JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT
2008
14(3): 189–198

ANALYSIS OF TIMED PETRI NETS FOR REACHABILITY IN CONSTRUCTION APPLICATIONS

Khaled Nassar[1], **Albert Casavant**[2]

[1]*Dept of Construction and Architectural Engineering, American University in Cairo, 113 Kasr El Aini St., P.O. Box 2511, Cairo, 11511, Egypt. E-mail: knassar@sharjah.ac.ae*

[2]*Dept of Math and Computer Science, University of Maryland Eastern Shore, 1102 Kiah Hall, Princess Anne, MD 21853, USA. E-mail: acasavant@umes.edu*

Received 11 Sept 2007; accepted 4 July 2008

Abstract. Petri nets have been used for many years to model complex processes. Examples are software design, workflow management, data analysis, concurrent programming, reliability engineering, real-time computing systems, operating systems, distributed systems, and biological processes. Timed petri nets enable the study of timed process reachability, i. e. can a certain state in the process be reached, given an initial state of TPN representing a given process. This paper shows how the reachability of TPN can be utilized to analyze several aspects of construction projects. TPN are introduced and contrasted to Activity Cycle Diagrams and are also formalized to allow for the reachability problem to be assessed. The benefits and limitations of the analysis are presented through the classic earth-moving problem.

Keywords: Timed Petri nets, activity cycle diagrams, simulation, construction, reachability.

1. Introduction

In recent years, process modeling of construction activities has gained interest by researchers and practitioners alike. With the development of several modeling tools in the eighties and nineties (e.g. Cyclone (Halpin and Riggs 1992), Stroboscope (Martinez and Ioannou 1999) and Simphony (AbouRizk and Shi 1994; Zhang *et al.* 2007; Seppanen and Kenley 2005; Mallasi and Dawood 2004; Akinci *et al.* 2002), construction process modeling is becoming an increasingly easier task. The majority of the construction processes modeling tools, with very few exceptions, utilizes the Activity Cycle Diagram (ACD) paradigm or similar probabilistic techniques. ACDs are networks of circles and squares that represent idle resources, activities, and their precedence. These networks describe how construction operations are carried out as well as how and when the resources needed for these operations are used.

The traditional approaches to construction project modeling, such as ACDs, however, rely on a representation that lacks an underlying mathematical formalism. If one is only interested in simulation of construction projects, then the traditional approach is sufficient. However, if one takes an interest in analyzing and showing properties of construction projects, then a formalism having a firm mathematical basis and availability of sophisticated analytical techniques, such as Timed Petri nets is more appropriate. Timed Petri nets (TPN) provide a formal modeling tool that can be effectively used as a process modeling and analysis tool. TPN can systematically analyze construction process models in a formal manner by utilizing several inherent properties including reachability and reversibility, absence of deadlock, liveness, boundedness and mutual exclusion. This paper shows, how the reachability of TPN can be utilized to analyze several aspects of construction projects. TPN are introduced and contrasted to ACD and are also formalized to allow for the reachability problem to be assessed. The classic earth moving problem is used as an example and several key analysis questions are answered. It is noted that reachability of TPN are deterministic in nature, as opposed to the stochastic analysis of ACD. This will be useful for critical projects analysis, where probabilistic analysis may be not sufficient or practical. Furthermore, the analysis of construction projects, using the reachability of TPN, eliminates the need for over assignment of project resources, since resource needs can be deterministically determined and optimized. Using TPN to analyze construction projects allows for a more efficient construction process design, which will, in turn, result in cost savings. The next section provides a brief introduction to Petri nets.

2. Timed Petri nets

Petri nets is a general modelling language that can be used for a wide variety of purposes, and it was the first formulated theory for discrete parallel systems. The language is a generalization of automata theory, such that the concept of concurrently occurring events can be expressed. There are plenty of applications of Petri nets in a wide variety of areas. It is beyond the scope of this paper to provide an introduction to traditional non-timed Petri nets. There are several excellent introductions available

ISSN 1392–3730 print / ISSN 1822–3605 online
DOI: 10.3846/1392-3730.2008.14.16

(Peterson 1981; Reisig 1982). Instead we will focus on the structure and behaviour of Timed Petri nets (TPNs), i. e. general Petri nets with time added. The treatment of Timed Petri nets in this paper is based closely on the works of Berthomieu and Menasche (1983) and Berthomieu and Diaz (1991).

Petri nets have been used for many years to model complex processes. Examples are software design, workflow management, data analysis, concurrent programming, reliability engineering, computer architecture, computer networks, real-time computing systems, operating systems, distributed systems, hardware systems and biological processes. The bulk of this work has involved the used "standard" Petri nets, i. e. nets that model concurrency and parallelism, synchronization, mutual exclusion and causal dependency related to processes. The standard Petri net does not model the passage of time. One version of Petri nets with time included is a Timed Petri net (TPN). TPNs reduce the non-determinism in the duration of activities in Petri nets by associating a time interval with each transition of the net. Petri nets have been proposed as a tool for several applications in project management. Shih and Leung (1997) and Desrochers and Sanderson (1995) proposed Petri nets as a modelling tool for management systems. Maggot (1989) used Petri nets to evaluate the performance of concurrent processes, while Kumar and Ganesh (1998) and Holliday and Vernon (1987) used Petri nets for resource allocation in projects. This paper explores the application of TPNs to construction projects.

A Timed Petri net (**TPN**) is a tuple $< P$, T, B, F, M_0, $SIM >$, where P is a finite non-empty set of places; T is a finite nonempty set of transitions t_i, which can be viewed as an ordered set $\{t_1, t_2 ..., t_i, ...\}$; $B: T \times P \rightarrow \mathbf{N}^P$ is the backward incidence function, where $\mathbf{N}^P$ is a vector of positive integers of size P; $F: T \times P \rightarrow \mathbf{N}^P$ is the forward incidence function; $M_0 : P \rightarrow \mathbf{N}^P$ is initial marking function; $SIM : T \rightarrow Q^+ \times (Q^+ \cup \infty)$ is the static interval mapping function, where Q^+ is the set of positive rational numbers.

