ISSN 1392 – 1215 ELEKTRONICS AND ELECTRICAL ENGINEERING 2006. Nr. 5(69)
ELEKTRONIKA IR ELEKTROTECHNIKA

T121 SIGNAL TECHNOLOGY
SIGNALŲ TECHNOLOGIJA

Global Variables in Colored Petri Nets

S. Bartkevičius
Department of Theoretical Electrical Engineering, Kaunas University of Technology
Studentų str. 48,LT-51367 Kaunas, Lithuania; phone: +370 37 300253, e-mail: Stanislovas.Bartkevicius@ktu.lt

R. Kragnys, K. Šarkauskas
Department of Control Engineering, Kaunas University of Technology
Studentų str. 48, 51367 Kaunas, Lithuania; phone: +370 37 300276, e-mail: Kastytis.Sarkauskas@ktu.lt

Introduction

There is necessity to use special program tools for analysis and design of control systems, especially if they are large, non-linear or stochastic. These tools or methods can radically simplify and make this job easier. Simulation is one of the analysis methods. Petri nets are often used to simulate systems based on event logics. These nets allow simulation of parallel and concurrent processes.

Traditionally properties of Petri nets, such as safeness, boundness, conservation, livness, reachability, coverability and *etc* are accented. These properties can be formally analyzed only of closed Petri nets and are unapproachable if a net is open [1]. If Petri net is used to simulate a control program (program of a controller), it perforce is open, because the program must interfere with an object of control, react into disturbances. This does not exclude usage of Petri nets on the purpose, because many of before mentioned properties may be checked as a result of simulation and parallelism and possibility of simulation of concurrence looks very attractive.

There are many papers analyzing usage of Petri nets to simulate control systems [2, 3, 4], but universal mean does not exist, therefore systems are often met, which can not be simulated using traditional Petri nets. On the other hand, it is very important to have possibility to simplify models of complicated control systems. The authors of this paper propose to let use (with some restrictions) the global variables in Petri nets applied for simulation of control systems.

Presence of global variables means hidden links in a net and disables before mentioned analysis, but for the authors this loss does not look very significant. A systems which can be simulated by simple and ostensive models, if global variables used, are discussed in the paper. The job is controversial and reflects an experience of authors in simulation of navigation of mobile robots in flexible manufacturing systems. Proposed extension of global variables in colored Petri nets is realized in program package CENTAURUS COLOR.

Peculiarities of Colored Petri Nets

Any colored Petri net consists of two subsystems. One of them is inherited from classic (place/transition) Petri nets [5] and is graphic structure representing an oriented graph and consisting of positions, transitions and oriented links – chords. Appearance or disappearance of tokens in positions, sometimes treated as "movement", enables visually observe and understand operation of a model (net). Formal analysis of the structure lets to detect deadlocks, cycles and other properties of the net.

Another subsystem is programmable and is not met in classic Petri nets. It consists of expressions associated with chords (chord expressions), transitions (guard expressions) and places (init expressions) [5]. Tokens of the colored nets may be complex data structures and contents of the structures may be changed. The SML [6] program language is traditionally used to denominate these expressions. In other words, any Petri net represents a program, flow control of which is supervised by graph structure. A user can create desirable types (colors) of tokens, declare variables of various colors in colored Petri nets, but all variables are strictly local with respect of any transition – any variable can obtain a particular meaning, from token in appropriate place, only in expression of an input chord of a transition, if it is fired, and this meaning can be used only in expressions associated with this transition (guard expression) or in expressions of chords outgoing from this. The result of expressions of outgoing chords is outgoing tokens. So, information can be transmitted through the transition.

Continuing this correspondence between colored Petri nets and high level programming languages some similarities may be noticed:

- Firing of a transition corresponds to calling of a subroutine;
- Expressions of input chords of a transition define input parameters of the subroutine;
- Expressions of output chords define values of output variables of the subroutine;

- All variables in the environment of the transition are local, what is usual practice of subroutines in high level programming languages.

The main difference between colored Petri nets and other high level languages is absence of global variables in the first. This absence ensures strict flow control by the graph structure of a Petri net.

Many control systems operate at real time – any process under control has duration and model of such system must reflect this, usually, by including delays associated with transitions [7]. So the time, global variable, appears in Petri nets. Usage of this variable is specific, because the time is independent variable and influences the flow control of a net, but does not create hidden links between elements of it. Sometimes it is necessary to control duration of delays dynamically [8], so inflating flow control, making it dependent of other variables or events in a net, what is equivalent to presence of hidden links. There are control systems where even before mentioned extension of Petri nets is not sufficient.

