Synchronizing Parallel Processes using Generalized Nets

Pavel Tcheshmedjiev

Centre of Biomedical Engineering, Bulgarian Academy of Sciences
105 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
E-mail: pavel@clbme.bas.bg

Received: January 20, 2010 **Accepted: March 12, 2010**

Published: April 15, 2010

***Abstract**: In the present paper Generalized Nets are proposed as a common ground for synchronizing parallel processes. We will demonstrate how a GN model can be applied in the context of software maintenance procedures and present the advantages of this approach.*

***Keywords**: Generalized Net, Modelling, Parallel processes, Bioinformatics.*

Introduction

The purpose of this paper is to propose a new approach for designing parallel programs by modeling synchronization mechanism for parallel processes in the terms of generalized nets (GN). There are different approaches and methodologies to describe parallel processes – most of them are using specialized modeling languages, UML sequence and activity diagrams (see [1]). Every approach has advantages and disadvantages. The proposed solution is based on the parallel nature of the generalized nets (see [2]) and provides general purpose solution for designing, analyzing and simulating execution and synchronization of parallel processes.

The paper shows the stated advantages of the GN approach by proposing a new solution for controlling and synchronizing parallel processes in the area of software maintenance. The solution is based on GN models and synchronization points. Using GN transitions to describe process fragments, synchronization points and events we can easily describe complicated parallel scenarios. After the design phase the GN xml representation is translated into complete configuration for direct execution by available Logistics Controller (see [3]).

Another application of the GN parallel approach is the transparent representation, analysis and simulation of various models in the area of bioengineering (bioinformatics, bioprocess systems, biomedical engineering, biotechnology, (see [4])). For example GN models can be used to create effective models of Boolean genetic networks (see [5]) and the proposed GN solution can be applied to solve synchronization and process control issues.

The concept of Generalized Nets

Generalized Nets (see [2]) are extensions of the Petri nets and their other modifications. They are a tool intended for detailed modelling of parallel processes.

A generalized net is a collection of ***transitions*** and ***places*** ordered according to some rules (see Fig. 1). The places are marked by circles. The set of places to the left of the vertical line (the transition), are called input places and the ones that are to the right are called output places. For each transition there is an index matrix with elements – predicates. Some GN-places contain ***tokens*** – dynamic elements entering the net with initial characteristics and getting next ones during their movement in the net. Tokens proceed from the input to the output places of the transitions if the predicate corresponding to these places is evaluated as "*true*". Every token has its own identifier and collects its own history that could influence the development of the whole process modelled by the GNs.

Two time-moments are specified for the GNs: startup and termination of functioning, respectively.

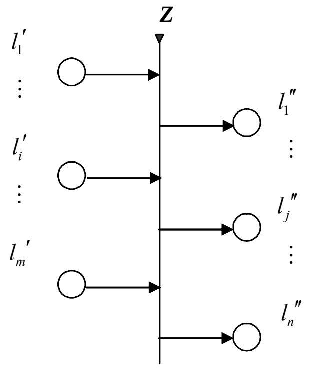

Fig. 1 A GN transition

A GN can have only a part of its components. In this case it is called a reduced GN. Here we shall give the formal definition of a reduced GN without temporal components, place and arc capacities, and token, place and transition priorities.

Formally, every transition in the used below reduced GN is described by a three-tuple:

$$Z = < L', L'', r >,$$

where:

(a) L' and L'' are finite, non-empty sets of places (the transition's input and output places, respectively); for the transition these are
$L' = \{l'_1, l'_2, \dots, l'_m\}$ and $L'' = \{l''_1, l''_2, \dots, l''_n\}$;

(b) r is the transition's *condition* determining which tokens will pass (or *transfer)* from the transition's inputs to its outputs; it has the form of an Index Matrix (IM):

$$r = \begin{array}{c|ccccc} & l''_1 & \dots & l''_j & \dots & l''_n \\ \hline l'_1 & & & & & \\ \vdots & & & r_{i,j} & & \\ l'_i & & & (r_{i,j} - \text{predicate}) & & \\ \vdots & & & (1 \le i \le m, 1 \le j \le n) & & \\ l'_m & & & & & \end{array}$$

$r_{i,j}$ is the predicate that corresponds to the i-th input and j-th output place. When its truth value is "*true*", a token from the i-th input place transfers to the j-th output place; otherwise, this is not possible.