Here, $F(t_i)$ describes, how many tokens (markers in places) are placed in the output places of a transition t_i, when that transition fires. Conversely, the $B(t_i)$ describes the number of tokens removed from the places of the transition when, it fires. $M_0(p_i)$ is the number of tokens at place p_i in initial marking M_0. $SIM(t_i)$ returns $[\alpha_i^s, \beta_i^s]$ describing the *static firing interval* for transition t_i, where the left bound α_i^s is the static earliest firing time (EFT), and β_i^s is the static latest firing time (LFT). Times α_i^s and β_i^s are both relative to the moment, that t_i is enabled. EFT and LFT are described as static because one or both may decrease as the result of the firing of other enabled transitions and the passage of time, when they become known as the *dynamic firing interval*. The mathematical formalism of TPN allows for several kinds of analyses that explore the behaviour of various models. These are shown in Table 1. In the next section, we start modeling the basic earth-moving problem using Petri nets.

3. The classic earthmoving problem model using Timed Petri nets

The classic earth-moving problem is the movement of a certain amount of soil by a number of trucks and a wheel loader. Fig. 1 shows the Activity Cycle Diagram for a sample application of the earth-moving problem. As was mentioned earlier, ACDs are networks of circles and squares that represent idle resources, activities, and their precedence. The squares represent activities, while the circles show resources. The the model activities are in Table 2.

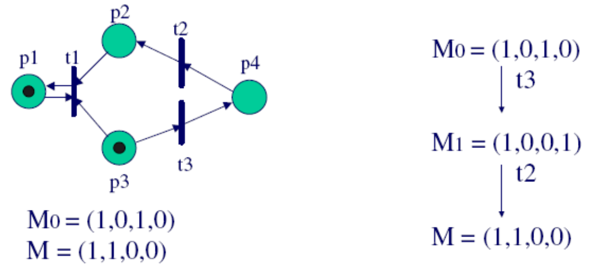
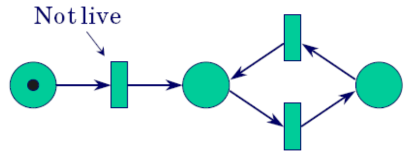
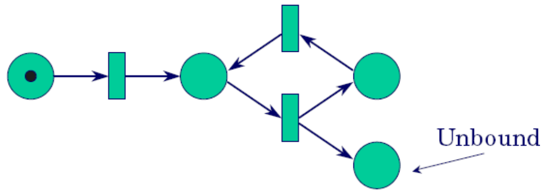
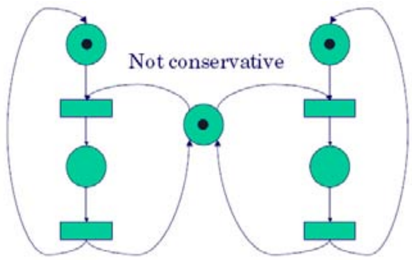
The ACD of Fig. 1, for example, is a graphical representation of the information in Table 2. The rectangles represent activities (resources collaborating to achieve a task), the circles represent queues (idle resources), and the links between them represent the flow of resources. ACDs of this type are typically used to express the main concepts of a simulation model. Loaders are used to fill a number of trucks on the cut side and then haul the soil to the fill side. The loading time is required for the wheel loader to completely fill a truck. There is no restriction for dumping, which can take place immediately after hauling. This model will be amended in the next section to demonstrate the applicability of reachability analysis.

The corresponding TPN model for the operation is shown in Fig. 2. The figure shows a model for the soil basic movement by trucks and a wheel loader. Three trucks are shown, one being loaded by the wheel loader, one in hauling to the dump point, and one returning from dumping. It is assumed that there is an infinite pile of soil and that operations have just begun, so at this moment elapsed time is zero and each process, i.e. loading, hauling or returning, has just begun. M_0 is TrkQueue, LoadToHaul, DumpToReturn. There are 3 enabled transitions: WhlLoader, Haul and Return, so the I_0 vector corresponding to M_0 has 3 components ([7, 9], [10, 13], [12, 15]). Each interval is the static firing interval associated with the enabled transitions WhlLdr, Haul and Return, respectively. Now the basic earthmoving problem is presented, the reachability of TPNs is presented next.

4. Reachability of TPN

Timed Petri nets enable to study the timed process reachability, i.e. can a certain state in the process be reached, given an initial state and a TPN representing a process. In some sense, this is the basic problem that simulation is attempting to solve. Indeed, there have been extensive efforts devoted to establishing simulation frameworks for construction projects. One of the big disadvantages of simulation is that the full range of solutions cannot be

Table 1. Behavioural properties of PN

Properties	Meaning	Example
Reachability	Marking M is reachable from marking $M0$, if there exists a sequence of firings transforming $M0$ to M. The reachability problem is decidable.	
Liveliness	From any marking any transition can become fireable. Liveness implies deadlock freedom, and not vice-versa.	
Boundedness	The number of tokens in any place cannot grow, indefinitely (1-bounded also called *safe*).	
Schedulability	Analysis for a given PN answers the question a set of firing sequences that can be infinitely repeated exists in the reachability space of the net.	
Conservation	The total number of tokens in the net is constant.	

