition.
Take-off time is the time interval while the given condition is provided after the function has been executed.
It depends on the system itself whether the condition necessary to be provided during the execution time.
(For example, at a shooting championship it may take some time to set up the target to a shooter and it takes some time to take it off and to set up a new target to a new shooter.)
It depends on the system itself whether the condition should be *held* or not during the function execution. It depends on the system too whether a C element gives a start or finish signal to an F element.

Timing of agent (A) elements
The *agent time* is the time during which the *agent* is used: the agent time in our definition is equal to the execution of a function.
(For example a supervisor at a boxing match should monitor the fight and its conditions during the fighting just to *motivate* the participants to keep to the rules. Before and after the fight monitoring is not necessary but possible.)

Figure 1 shows a summary of the features of the function (subsystem) execution time.

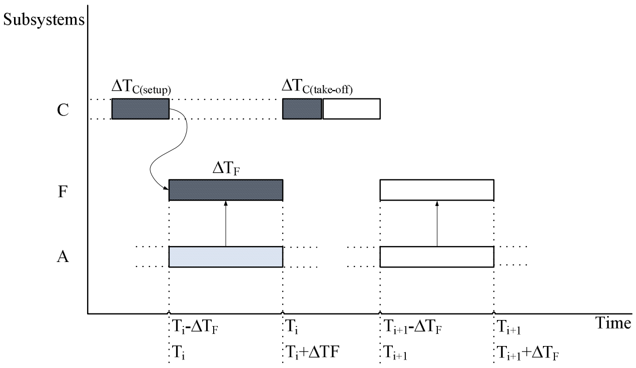

Figure 1 Elements of Function Execution Times

Synchronisation and Execution Time

Different MCMs can be connected through logical connections. These connections are realised by transient edges going between the MCMs and condition elements of MCMs.

Through transient edges and condition elements *virtual times* of MCMs may also be *synchronised*. (This is the situation in the case of the Customer Help Desk (MCM1) and the Service Department (MCM2) shown in Figure 5. These are separate but cooperating departments.)
Figure 2 shows an example for the execution time in synchronised conceptual models MCM1 and MCM2.
Let us examine the synchronisation of modified conceptual models MCM1 and MCM2: from function element F_{iMCM1} goes a transient edge to C_{jMCM2} and after the execution of F_{jMCM2} it goes back to $C_{(i+k)MCM1}$.

In case of synchronisation of MCMs the setup times of *condition elements* (ΔT_{jSETUP} and $\Delta T_{(i+k)SETUP}$ in Figure 2) used in *synchronisation* behave as execution time being in a *serial order* with function execution times ΔT_{i}, ΔT_{j} and $\Delta T_{(i+k)}$.

The other feature of condition elements in synchronisation is that they get initialisation from other function MCM elements and they act as *initialisation* for function elements. (The situation is similar for condition element C_9 in Figure 6.)

These timing features of different condition elements should be taken into account in the model building for time decomposition.

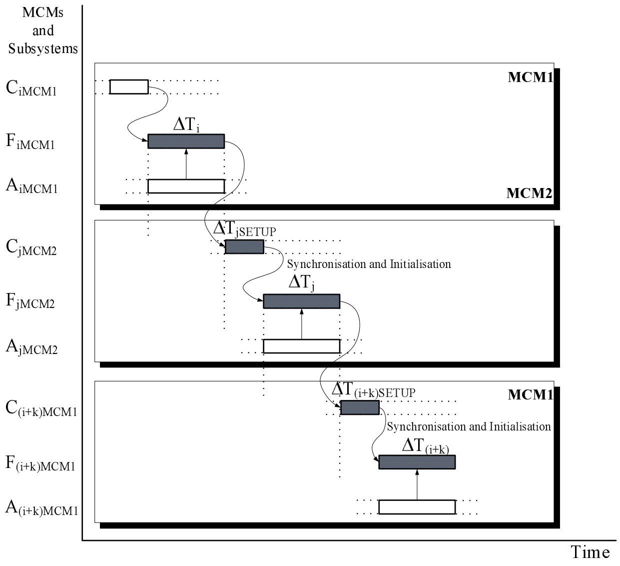

Figure 2 Execution Time in Synchronised Modified Conceptual Models MCM1 and MCM2

RESOLUTION OF MODIFIED CONCEPTUAL MODELS

Resolution of the models is an important factor from the point of view of efficiency of simulation: if the resolution is *too low* the necessary results cannot be reached but if the resolution is *too high* then we loose in modelling work and computing time.
The most *efficient resolution* is a *compromise* between these two requirements.
The appropriate level of resolution may be found by applying a set of transformations to the different parts of the model.