The ordered four-tuple
$E = < A, K, X, \Phi >$

is called a *reduced Generalized Net* if:

(a) A is the set of transitions;
(b) K is the set of the GN's tokens;
(c) X is the set of all initial characteristics which the tokens can obtain on entering the net;
(d) Φ is the characteristic function that assigns new characteristics to every token when it makes the transfer from an input to an output place of a given transition.

A lot of operations (e.g., union, intersection and others), relations (e.g., inclusion, coincidence and others) and operators are defined over the GNs. Operators change the GN-forms, the strategies of token transfer and other. There are six types: global, local, hierarchical, reducing, extending and dynamic operators.

Conceptual GN model for synchronizing parallel processes

Fig. 2 describes a conceptual GN model, used to show how different parallel processes and synchronization points can be modeled with GNs and to design the synchronization logic. It can also be used to validate the correct execution of the processes.

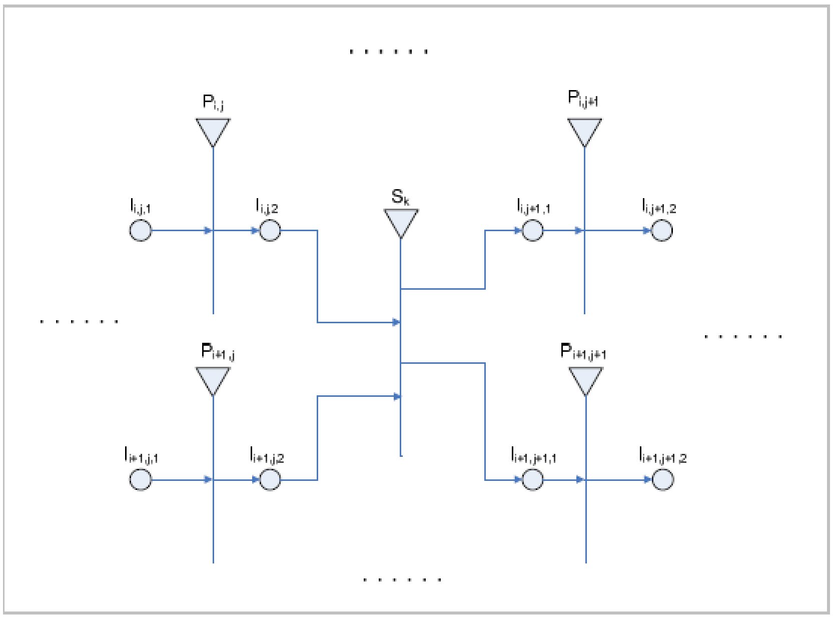

Fig. 2 Conceptual GN model for synchronization of parallel processes

The GN model defines the main principles of execution as follows:

- Transition $P_{i,j}$ represents the j-th phase of the execution of the process P_i;
- Transition S_k represents the k-th synchronization point;
- If an output place of transition $P_{i,j}$ is input place for S_k, the process P_i fire or receive an event related to synchronization point S_k;
- Tokens can be two types:
  - ➢ Synchronization – represent events with the following characteristics:
    - ♦ Identifier;
    - ♦ Registered event identifier;

        - ♦ Event producer identifier – the id of the process;
        - ♦ List of consumer identifier – the ids of the processes.
    - ➢ Information tokens – containing the program segment of $P_{i,j}$ for the respective process P_i.
- The matrix of the predicates evaluates if the movement of the kernels and represents the controlling logic of the program.

The proposed model can be represented in XML format using the unified GN model schema and later transformed to execution configuration. Using this configuration the modeled scenario can be executed on available execution platform.