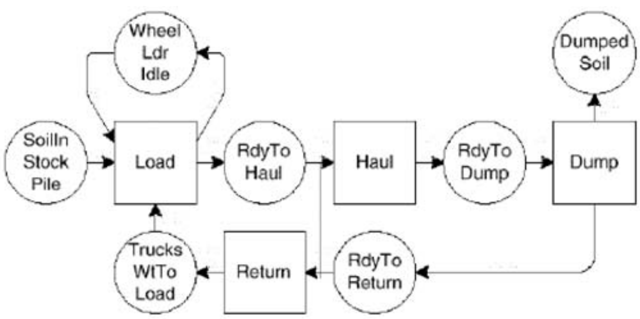


Fig. 1. The activity cycle diagram of the basic earthmoving problem

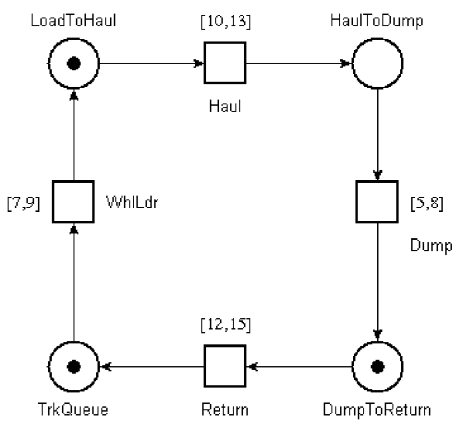


Fig. 2. The Petri net formulation of the basic earthmoving problem

Table 2. Activities of the basic earthmoving problem

Conditions needed to start	Activity	Outcome of activity
Wheel loader idle at source. Empty truck waiting to load. Enough soil in stockpile	Load	Wheel loader idle at source. Loaded truck ready to haul
Loaded truck ready to haul	Haul	Loaded truck ready to dump
Loaded truck ready to dump	Dump	Dumped soil. Empty truck ready to return
Empty truck ready to return	Return	Empty truck waiting to load

determined. Simulation bases changes of state in the process on probability. Using simulation, there is no guarantee that all possible states of the system have been exposed. Indeed, there is no guarantee that a certain percentage of possible states have been exposed. If it is desired to know, whether a certain state of a process can ever occur, then reachability analysis fits the bill. This is especially important in situations, where reaching a certain state involves a significant safety risk or produces a situation, where gridlock can occur, i. e. too many resources in the same place at the same time.

Reachability analysis has an advantage that all possible states of the process are exposed based on the range of firing times of transitions. Hence, one can say unequivocally, whether it is possible or not for a system to reach a certain state. Thus, the confidence level in the result of reachability analysis can be much higher than that for simulation. The goal of this paper is to demonstrate that reachability analysis is useful for solving problems in construction, and can answer questions about construction project models that cannot be answered by simulation.

A solution of the *reachability problem* is answering the question whether or not a given marking belongs to $R(M_0)$. Similarly, the *boundedness problem* is whether or not all markings in $R(M_0)$ are bounded, i. e. have a finite number of tokens. For all markings in $R(M_0)$ and all places p, $M(p) \le k$ for some k in $\mathbf{N}$, the set of natural numbers. A TPN is said to be *T-bounded* if there exists a natural number k such that none of its transitions may be enabled more than k times simultaneously by any reachable marking i.e. for all $\mathbf{t}(i)$ in T there exists p in P such that: $M(p) < (k+1)\ B(\mathrm{t}(i), p)$. When $k = 1$, the TPN is said to be *T-safe*. To solve the reachability problem of TPN, we need to construct what is called a reachability graph. Basically, the reachability graph describes different states of a TPN.

In a traditional non-timed Petri net, the state of the net consists solely of a marking, *M*. In TPNs, where time is part of the state, the state of the net is a pair (M, I), where M is the marking and I is a *firing interval vector.* This vector associates with each transition enabled by M, the time interval in which the transition is allowed to fire. The number of entries in the firing interval vector is the number of transitions, enabled in M. The number of entries in I will vary as markings change and the set of enabled transitions changes. Each entry in the firing interval vector gives the range of possible firing times for its associated enabled transition in marking M. A state S is entered at an absolute time τ measured from the beginning of $S_0 = (M_0, I_0)$.

Now consider the classic earth moving problem in Fig. 2. Boundedness is decidable with coverability tree. For bounded nets the Coverability Tree is called Reachability Tree since it contains all possible reachable markings. The algorithm for creating a coverability tree builds a (finite) tree representation of the markings. This algorithm is known as the Karp-Miller algorithm (Fig. 3).

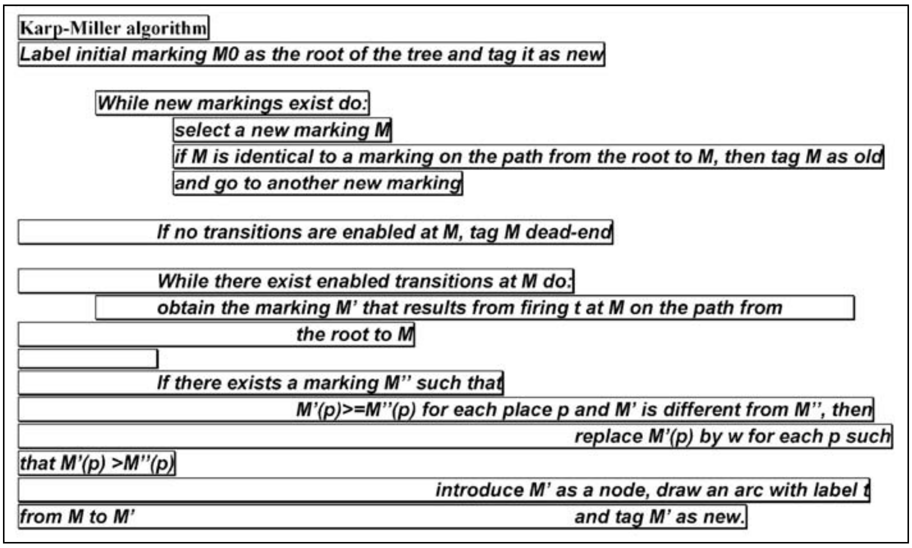


Fig. 3. The Karp-Miller algorithm

For the soil movement above, we use a tool called Tina (Berthomieu *et al.* 2004; Berthomieu 2008], developed by Bernard Berthomieu of LAAS-CNRS in Toulouse (France) to generate the reachability graph (Fig. 4). The marking reachability graph, shown in Fig. 5, has 26 states. Using this information, one can start to decide, whether certain conditions can be reached or not. Sample applications are shown in the next section.

5. Reachability applications to the developed earth-moving problem

In this section we examine the use of reachability analysis to answer several practical questions concerning each soil movement model. The basic earth moving problem presented above is not revised to include a restrictive road that trucks are required to use both to and from the dumping destination. Fig. 6 shows a sample schematic of the problem. We assume that the road narrows down at a treacherous bridge/curve, where trucks are permitted to travel in one direction only. While trucks are in the curve going in a certain direction, no trucks are permitted for safety reasons to enter the curve in the opposite direction. The ACD for the new developed model is shown in Fig. 7. This model has 4 trucks, initially waiting to be loaded by a wheel loader.