MCM Transformation Methods

Expansion may be used to increase resolution that is to show in greater details in MCM.
Expansion of an IT element gives IT elements showing the internal structure of the original IT element.
Expansion of a P element gives clearly P elements or P and IT elements
Grouping and *integration* are transformations decreasing the resolution of MCM.
Grouping is a transformation treating a group of elements together in modelling and also in simulation.
Integration transformation makes from two or more element one model element.
The main difference between grouping and integration is that grouping will not change the set of model elements, while integration will change it.

SUN Transformation

Now we examine how to increase the resolution of MCMs in the sense of showing different possibilities.
In Figure 3 $C_{2.1}$ – $C_{2.5}$ show the SUN (sufficient but unnecessary) conditions for F_2 as different realisation possibilities of condition C_2.
We suppose that $C_{2.1}$ – $C_{2.5}$ show all the possible realisations of the condition C_2 within the *model limitations*, that is, they are *exhaustive* in the model. This type of a set of exhaustive conditions we call *conceptionally exhaustive set of SUN conditions*.
The conceptionally exhaustive set of SUN conditions uses the limitations derived from the requirements and features of the MCM to set the boundaries in looking for the realisation possibilities of a condition.
It means that instead of purely logical consideration (Gregory 1993) we propose to operate on the requirements and features of MCM and of the system to be modelled too.

Now, we describe an approach of using SUN condition elements in our modified conceptual models.
The possibilities for SUN conditions may be collected in the following steps:

- The interval boundaries for the given condition are defined depending on the model requirements
- If it helps segments (sub-intervals and their boundaries) are defined between boundaries (Figure 3 shows segments SUN Segment1 and SUN Segment2.)
- The realisation possibilities of the condition are listed for the segments
- The set of SUN conditions is checked whether it is *conceptionally* exhaustive

Let us see an example. If we want to by a flat then both the lower and upper limit to find SUN conditions may be given by the flat area: by the accommodation feature in the case of lower limit and by the “too-big-to-clear” feature in the case of the upper limit.

In this case different SUN segments may be the different types of flats: a house with a garden, a flat in a living estate, etc.

The *conceptionally exhaustive* feature of this set of SUN conditions may be from the condition that we decide to use the advertisements only from one internet agency.

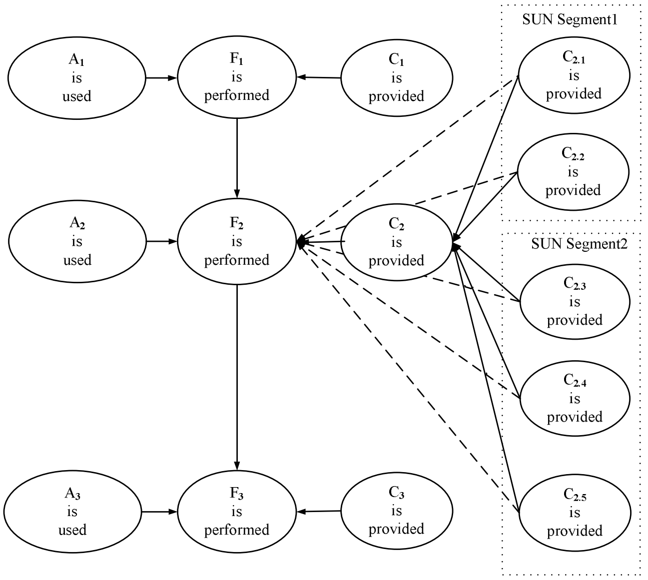

Figure 3 SUN Segments of Condition Elements

If there is an exhaustive set of SUN conditions in the model then at least one member of the set is true. (Dashed line connections in Figure 3 show the “direct influence” of SUN conditions that is the working SUN condition is the condition of F_2.)