Problems of Simulation by Petri Nets of Control Systems with Navigation

Modern flexible manufacturing systems use mobile robots. It is necessary to know position of a robot at any time to avoid collisions with other robots or stationary objects. Coordinates of a robot, using conventional colored Petri nets, must be an attributes of a token because only these attributes can be changed dynamically. On other hand, any token is accessible when it is in a place of net. When a transition simulating a “step” of a robot fires the token with coordinates disappears from input place and information about the robot’s position becomes not accessible during the all delay of the transition (duration of the “step”). It makes a problem to supervise movement of all robots, if a system contains some robots moving with different velocities (different “step” durations), because information about positions of all robots is necessary to make a decision – this causes synchronization of “steps” what means loss of real parallelism of processes associated with different robots. It is necessary to have possibility to get coordinates of any robot even if it is making “step” – appropriate transition is fired and token not accessible, to avoid this loss. The evident solution of this problem is to use global variables. The same problem arises then controller of above mentioned system works with interruptions – an interruption can not be served if information about a position of a robot is not accessible. This causes delays of interruptions, what can essentially complicate all the control process. The last problem was solved in [8] by introducing global variables allowing to control delay of a transition dynamically – to break running delay, make coordinates of a robot accessible and make possible to do a new decision.

Complication of flexible manufacturing systems is still growing and supervision of operation of such systems becomes more and more complex, especially when the same manufacturing system produces different products in parallel. Such systems consist of production centers (machine-tools) and mobile robots. The “behavior” of, both – production centers and robots, can change dynamically. The all possible variants of, for example, a production center must be reflected in the structure of a Petri net simulating such center, if traditional colored Petri net is used. This can be done using subnets (sub models) (Fig. 1), but number of variants may be very large and net - very complicated.

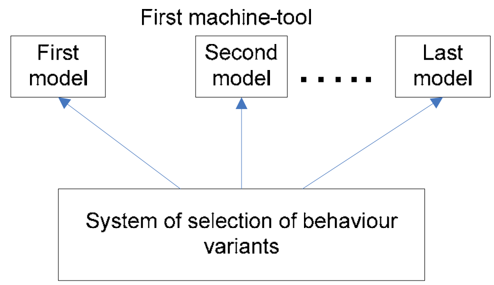

Fig. 1. Diagram of action variants of a machine-tool, when operation conditions changes

So organized model has tendency to grow very rapidly, with increasing number of machine-tools and transportation robots – if machine-tools are two, number of variants increases twice, if three – nine times, so square dependence is noticed. This dependence may be of even lager degree, if number of machine-tools is less then number of products types.

Links in Petri net are not strictly predefined, if mobile robots are used, because what robot in what sequence will serve which machine-tool depends on product to make. Of cause, it is possible to make all possible links (chords) between robots and tolls (Fig. 2), but such net clearly have excessive chords.

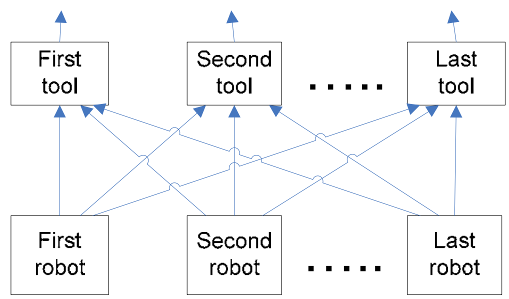

Fig. 2. All possible links of robots in flexible manufacturing system

Usage of Global Variables and Restrictions

It seems to be reasonable to use global variables to overcome before mentioned problems, but presence of global variables creates hidden links in Petri net – flow control of a net is not strictly defined by graph structure. Usage of global variables can cause ambiguous results of simulation also. There is only one way to assign value to a variable in colored Petri nets – to use constructors (structures, tuples, consisting of variables) as expressions of input chords. Suppose that a global variable is used in a constructor associated with one transition and in an expression of an input chord of another one, and then result of the last expression depends on value assigned by the constructor to the global value. But assignment may be done only if appropriate transition fires. So firing conditions of the second transition becomes dependent on a fact of firing of the first one. On another hand, conditions of firing of transitions in fact are calculated not parallel, but in series, in computer with one processor. So marking of a net becomes dependent on order of calculation of firing conditions, if before mentioned situation establishes. It is evident that values of global variables can not be changed while not firing conditions of all transitions are checked in a simulation step. Usage of global variables in constructors is very dangerous and another mechanism, changing global values only at the end of current simulation step, must be formulated.

Authors propose to use an additional attribute of transition – *finish expression,* which would be evaluated then an appropriate transition closes. To minimize danger of usage of global variables the following rules are proposed:

1) usage of global variables in constructors is prohibited;
2) values may be assigned to global variables only using *finish expressions;*
3) new values are written into global variables at *the end of current simulation step;*
4) a special function to change values of global variables must be used, so avoiding unauthorized assignments.