GN model of parallel scenario with dependent processes

Using the proposed conceptual GN model for synchronization of parallel processes we can design practical parallel scenario with dependent processes in the area of bioinformatics. Let's assume that we have three concurrent routines that are executing their own sets of instructions in order to analyze data in a given regulatory network, consisting of three sub networks *A*, *B* and *C*.

- the first routine is responsible for analyzing network *A* and is represented by process **A**;
- the second routine is responsible for analyzing network *B* and is represented by process **B**;
- the third routine is responsible for analyzing network *C* and is represented by process **C**.

We have the following dependencies between the processes:

- at a given moment process **B** should wait for process **A**, because of dependency on specific activation data, processed by process **A** in network *A*;
- after that process **C** should wait for process **B**, because of dependency on specific activation data, processed by process **B** in network *B*;
- Finally processes **A** and **B** should wait for process **C** because of dependency on specific activation data, processed by process **C** in network *C*.

Fig. 3 shows the GN model of the described scenario.

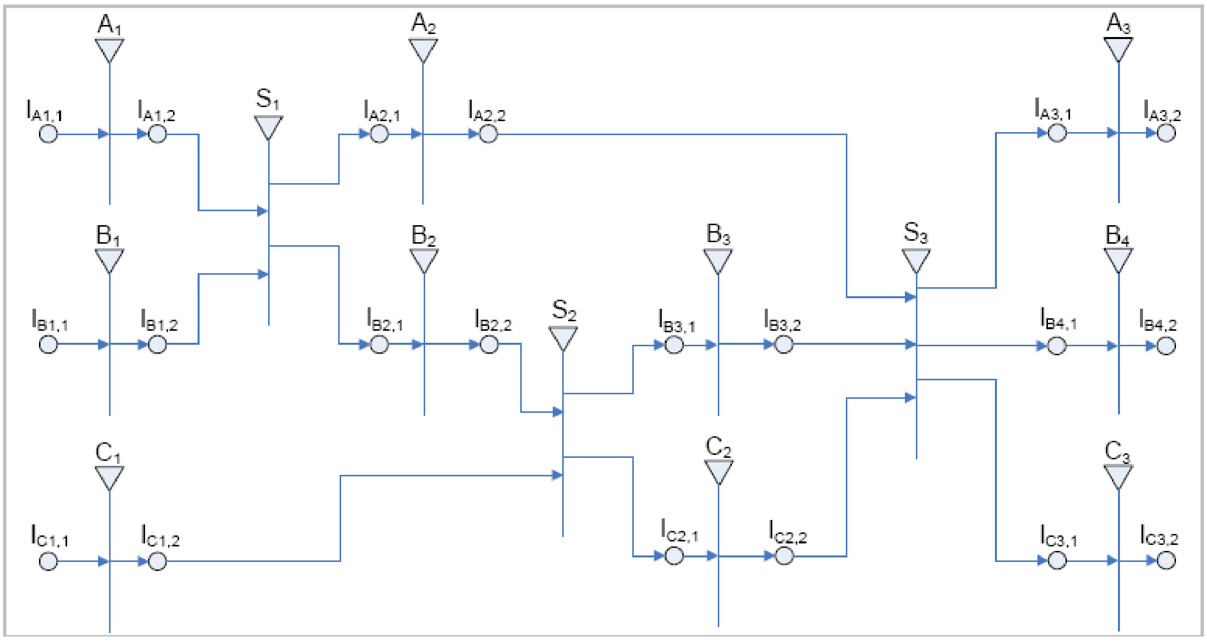

Fig. 3 GN model of parallel processing of example regulatory scenario

Transitions A_1, A_2 и A_3 represent the three phases of process **A**. Transitions B_1, B_2, B_3 и B_4, represent the three phases of process **B**. Transitions C_1, C_2 и C_3 represent the three phases of process **C**. Transitions S_1, S_2 и S_3 represent the three synchronization points.