Transformations and Timing

Grouping: the timing of the elements defines the timing of the group.

Expansion and integration: the timing of the new elements should be defined first after the situation is the same then in case of grouping.

SUN conditions: the realisation of different SUN conditions may give different execution times. The possibilities of a *conceptionally exhaustive set of SUN conditions* may be examined in serial manner, one-by-one, or in a parallel way as *competing* possibilities.

TIME DECOMPOSITION IN MODIFIED CONCEPTUAL MODELS

To the examination of the time decomposition we use two slightly different approaches:

Execution-state view: the task processing is performed by the changing states defined by the elements of MCMs

Execution-path view: the task travels through the conceptual model using a series of elements of MCMs

Execution-state Model

MCMs show the key activities or subsystems performing key activities of an organisational system.

The execution of a business task initiated by a customer can be shown in MCMs.

A *task* is a sequence of *key activities* initiated by an entering User Request in order to produce an Answer to the User when exiting the MCM.

The *execution-state* is *a set of active subsystems* (F or C) that is subsystems which are taking part in the execution.

When the system is in an execution state the execution of the task is *assigned to this active state.*

During the execution of the task, from the Entry point of time till the Exit point of time there is a series of *changing execution states*, which are necessary to perform the task.

The *transition* among the execution states is performed along the logical connections of the subsystems.

A *time slice* is a time interval during which the execution of a task is assigned to its execution state. Time slices may be different. The time between the Entry and Exit points of MCMs can be shown as a sequence of time slices (Figure 4).

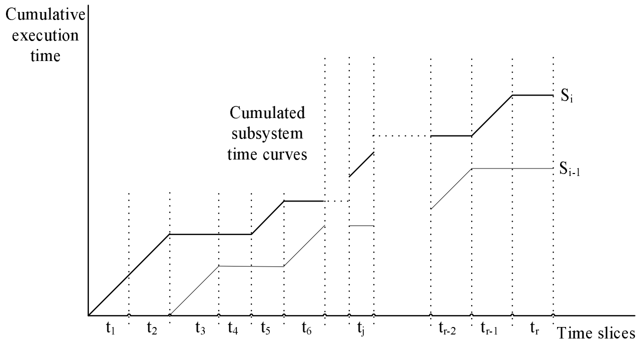

Figure 4 Cumulated Execution Time
Assigned to a Subsystem

Figure 4 shows the cumulated time spent by a task in states (subsystems) S_i and S_{i-1} during the execution.

Now, let us express the execution time in formulas:

Let us introduce the *activity variable* a_{ij} with the following meaning:

$$a_{ij}\begin{cases} = 1, \text{ if the task assigned to the subsystem } S_i \ (i = 1, 2, \ldots, n) \\ \quad\ \text{in time interval } t_j \\ = 0, \text{ otherwise} \end{cases}$$

The time spent in the execution state i may be expressed with the formula:

$$t_{Si} = \sum_{j=1}^{r} a_{ij} t_j \quad \text{where}$$

t_j is the length of time slice j (j=1,2,…,r)
r is the number of time slices
t_{Si} is the time spent in execution state S_i during the task execution.

The full time can be found using the following formulas:

$$t = \sum_{i=1}^{n} \sum_{j=1}^{r} a_{ij} t_j \quad \text{or} \quad t = \sum_{i=1}^{n} t_{Si} \quad \text{where}$$

n is the number of execution states.

The following equation shows the ***decomposition of execution (system) time*** according to the execution state assignment

$$t = t_{S1} + t_{S2} + t_{S3} + \ldots + t_{Sn} = t_{system} \geq t_{execution}.$$

If the execution is assigned to mutually exclusive states that is the set of execution states contains *mutually exclusive elements,* then

$$t_{system} = t_{execution}.$$

If *parallel assignment* is allowed (the states do not exclude each other mutually) the execution time may differ from the sum of time when the subsystems are active. In this case matrix element a_{ij} may be equal to 1 for different values of i at a given j.