One step of simulation of a Petri net with global variables can be defined by these stages:

1) verification of conditions of firing of all transitions – evaluation expressions of all input chords, checking, if got multisets (results) exist in appropriate places, checking if value of guard expression is *TRUE*;
2) evaluation of expressions of output chords, if a transition fires, stowing of calculated multisets (tokens) into appropriate places;
3) evaluation of *finish expressions* of all transitions and assignment new obtained values to global variables.

Values of global variables do not change along a simulation step, but can be changed just before start of another one.

Authors propose to apply predefined *BOOL* color for *finish expressions*, assign new values for global variables and test the fact of assignment using two new functions:

zap(glob_var, expr): bool;

or

zap(field_of_glob_var, expr):bool;

and

zapped(glob_var): bool,

here ***glob_var*** – name (identifier) of global value, ***expr*** – expression, color of which is the same as ***glob_var,*** o ***field_of_glob_var*** – name or number of a field of *(record* or *product*) structured global variable. Result of ***zap*** function is always *TRUE*. Examples the usage of ***zap*** function:

declaration –

*color example = record a:real * b: int * c: bool;*
global var x: real;
global var y: example;

and usage –

zap(x, 2.56) and zap(#b y, 138);

Result of function ***zapped*** is *TRUE*, if value of a variable ***glob_var*** is changed and *FALSE,* if nof changed.

Example of Simulation using Global Variable

A transportation robot can move being in one of two states – busy or idle. The first state presumes necessity to finish started movement and the second state allows breaking the movement and doing another, more useful task. The advantage of use of global variable may be more perspicuous from an example (result of simulation) shown in Fig. 3.

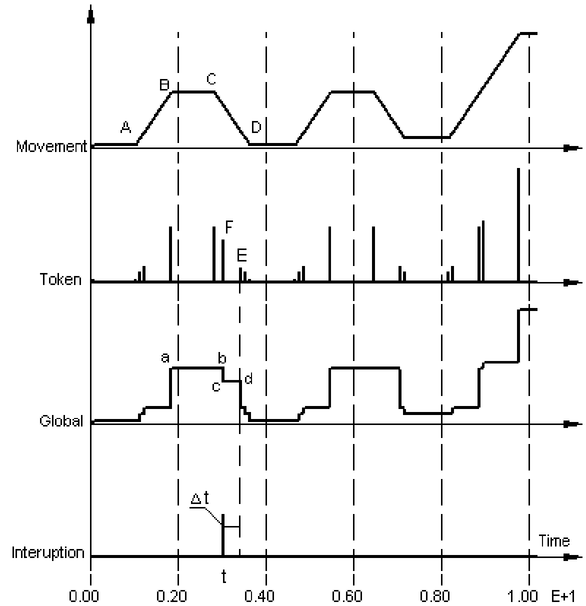

Fig. 3. Simulation of movement of robot: “movement” – the way (coordinate) ‘walked” by the robot (AB – the robot brings a thing from a store to machine-tool, BC – the robot standing by the tool and doing some service job, CD – the idle robot moves to the store); “token” – presence of token, with value of coordinate, in a place, representing the robot; “global” – coordinate of the robot held in a global variable; “interruption” – a moment of time *t*, when request of new job for robot occurs

A robot is represented by structured token in a position of Petri net. One of elements of the structure is a coordinate – the way “walked” by the robot. The value of coordinate is accessible if the token is in the position, but this token disappears, used by transition firing procedure, then robot starts to move or to do service job. Presence or absence of the token is shown in Fig. 3, “Token”. An interruption (Fig. 3, “Interruption”), which requires information about the position of robot, can not be served immediately, because the information is hidden. The delay of interruption Δt , until token E appears, occurs. The coordinate of the robot becomes accessible any time, if it is hold as a global variable (Fig. 3, “Global”). No delay is necessary to get information about the position of the robot in this case and new task for the robot, “go to the store”, delay has been broken and token F produced, for example, can be initiated.

Conclusions

1. Usage of global variables in colored Petri nets for simulation of control systems allows to simplify nets and avoid undesirable synchronization of parallel processes.