In this model the haul road has been broken into 3 parts. Haul now consists of a Haul1, HaulCrv, and Haul2. Similarly, Return now consists of Return1, Return Crv, and Return2. HaulCrv and ReturnCrv are for crossing a narrow, birdge segment. This segment will eventually allow traffic in only one direction with no passing (single file loaded or empty traffic, but not both at the same time).

The restrictive road is modeled by 4 places, InHaulCrv, InRetCrv, HP, RP and 4 transitions, EnterHaulCrv, RestHaulCrv, EnterRetCrv, and RestRetCrv. These places and transitions model entry and presence on the road in one direction only, a concept known as *mutual exclusion*. One direction of the road is modeled with EnterHaulCrv, InHaulCrv and RestHaulCrv. The other direction is modeled with EnterRetCrv, InRetCrv and RestRetCrv. If one truck is waiting to enter the road in place WaitHaulCrv and a second truck is waiting to enter in place WaitRetCrv, only one may enter. Suppose that the truck in WaitHaulCrv enters. One token will be removed from place RP. Since 4 tokens are required to fire EnterRetCrv, no truck may simultaneously enter the curve in the return direction. The road in the haul direction is now open for more trucks to enter that direction. Each will take another token from RP. As trucks in the haul

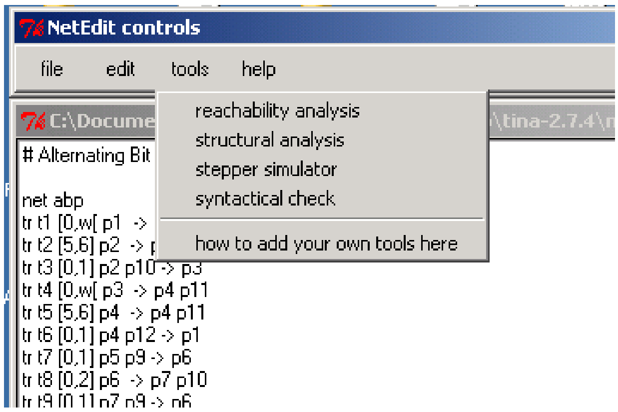


Fig. 4. Tina's interface

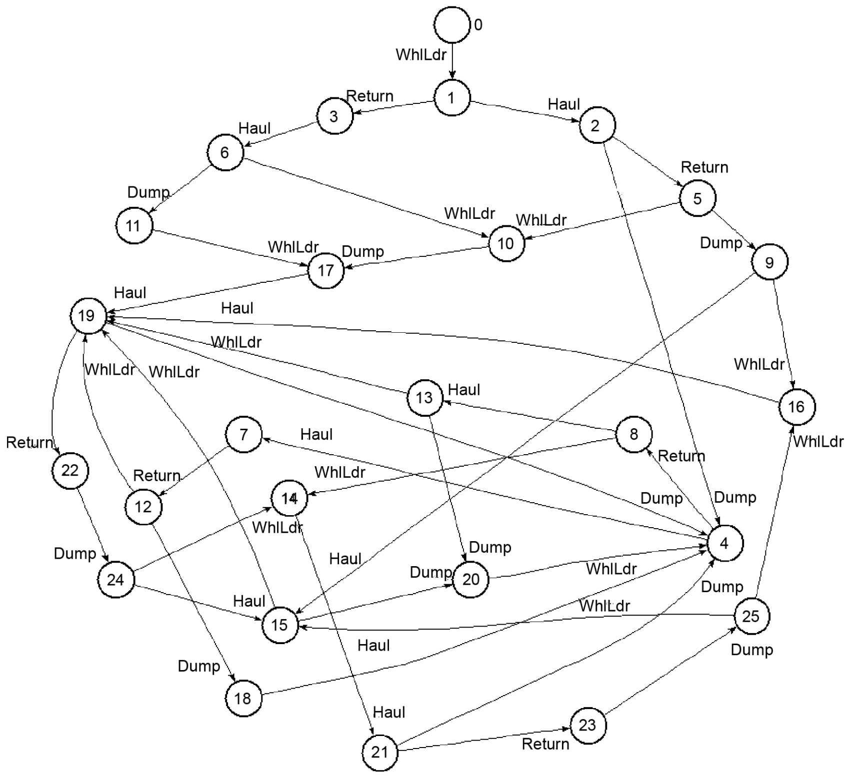


Fig. 5. The marking reachability graph

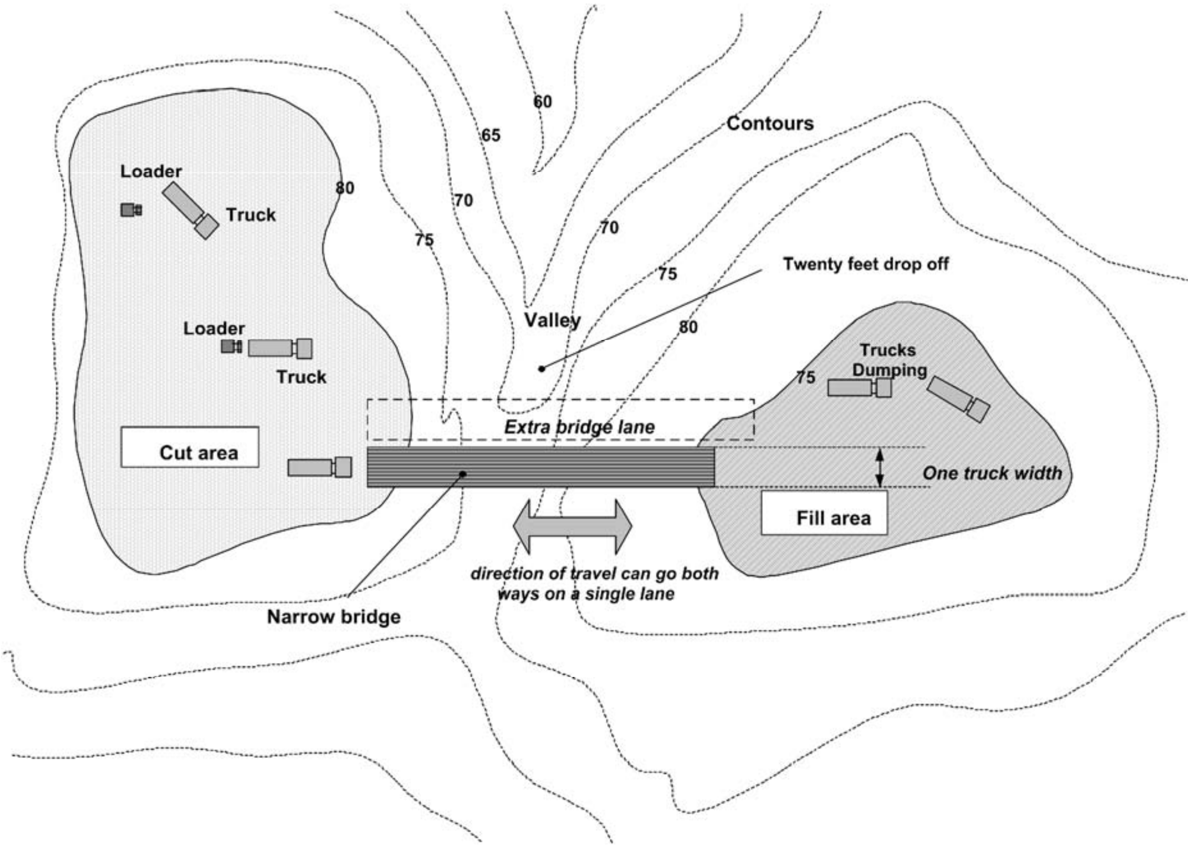


Fig. 6. The developed earth-moving problem schemetic

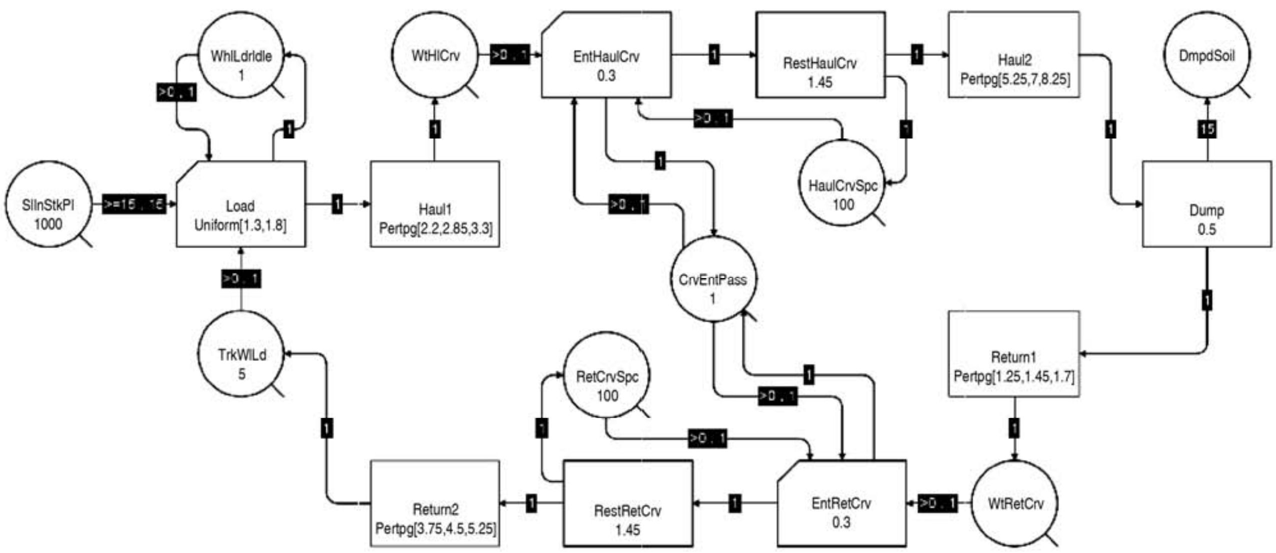


Fig. 7. The ACD of the earth-moving problem

direction leave the curve by firing transition RestHaulCrv, they return tokens to RP. When all the tokens have been returned to RP, EnterRetCrv could fire, if there are any trucks waiting in WaitRetCrv. If a truck fires EnterRetCrv, then a token is removed from HP, thereby blocking entry to the curve in the haul direction. The haul and return directions on the road have mirror images. If two trucks are able to simultaneously fire EnterHaulCrv and EnterRetCrv, following the semantics of reachability