Decomposition supports the definition of the critical set of subsystems to be modelled.

The system assignment time may be longer then the Entry-Exit execution time of a task.

Both the $t_{execution}$ and the t_{system} are useful:

- $t_{execution}$ is critical from the point of view of reaction/answer time – that is where and how to decrease these parameters.
- t_{system} is critical from the point of view of resources used by the subsystem – that is how to use a reduced amount of resources.

In a *preliminary simulation* we may take care of the problem of the parallel assignment that is, about the execution time overlapping by just collecting data on parallel assignment too.

In a *measurement,* based on historical data the data on parallel assignment may be missing thus special attention should be paid to the evaluation.

Execution-path Model

The execution path is a route along which the *activating information (message) and/or entity* travel in and between MCMs within the Entry and Exit points of time.

The modified conceptual model can be described as a directed graph MCM(F;C;A;E;TR;ENTRY;EXIT)

where

F, C, A are the set of function, condition and agent nodes respectively (the F, C and A sets may contain P-type or IT-type elements)

E is the set of directed edges of the given MCM

TR is the set of directed transient edges, going to or coming from another MCM, that is transient edges connect elements in different conceptual models

ENTRY, EXIT is the entry and exit edge.

Thus the elements of an execution path may be: F and C elements of MCMs, E edges inside of MCMs, TR edges between the MCMs and ENTRY, EXIT edges to enter and leave the path.

The time is elapsing in nodes while the edges do not cause any delay.

Arrows of E, TR, ENTRY and EXIT edges show the direction of movement along the path.

The sequence of MCM elements which are taking part in the task processing forms an *execution path.*

These elements can be mainly function elements or condition elements in special cases and also logical connections among the elements of MCM.

The execution time of a task depends on the execution time of the function elements taking part in the execution path.

PROCESS SIMULATION APPROACH TO TIME DECOMPOSITION

In the next, we assess the similarities and differences between BPs and MCMs to examine how to apply BP simulation tools to *preliminary simulation* of time decomposition in MCMs, using the principle of *parsimony* (Pidd 1991).

BP and MCM

There are different definitions of a business process:

Processes are structured, measured sets of activities designed to produce a specified output for a particular customer or market (Davenport 1993).

A *business process* is a partially ordered set of Enterprise Activities which can be executed to realise a given objective of an enterprise or a part of an enterprise to achieve some desired end-result (Savén 2002, Koubarakis and Plexousakis 1999).

A *process system* is a set of business processes linked together to perform some Enterprise Function or Subfunction (Lencse and Muka 2006).

From the point of view of the *time decomposition problem,* the MCMs may be treated as a business *process system.*

There are some differences between the process systems and MCMs:

- Key activities, performed by a subsystem of an MCM, may be defined on a higher level then it is usually made in case of processes thus assigning resources for execution should be made differently.
- In the analysis, P-type and IT-type elements (or mixed type elements in case of expanded P elements) can perform the key activity.
- Elements (P and IT too) may be F, C and A elements which play special roles in a system description:
    - F elements (usually) play the role of activity except the feature they do not use resources in a classical way but they do need the presence of conditions C and A.