2. Formulated rules of usage of global variables preserve parallelism and concurrence of processes in the Petri nets.

References

1. **Baldan P., Corradini A., Ehrig H., Heckel R.** Compositional Modelling of Reactive Systems Using Open Nets // Proceedings of CONCUR 2001.
2. **Milner R.** Performance Modelling with Deterministic and Stochastic Petri Nets// The American Mathematical Monthly. – 1998–105, 9. – 885 p.
3. **Bartkevicius S., Macerauskas V., Sarkauskas K.** Simulation of hybrid control systems using open Petri nets. Industrial Simulation 2003: 1st International Industrial Simulation Conference. – ISC'2003, June 9–11, 2003. –Valencia, Spain. – P. 131–135.
4. **Cabac L., Kohler M.** Relating Higher Order Reference Nets and Well-Formed Nets // Fifth Workshop on Practical Use of Colourd Petri Nets and the CPN Tools. – 2004 october 8–11, University of Aarhus, Denmark.
5. **Jensen Kurt**. Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag, 2nd corrected printing 1997.
6. **Milner R., Tofte M. and Harper R.** The definition of standard ML.– The MIT Press, 1990.
7. **Bartkevičius S. K., Mačerauskas V., Šarkauskas K.** Spalvotųjų Petri tinklų taikymas valdymo sistemoms modeliuoti // Elektronika ir elektrotechnika. – 2003. – Nr. 4(16). – P. 7–11.
8. **Bartkevičius S., Daunoras J., Kragnys R., Šarkauskas K.** Vėlinimų valdymas modeliuojant sistemas Petri tinklais// Elektronika ir elektrotechnika. – 2006. – Nr. 1(65). – P. 16–19.

Submitted for publication 2006 03 01

S. Bartkevičius, R. Kragnys, K. Šarkauskas. Global Variables in Colored Petri Nets // Electronics and Electrical Engineering. – Kaunas: Technologija, 2006. – No. 5(69). – P. 49–52.

Simulating the complex control systems and desiring the results of simulation to maximally draw nearer the real processes, it will be already not enough of those possibilities, which are provided by colored Petri nets. There exist such control systems, for which it is not possible to create adequate model or the model complexity grows to such extent at which it loses its functionality. For the solution of such problems it is proposed to use global variables in the colored Petri nets. Global variables create the possibility to simulate the programs of control, when control system is built using a mechanism of interruptions. The use of global variables breaks the rigidly specific graphic structure of the colored Petri nets, since the concealed connections appear, so that for their use rigid rules are determined. Usage of global variables enables to make models with dynamic control of delays and fast service of interruptions, without damaging causal connections and parallelism of processes in a control system. Ill. 3, bibl. 8 (in English, summaries in English, Russian, Lithuanian).

С. Барткевичюс, Р. Крагнис, К. Шаркаускас. Глобальные переменные в цветных Петри сетях // Электроника и электротехника. – Каунас: Технология, 2006. –№ 5(69). – P. 49–52.

Для моделирования сложных систем управления с целью максимального приближения результатов моделирования к реальным процессам уже нехватает тех возможностей, которыми обладают цветные Петри сети. Существуют такие системы управления, для которых невозможно создать адекватную модель или модель становится такой сложной, что теряет свою функциональность. Для решения таких проблем предлагается в цветных Петри сетях использовать глобальные переменные. Глобальные переменные создают возможность моделировать и программы управления, когда система управления построена используя механизм прерываний. Использование глобальных переменных ломает жестко определенную графическую структуру цветных Петри сетей, так как появляются скрытые связи и для их использования определены жесткие правила. С использованием глобальных переменных могут быть построены модели, в которых возможно динамическое управление запаздывания, при том, не повреждая причинных связей, чтобы не повредить алгоритм функционирования системы. Ил. 3, библ. 8 (на английском языке; рефераты на английском, русском и литовском яз.).

S. Bartkevičius, R. Kragnys, K. Šarkauskas. Globalieji spalvotųjų Petri tinklų kintamieji // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2006. – Nr. 5(69). – P. 49–52.

Modeliuojant valdymo sistemas ir norint modeliavimo procesą kuo labiau priartinti prie realiai vykstančių procesų, nebeužtenka tų galimybių, kurias užtikrina spalvotieji Petri tinklai. Yra tokių valdymo sistemų, kurių nebegalima aprašyti tomis priemonėmis, kurias leidžia spalvoti Petri tinklai, arba modeliai tampa tokie sudėtingi, kad realizuoti jie praranda funkcionalumą. Šioms problemoms spręsti siūloma spalvotuosiuose Petri tinkluose naudoti globaliuosius kintamuosius. Globalieji kintamieji leidžia modeliuoti ir valdymo programas, kai aptarnaujant remiamasi kompiuterių pertraukčių mechanizmu. Globaliųjų kintamųjų naudojimas griauna aiškiai determinuotą grafinę spalvotųjų Petri tinklų struktūrą, atsiranda paslėptų ryšių, todėl yra nustatytos aiškios jų naudojimo taisyklės. Naudojant globaliuosius kintamuosius gali būti organizuotas modelio elementų vėlinimo valdymas nepažeidžiant priežastingumo ryšių ir sistemos funkcionavimo algoritmo. Il. 3, bibl. 8 (anglų kalba, santraukos anglų, rusų ir lietuvių k.).