The predicate matrix of transition A_1:

$$r_1 = \begin{array}{c|c} & l_{A1,2} \\ \hline l_{A1,1} & true \end{array}$$

The condition evaluates to true, showing that the process can execute its specific analyzes. The matrices from r_1 to r_{10} for transitions A_2, A_3, B_1, B_2, B_3, B_4, C_1, C_2 and C_3 are identical. Predicate matrices for transitions S_1, S_2 и S_3 contain the synchronization logic of the scenario.

Predicate matrix S_1:

$$r_{11} = \begin{array}{c|cc} & l_{A2,1} & l_{B2,1} \\ \hline l_{A1,2} & true & false \\ l_{B1,2} & false & W_{B1,B2} \end{array}$$

where:

$W_{B1,B2}$ = "Process **B** can jump from phase B_1 to phase B_2, if there is synchronization token from $l_{A1,2}$ to $l_{A2,1}$".

Predicate matrix S_2:

$$r_{12} = \begin{array}{c|cc} & l_{B3,1} & l_{C2,1} \\ \hline l_{B2,2} & true & false \\ l_{C1,2} & false & W_{C1,C2} \end{array}$$

where:

$W_{C1,C2}$ = "Process **C** can jump from phase C_1 to phase C_2, if there is synchronization token from $l_{B2,2}$ to $l_{B3,1}$".

Predicate matrix S_3:

$$r_{13} = \begin{array}{c|ccc} & l_{A3,1} & l_{B4,1} & l_{C3,1} \\ \hline l_{A2,2} & W_{A2,A3} & false & false \\ l_{B3,2} & false & W_{B3,B4} & false \\ l_{C2,2} & false & false & true \end{array}$$

where:

$W_{A2,A3}$ = "Process **A** can jump from phase A_2 to phase A_3, if there is synchronization token from $l_{C2,2}$ в $l_{C3,1}$",

$W_{B3,B4}$ = "Process **B** can jump from phase B_1 to phase B_2, if there is synchronization token from $l_{C2,2}$ to $l_{C3,1}$".

Concluding comments

Using generalized nets for modeling parallel processes can solve many of the issues, related to the standard modeling approach. The GN models provide intuitive representation of the synchronization logic as well as powerful background for analyzing and simulating the parallel execution. The given example scenario for executing parallel routines for analyzing regulatory network shows the advantages of the GN approach – the complicated procedures and synchronizations points can be easily designed as GN transitions and the logic represented by GN predicates. After the parallel scenario is validated, its XML description can be translated to execution configuration of Execution Platform. By skipping low level technical implementation of the synchronization mechanism, we have minimized the danger of serious problems in the code and semantics of the program.

The next step will be to apply the proposed approach in the area of bioinformatics, where the rapid developments in molecular research and information technologies resulted in huge amount of collected biological data. The GN models can be used in both mathematical and computing approaches and can enhance the understanding of biological processes. The described conceptual model for synchronizing parallel processes can help for the effective and correct representation and execution of various algorithms, targeting the management and analysis of the biological data.

References

1. Penny G., T. Armstrong (2003). Software Maintenance, World Scientific Publishing.
2. Atanassov K. (1991). Generalized Nets, World Scientific, Singapore.
3. Tcheshmedjiev P. (2007). Representing MPI Parallel Processes for Cluster Environment with Generalized Nets, 6th International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets, Warsaw, Poland, 311-317.
4. Gilbert D. (2004). Bioinformatics Software Resources, Briefings in Bioinformatics, 300-304.
5. Kacprzyk J., M. Angelova, P. Vassilev, V. Tasseva, I. Pajeva, K. Atanassov, P. Tchesmedjiev (2007). On the Modelling of Genetic Networks with Generalized Nets", Issues in Intuitionistic Fuzzy Sets and Generalized Nets, 5, 87-96.

Pavel Tcheshmedjiev, Ph.D. Student
E-mail: pavel@clbme.bas.bg

Pavel Tcheshmedjiev is a Ph.D. student in the Centre of Biomedical Engineering, Sofia. He is currently in the process of defending his thesis under the supervision of Prof. Krassimir Atanassov and Prof. Mikhail Matveev. He holds M.Sc. degree in software engineering and his research interests are in the area of software development and design, bioinformatics and algorithms.