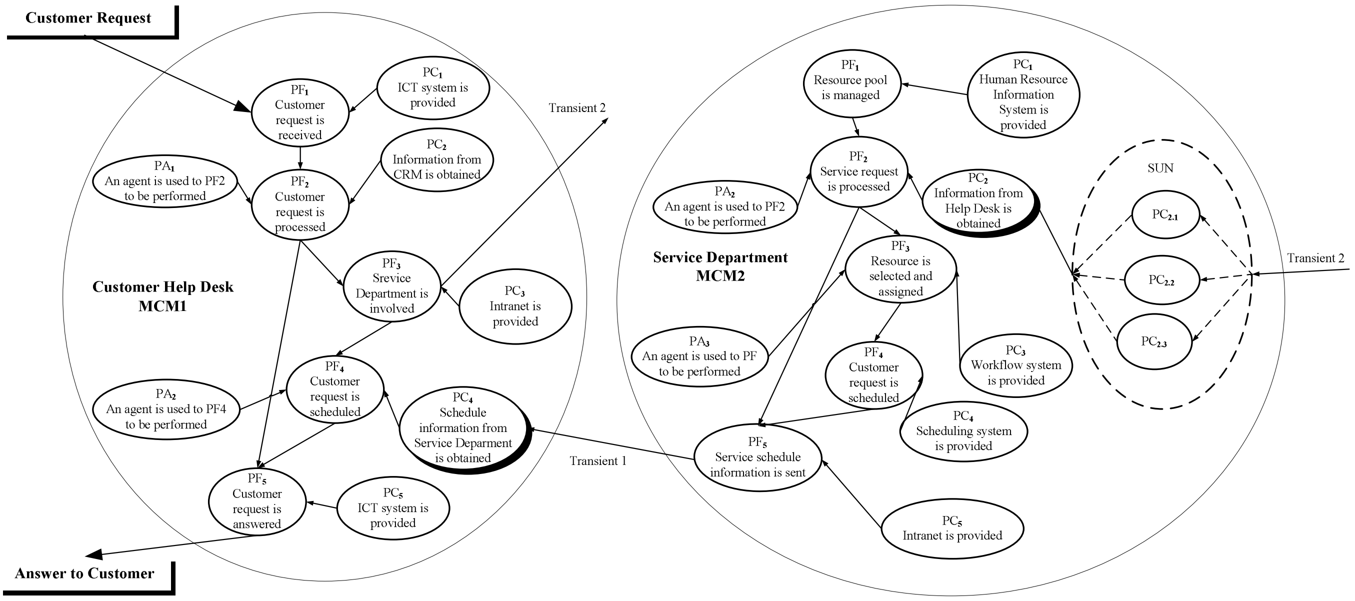

Figure 5 Conceptual Models MCM1 and MCM2 Synchronised Using Condition Elements

- C elements may be treated as "activities providing conditions". The traditional resource conditions of F elements may be connected to C elements.
- In a simple approach, A elements do not take part in time decomposition analysis.

When the execution of a task is in progress the execution of the F activities has an order defined by the routing rules, starting from the first element, the element where the task request enters the conceptual model.

The execution of a C element may become the part of the execution path when an F element has to wait for the condition provided by the C element, or the C element is on the execution path in case of synchronisation.

Process Simulation Elements to Preliminary Simulation of MCMs

Let us examine the elements of Process Simulation (PS) based on the features of *ImiFlow* simulator (Elassys 2007) from the point of view of Preliminary Simulation with MCMs in time decomposition (PSMCM):

PS: The *links* in PS are connections with a direction showing the performance order (time-precedence) of activities.

PS links may be *internal links* (connecting the activities of one process), *external links* (connecting processes forming a process system). The PS links are only logical connections with no capacity limit.

PSMCM: The links in PS are satisfactory to describe the logical connections in and between MCMs during the Preliminary Simulation of MCMs (PSMCM).

PS: *Performing an activity* in PS takes time which is described by a probability distribution (service time profile, activity time-consumption), because many factors are influencing the performance-time. In many cases it is enough to use normal distribution and the expected value of activity time.

T_{Cons}(activity, entity type) Time-Consumption of an Activity – expected value of time necessary to perform the given activity, that is necessary to process the entering entity-type.

PSMCM: The probability tools of activities in PS are adequate to examine for dynamic features of MCMs. Entity type may be used to describe task requests entering MCMs.

PS: The *entity-load* is produced by programmable *entity-generators*, the source of incoming entities. There may be different types of entities entering the process, which are produced by different sources. An entity of a given type has an *arrival profile* which is the arrival time distribution of the entity. The *destinations* of entities are the *exit points* of the process.

PSMCM: The entity-load of PS should model the information/entities traveling through MCMs.