analysis, *both* firings are modeled. Each firing will generate a branch in the reachability tree. Note that this does not mean that mutual exclusion has been violated; it means only, that at a particular moment, a truck can enter either direction, but once a direction decision has been made, mutual exclusion is in effect and trucks may enter and travel in one direction only.

Now we are in a position to explore the use of reachability analysis to answer some questions about the soil movement example. An interesting question is: (Question 1) *What is the minimum number of trucks that are needed to keep the wheel loader continuously busy?* The wheel loader will be busy if there is always a truck in TrkQueue enabling the WhlLdr transition in Fig. 7. Since we have no idea at this point, how many trucks will be needed, we initialize TrkQueue with 100 trucks (replacing the 4 truck initialization in Fig. 8), We estimate that 100 trucks is many more than will be needed. Unused trucks will remain in the queue. The additional unused trucks do not generate any additional state classes thereby causing time and memory efficiency problems. To see this, suppose that 8 trucks are needed for hauling, and we have one extra truck that always remains in the queue. If we initialize the queue with 100 trucks rather than 9, each state class will end up having 91 extra tokens rather than one extra token. Since TrkQueue will always be occupied, it will appear in all state classes with some multiplier. Whether the multiplier is one or 91, it does not affect the number of state classes.

Using Tina to do reachability analysis of this example, we search the output file for the minimum number of tokens in TrkQueue. Thoughout the remainder of this paper, answering reachability questions are solved by searching the text file generated by Tina. This can be done conveniently using regular expressions with a text editor that has support for this search method. Discussion of regular expression use is beyond the scope of this paper. We refer the interested reader to (Martinez 1999). The construction and use of regular expressions can be automated, as well.