PS: A routing decision may be made using different algorithms: *percentage distribution*, *entity-feature distribution*, *load-balancing distribution.*

There are some other elements influencing the generation and routing of entities:

Fork – Join, and *Split* elements make copies of an entity during the routing process.

PSMCM: We propose to build simple models to PSMCM thus these possibilities are more than satisfactory for this purpose.

PS: *Resource Capacity* has the following useful features:

- ***P_{ARes}*(type, month, week, day, time)** Resource Accessibility – the probability that a resource of the given type is accessible at the given point of time for a given activity
- ***R_n*** Expected value of accessible resource capacity of a given type for an activity ***n***

PSMCM: The resource capacity in the above described classical form should be taken into account as the resource capacity of C elements.

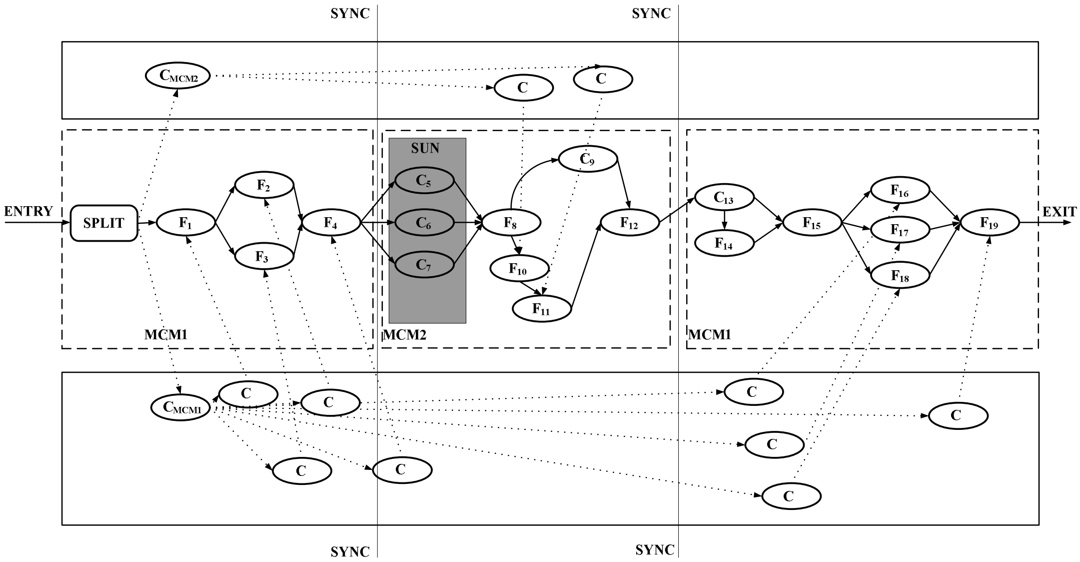

Figure 6 Example of an ENTRY-EXIT Execution Path

Process Simulation Model of an ENTRY-EXIT Path

Let us see an example. Figure 5 shows two cooperating departments, the Customer Help Desk (MCM1) and the Service Department (MCM2) of an enterprise. The Customer Request enters MCM1 through the PF_1 element.

Element PF_2 makes a decision to involve MCM2. The information from MCM1 reaches MCM2 through a set of SUN conditions of PC_2.

PC_2 is the condition "Information from Help Desk is obtained", the set of SUN conditions $PC_{2.1}$-$PC_{2.3}$ may be for example: to get information through e-mail, telephone call or direct database connection. The schedule information leaves MCM2 at PF_5 and returns to MCM1 through PC_4 of MCM1. The Answer to Customer is produced by PF_4 of MCM1.

The ENTRY-EXIT path is the next:

Customer Request- MCM1(PF_1-PF_2-PF_3)-transient edge-

- MCM2(PC_2(SUN:$PC_{2.1}$,$PC_{2.2}$,$PC_{2.3}$)-PF_2-PF_3-PF_4-PF_5)-

-transient edge-MCM1(PC_4-PF_4-PF_5)-Answer to Customer.

In Figure 6 we show the process-model like form of two synchronised systems. (The MCMs of the previous example after some transformations (for example after expanding PF_2, PF_4 elements of MCM1 and PF_2 element of MCM2) may get this structure.)

The ENTRY, through a *SPLIT* element, generates entities for the execution of F_1-F_{19}, SUN(C_5,C_6,C_7), C_9, C_{13} and to set up all the conditions for MCM1 (through C_{MCM1}) and for MCM2 (through C_{MCM2}). All the elements performing activities and their conditions are realised by process activities and by resource capacities connected to the conditions. Fork-Join elements or a percentage distribution may be used at F_1, C_{13} and F_{15}. The SUN element may implement all the three SUN condition possibilities simultaneously in concurrent mode, that is, for example the result produced by the condition element with the shortest execution time may be used each time, thus after the simulation, assessing the results, we may choose the best possibility.

A DECOMPOSITION EXAMPLE

In the case of the execution time decomposition, either we supposed that the execution of the subsystems does not overlap or we restructured the model using transformations to reach this situation in order to be able to approximate the total execution time of a task. However, in a real system the execution of the subsystems may overlap, that is the parallel assignment may be allowed.

Subsystem	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
Execution Time	3	4	2	3	7	6	7	8	5

Table 1 Estimated Execution Times of Subsystems

Figure 7 shows the execution order of the subsystems of an MCM. Table 1 shows the rounded expected value of the execution time of the subsystems in the units of *simulation.*

We have a certain number of processors (in a parallel computer or in a cluster of workstations; let us have 7 processors now) for the execution of our simulation model. *Execution time decomposition* helps us in the assignment of the subsystems to the processors. Figure 7 shows a possibility for the assignment with dotted line. Our goal was that all the processors have approximately equal load.

If we have only 3 processors, we do the assignment according to the continuous line in Figure 7.

Note that besides the execution times, we should also consider the frequency of information exchange of the subsystems during the decision of the assignment.

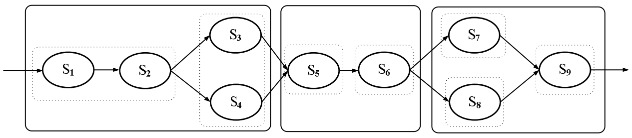

Figure 7 Subsystems of an MCM and Their Assignment to Processors

CONCLUSIONS

In this paper we addressed the efficiency of simulation: we described a time decomposition method in Modified Conceptual Models in order to choose the relevant systems to be modelled, to manage the appropriate model resolution and to support the decision about parallel and sequential simulation. After a brief summary of the essences of conceptual models, we showed a way how timing can be introduced into MCMs.

We described the necessary elements of time decomposition method in MCMs:

- we defined execution time to the elements of MCMs,
- we described transformations necessary to manage the resolution of MCMs,
- we emphasised the importance of appropriate level of resolution of our simulation models and explained how one can achieve it by expansion, grouping, integration and SUN transformations,
- we defined the conceptionally exhaustive SUN conditions as an MCM resolution-increasing transformation, in the way of showing different available possibilities,
- we described the execution-state and the execution-path models of the execution time in MCMs,
- we explained the importance of evaluation of execution time and system time from the point of view of system speed and resources.

In order to use process models to model MCMs, we analysed the similarities between BPs and MCMs and we examined the features of business process simulators as a preliminary simulation tool of the execution time decomposition in MCMs.

We introduced an example of the implementation of the execution-path approach in a process simulation environment.

In the end, we showed an example of using decomposition results to support sequential-parallel simulation decisions increasing the efficiency of simulation.

REFERENCES

Checkland, P. 1989. *Soft systems methodology* In Rational Analysis for a Problematic World, Edited by J. Rosenhead, John Wiley & Sons Ltd

Checkland, P., Scholes, J., 1990. *Soft Systems Methodology in Action* JohnWiley & Sons Ltd., Chichester UK

Curtis, G. 1989. *Business Information Systems* Addison-Wesley, Wokingham, UK.

Davenport, T. H. 1993. *Process innovation: Reengeneering work through information technology* Harvard Business School Press, Boston, Massachusetts

Elassys Consulting Ltd. 2007. *Iminet and Imiflow Systems* http://www.elassys.hu

Gregory, F. 1993. "Cause, Effect, Efficiency and Soft Systems Models*" J. Opl. Res. Soc.* Vol. 44, No. 4.