Returning to the question, we find the minimum number of trucks remaining in TrkQueue to be 95, indicating that 5 trucks can be in places other than in TrkQueue. Thus, with 5 trucks, the truck queue can be empty at times. Thus 6 trucks are needed to keep the wheel loader continuously busy. Recall from a previous discussion, that 6 trucks are needed with some probability. We do not attempt in this paper to quantify the probability. To further verify this result, we run the same problem with 5, 6 or 7 trucks in WhlLdr. We see that the 5 truck example generates 2184 classes, whereas the 6 and 7 truck examples both generate 2481 classes. Thus, more classes are generated going from 5 to 6 trucks, indicating that adding another truck generates new states, whereas going from 6 trucks to 7 trucks does not generate any new states. The 7th truck continues to sit in TrkQueue and is never used. Similar techniques can be applied to answer a wide array of questions on the process (Table 3). There are several advantages of Timed Petri nets over ACDs, and these are explained below.

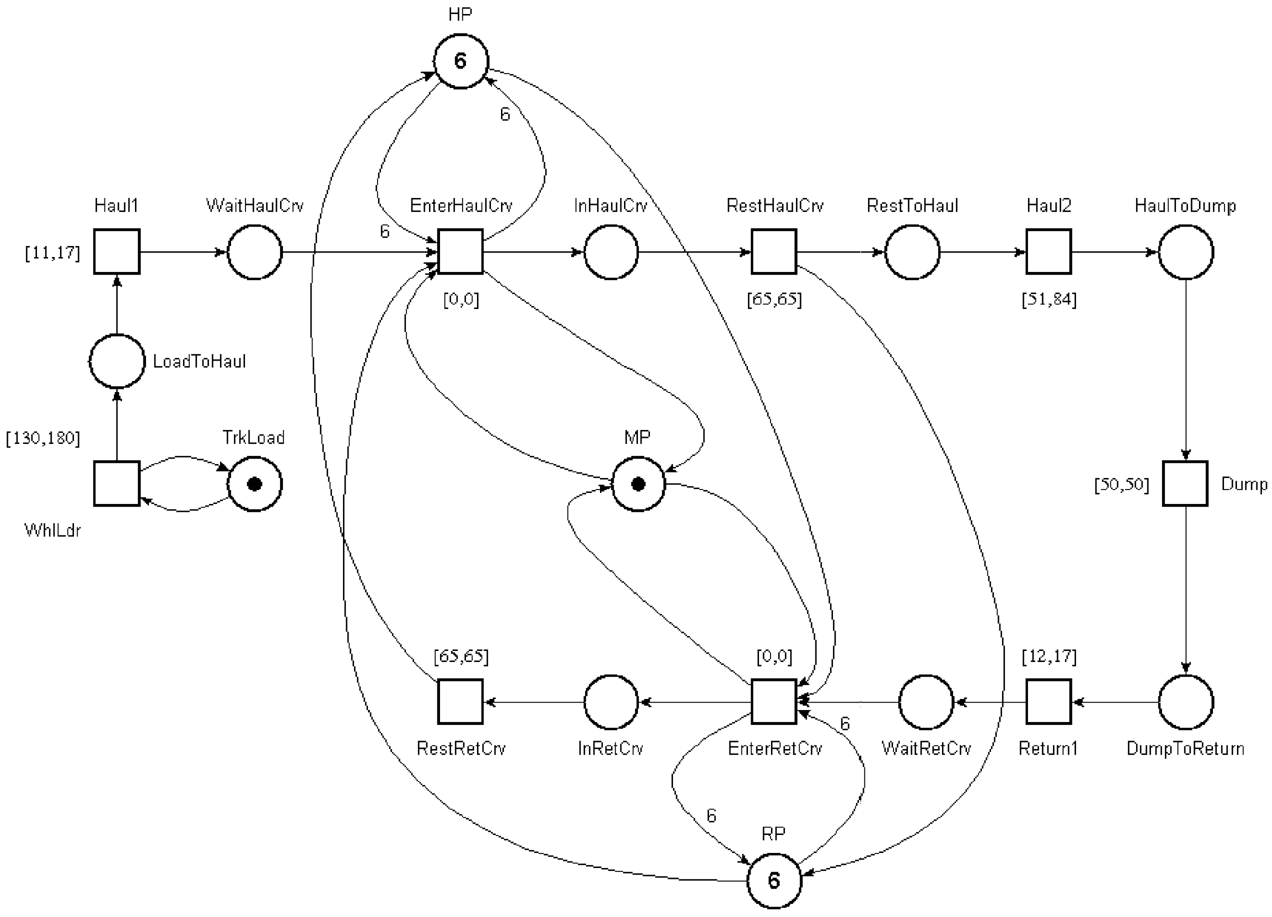


Fig. 8. The TPN model for the earth-moving problem

Table 3. Results of reachability analyses for certain conditions of the developed earth-moving porblem

No	*Question*	Answer
1	*What is the minimum number of trucks needed to keep the wheel loader continuously busy?*	The minimum number of trucks remaining in TrkQueue to be 95, indicating that 5 trucks can be in places other than in TrkQueue. Thus, with 5 trucks, the truck queue can be empty at times. Thus 6 trucks are needed to keep the wheel loader continuously busy.
2	*What is the penalty (in terms of trucks needed to keep the wheel loader continuously busy) to maintain one way travel in the curve?*	The delays waiting to enter the curve require 3 more trucks, or 100 % more transportation resource.
3	*With a given number of trucks, what is the maximum number of loads that will keep the wheel loader busy?*	With 14 loads and 5 trucks, the wheel loader will be busy all the time.
4	*With a certain number of loads to deliver, what is the minimum number of trucks needed to keep the wheel loader busy?*	5 trucks
5	*Without considering the number of loads to be delivered or the number of trucks available, what is the maximum number of trucks that can simultaneously be in the curve?*	The maximum number of trucks that can be in the curve simultaneously is 3.
6	*Without considering the number of loads to be delivered or the number of trucks available, what is the maximum number of trucks that can simultaneously be in the curve?*	Running the example with 6 followed by 5 followed by 4 trucks, we learn that 6 and 5 trucks produce 3 trucks in the curve, whereas 4 trucks produces only 2 trucks in the curve.
7	*With a certain number of trucks in use, what is the minimum number of loads that cause a given number of trucks to be simultaneously in the curve?*	3 loads or more are required to put 2 trucks in the curve using 5 trucks total.
8	*With a certain number of loads to deliver, what is the minimum number of trucks needed to cause a given number of trucks to be simultaneously in the curve?*	This turns out to be 15 loads. Thus, carrying one less load will limit the number of trucks in the curve to 2 maximum.
9	*What change in the number of trucks in use is required to keep the wheel loader busy, if the number of trucks in the curve is limited for safety reasons?*	Running Tina and examining the output we find that a maximum of 4 trucks in use will generate 2 trucks in the curve, but not 5 trucks in use.
10	*How many extra trucks are needed to keep the wheel loader busy when a road closure increases the distance to and from the curve?*	It requires 7 trucks to keep the wheel loader busy, an increase of 1.
11	*Can we get the same effect as adding trucks by decreasing loads?*	The answer is "yes". It turns out that if we limit the number of trucks to 6, we can complete 14 loads and still have trucks waiting.

6. Advantages of the proposed Petri net approach

The advantage of TPN reachability analysis lies in:

1. The ability to make deterministic conclusions about the construction project (versus the probabilistic analysis of activity cycle diagrams and its tools such as SIMPHONY, STROBOSCOPE and CYCLONE). Deterministic conclusions on the construction project is essential to several construction applications that have already been modeled by activity cycle diagrams, such as earth moving, tunneling, grading, paving and several others. Some practitioners have resisted the use of process modeling techniques in the field citing their lack of determinism. For example, contractors want to be as sure as possible as to the resources, they will need on a job. By removing the variability of the analysis and considering a deterministic analysis technique, such as TPN, one can attain a highest level of confidence in the result. Unlike activity cycle diagrams, TPN have a firm mathematical basis that allows a deterministic analysis of construction systems.

2. Fail-safe modelling. When simulating some construction processes, it is important to attain a level of confidence that certain conditions will not arise. For example, when modelling a concrete manufacturing facility, it may be enough to determine that there is a small chance that given a certain number of trucks, the hopper will be idle. This information can be used to calculate and optimize the overall cost of production by considering the cost of extra trucks, versus the idle cost of the hopper. In order to assess this, standard simulation tools can be used that employ ACDs. In other situations it is important to be certain that particular situations will not arise and not just attain a level of confidence. Consider, for example, the developed earthmoving problem with a narrow bridge. It may be imperative for structural safety reasons to assess the maximum number of trucks on the bridge at any time.

3. It was concluded from previous research that there is no consensus on the basic constructs needed for process modelling. It is argued, among several things that for reaching sufficient modelling power, and the depth of formal basic constructs (Sawhney *et al.* 1999). The traditional process modelling tools lack an underlying mathematical formalism that allows for systematic analysis of the properties of the construction process being modelled (Wakefield and Sears 1997). A formalism having a firm mathematical basis with sophisticated analytical techniques is needed for fully analysis of construction process models. TPNs provide that formalism.

4. The rules and logistics of using the modelling elements are also very simple, thus making the learning period very short, compared with the other process modelling tools available. The models obtained are easy to explain to both experts and non-experts.

7. Limitations of the analysis

There are two main limitations to the analysis presented here:

1. Unfortunately, the application of reachability of tree analysis method is greatly limited by the fact that the reachability tree of a Petri net may be an infinite tree for a given initial state or marking (Lu *et al.* 2003). Efforts have been made, however, to find some finite representations for reachability trees (Lu *et al.* 2003).

2. Although the results in Table 3 do not portend any computational difficulties performing reachability analysis with Petri nets, it is well known that this problem suffers from *state space explosion*. Problems of the size of the soil movement example and significantly larger can be solved without this becoming a problem. However, large examples may not be tractable, because the number of state classes becomes prohibitively large. Tina deals with the problem by collapsing equivalent state classes into a single state class, and by ignoring the domain part of the state class during construction of the marking reachability graph. More sophisticated techniques have been developed. These include the application of a region graph method to calculate the reachability graph (Gardey *et al.* 2004; Gardey *et al.* 2006]. This technique has been implemented in a tool, called Romeo, which is faster than Tina for constructing state space classes and allows testing on the fly for the reachability of a given marking (RTS Software 2008).

8. Conclusions

In this paper, Timed Transition Petri nets were presented as a formal modeling tool that can be employed as a construction process modeling and analysis tool. We solved the reachability problem for construction projects using a Timed Transition Petri net (TTPN), also known as a Time Petri net. The fireablity rules for transition and state were also presented and formalized. We demonstrated, how the reachability technique can be used to deterministically analyze typical construction activities. The reachability technique was applied to the traditional earth moving process and we used a computerized tool (TINA) to automate the process using regular expressions.

It was shown that solving reachability problems for traditional construction process models is practical and robust using standard hardware and freeware. Furthermore, analysis of construction projects using the reachability of TPN eliminates the need for over assignment of project resources, since resource needs can be deterministically determined and optimized. This is in contrast to the probabilistic ACD, where there may be extreme unforeseen situations mandating assigning more resources than needed. Using TPN to analyze construction projects allows for a more efficient construction process design, which will, in turn, result in cost savings in terms of the resources necessary for the projects being analyzed.

References

AbouRizk, S.; Shi, J. 1994. Automated construction simulation optimization, *Journal of Construction Engineering and Management* 120(2): 374–385.

Akinci, B.; Fischer, M.; Levitt, R.; Carlson, R. 2002. Formalization and automation of time-space conflict analysis, *Journal of Computing in Civil Engineering* 16(2): 124–134.

Berthomieu, B. 2008. *TINA: Timed Petri net analyzer* [cited July 2008]. Available from Internet: <http://www.laas.fr/tina/>.

Berthomieu, B.; Diaz, M. 1991. Modeling and verification of time dependent systems using Time Petri nets, *IEEE Transactions on Software Engineering* 17(3): 259–273.