Jackson, M.C., Keys, P. 1984. "Towards a System of Systems Methodologies" *J. Opl. Res. Soc.* Vol. 35, No. 6.

Jackson, M. C., 1991. *Systems Methodology for the Management Sciences* Plenum Press, New York, London

Koubarakis, M., Plexousakis, D. 1999. "Business process modelling and design – a formal model and methodology" *BT Technol. J.* Vol. 17, No. 4.

Lencse, G., Muka, L. 2006. "Expanded Scope of Traffic-Flow Analysis: Entity Flow-Phase Analysis for Rapid Performance Evaluation of Enterprise Process Systems" *Proceedings of the 2006 European Simulation and Modelling Conference* (ESM'2006) (Toulouse, France, 2006. Oct. 23-25.) EUROSIS-ETI, 94-98.

Lencse, G., Muka, L. 2007. "Combination and Interworking of Four Modelling Methods for Infocommunications and Business Process Modelling" *Proceedings of the 5th Industrial Simulation Conference* (ISC'2007) (Delft, The Netherlands, 2007. Jun. 11-13.) EUROSIS-ETI, 350-354.

Muka, L., Lencse, G., 2006. "Developing a Meta-Methodology Supporting the Application of Parallel Simulation" *Proceedings of the 2006 European Simulation and Modelling Conference* (ESM'2006) (Toulouse, France, 2006. Oct. 23-25.) EUROSIS-ETI, 117-121.

Muka, L., Lencse, G. 2007. "Hard and Soft Approaches in a Simulation Meta-methodology" *Proceedings of the 5th Industrial Simulation Conference* (ISC'2007) (Delft, The Netherlands, 2007. Jun. 11-13.) EUROSIS-ETI, 17-22.

Pidd, M. 1991. "Computer simulation methods" In *Operations Research in Management*, Edited by Littlechild, S., and Shutler. M., Prentice Hall, UK.

Savén, R. 2002. "Process Modelling for Enterprise Integration: review and framework*"* Department of Production Economics, Linköping Institute of Technology, Linköping, Sweden

Sierhuis, M., Clancey, W. J. 2002. "Modeling and Simulating Work Practice: A Method for Work System Design" *IEEE Intelligent Systems* Vol. 17, No. 5. pp. 32-41.

Warmerdam, M., Bredveld, P. 2003. *A Holistic Approach to Delivering the Value of IT: Business Service Management* IDC White Paper, Framingham USA, http://www.idc.com

BIOGRAPHIES

GÁBOR LENCSE received his M.Sc. in electrical engineering and computer systems at the Technical University of Budapest in 1994 and his Ph.D. in 2001. The area of his research is (parallel) discrete-event simulation methodology. He is interested in the acceleration of the simulation of communication systems. Since 1997, he works for the Széchenyi István University in Győr. He teaches computer networks and networking protocols. Now, he is an Associate Professor. He is a founding member of the Multidisciplinary Doctoral School of Engineering, Modelling and Development of Infrastructural Systems at the Széchenyi István University.

Dr Lencse does R&D in the field of the simulation of communication systems for the Elassys Consulting Ltd. since 1998.

He works part time at the Budapest University of Technology and Economics (the former Technical University of Budapest) since 2005. There he teaches computer architectures.

LÁSZLÓ MUKA graduated in electrical engineering at the Technical University of Lvov in 1976. He got his special engineering degree in digital electronics at the Technical University of Budapest in 1981, and became a university level doctor in architectures of CAD systems in 1987. Dr Muka finished an MBA at Brunel University of London in 1996. Since 1996 he has been working in the area of simulation modelling of telecommunication systems, including human subsystems.

Dr. Muka is a regular invited lecturer in the topics of application of computer simulation for performance analysis of telecommunication systems at the Multidisciplinary Doctoral School of Engineering, Modelling and Development of Infrastructural Systems at the Széchenyi István University of Győr.