Berthomieu, B.; Menasche, M. 1983. An enumerative approach for analyzing time Petri nets, in *Proc of the 9th IFIP World Congress "Information Processing 83" September 1983*, Paris, France. Vol. 9, Paris, 41–46.

Berthomieu, B; Ribet, P.-O.; Vernadat, F. 2004. The tool TINA – construction of abstract state spaces for Petri nets and time Petri nets, *International Journal of Production Research* 42(14): 2741–2756.

Desrochers, K. J.; Sanderson, A. C. 1995. Task planning and project management using Petri nets, in *Proc of IEEE International Symposium on Assembly and Task Planning (ISATP'95), August 10–11, 1995*, Pittsburg, USA, 254–271.

Gardey, G.; Roux, O. H.; Roux, O. F. 2004. Using zone graph method for computing the state space of a time Petri net, *Lecture Notes in Computer Science* 2791/2004: 246–259.

Gardey, G.; Roux, O. H.; Roux, O. F. 2006. State space computation and analysis of Time Petri nets, *Theory and Practice of Logic Programming* 6(3): 301–320.

Halpin, D. W.; Riggs, L. S. 1992. *Planning and analysis of construction operations*. New York: John Wiley & Sons.

Holliday, M. A.; Vernon, M. K. 1987. Generalized time Petri net model for performance analysis, *IEEE Transactions on Software Engineering* SE–13(12): 1297–1310.

Kumar, V. K. A.; Ganesh, L. S. 1998. Use of Petri nets for resource allocation in projects, *IEEE Transactions on Engineering Management* 45(1): 49–56.

Lu, M.; Anson, M.; Tang, S. L.; Ying, Y. C. 2003. HKCONSIM: A practical simulation solution to planning concrete plant operations in Hong Kong, *Journal of Construction Engineering and Management* 129(5): 547–554.

Magott, J. 1989. Combined generalized stochastic Petri nets and PERT networks for the performance evaluation of concurrent processes, in *Proc of the 3rd IEEE International Workshop on Petri nets and Performance Models, December 11–13, 1989*, Kyoto, Japan, 249–256.

Mallasi, Z.; Dawood, N. 2004. Workspace competition: assignment, and quantification utilizing 4D visualization tools, in *Proc of Conference on Construction Application of Virtual Reality (CONVR 2004), 14–15 September, 2004*, Lisbon, Portugal, 13–22.

Martinez, J. C. 1998. EZStrobe – general-purpose simulation system based on activity cycle diagrams, in *Proc of the 1998 Winter Simulation Conference, December 13–16, 1998*, Washington, USA, 341–348.

Martinez, J. C.; Ioannou, P. G. 1999. General-purpose systems for effective construction simulation, *Journal of Construstion Engineering and Management* 125(4): 265–276.

Peterson, J. L. 1981. *Petri net theory and the modeling of systems*. New Jersey: Prentice-Hall.

Reisig, W. 1982. *Petri nets: an introduction*. New York: Springer-Verlag.

RTS Software. 2008. Real Time Systems Software [cited July 2008]. Available on Internet: <http://www.rts-software.org/>.

Sawhney, A.; Abudayyeh, O.; Chaitavatputtiporn, T. 1999. Modeling and analysis of concrete production plant using Petri nets, *Journal of Computing in Civil Engineering* 13(3): 178–186.

Seppanen, O.; Kenley, R. 2005. Performance measurement using location-based status data, in *Proc of International Group for Lean Construction (IGLC)–13, July 2005*, Sydney, Australia, 263–269.

Shih, H. M.; Leung, C. K. H. 1997. Management Petri net a modelling tool for management systems, *International Journal of Production Research* 35(6): 1665–1680.

Wakefield, R. R.; Sears, G. A. 1997. Petri nets for simulation and modeling of construction systems, *Journal of Construction Engineering and Management* 123(2): 105–112.

Zhang, C.; Hammad, A.; Zayed, T. M.; Wainer, G.; Pang, H. 2007. Cell-based representation and analysis of spatial resources in construction simulation, *Automation in Construction* 16(4): 436–448.

PETRI LAIKO TINKLŲ PASIEKIAMUMO TAIKYMO STATYBOJE ANALIZĖ

K. Nassar, A. Casavant

Santrauka

Petri laiko tinklai jau daugelį metų yra taikomi sudėtingiems procesams modeliuoti. Jų pavyzdžiai yra programinės įrangos kūrimas, organizacinių procesų valdymas, duomenų analizė, vientisas planavimas, patikima inžinerija, kompiuterinės sistemos, operacinės sistemos, išskirstytos sistemos ir biologiniai procesai. Petri laiko tinklai leidžia tirti apibrėžtų procesų įvykdymo pasiekiamumą per nustatytą laiką, t. y. ar tam tikra proceso būklė gali būti pasiekta turint Petri laiko tinklų pradinę būseną, aprašančią procesą. Šiame straipsnyje parodoma, kaip Petri laiko tinklų pasiekiamumas gali būti pritaikytas statybos projektams analizuoti keliais aspektais. Apibūdinami Petri laiko tinklai yra gretinami su veiksmų ciklinėmis diagramomis ir suformuoti taip, kad būtų galima įvertinti pasiekiamumo problemas. Taip pat yra pristatomi analizės privalumai ir trūkumai nagrinėjant klasikinį žemės darbų uždavinį.

Reikšminiai žodžiai: Petri laiko tinklai, ciklinės veiksmų diagramos, modeliavimas, statyba, pasiekiamumas.

Khaled NASSAR. Associate Professor of Construction and Architectural Engineering at the American university in Cairo. He was on the faculty of Bradley University in Peoria, as well as the University System of Maryland, where he taught and conducted research project for the Illinois Department of Transportation and Caterpillar Inc, as well as NSF. He received a PhD from Virginia Tech. He worked for several years in the building industry. Dr. Nassar has also taught in a number of industry workshops, and published several journal papers. His research interest includes computer applications in design and construction as well as simulation and visualization.

Albert CASAVANT. Assistant Professor of Math and Computer Science at the University of Maryland Eastern Shore. He previously worked as a researcher in Electronic CAD for GE Corporate R&D and NEC C&C Research Labs. His current interests include timing analysis, Petri net reachability, and probability analysis techniques using Petri nets.