Two Simple Models of Petri Nets over Ontological Graphs
Extended Abstract

Krzysztof Pancerz, Piotr Grochowalski, and Wiesław Paja

Department of Computer Science, Faculty of Mathematics and Natural Sciences
University of Rzeszów
Prof. S. Pigonia Str. 1, 35-310 Rzeszów, Poland
{kpancerz,piotrg,wpaja}@ur.edu.pl

Abstract. Two new simple models of high-level Petri nets, called generally Petri nets over ontological graphs, are briefly described. The first model is a conceptually marked Petri net over ontological graphs. In this model, tokens are concepts from ontological graphs associated with places of a Petri net. The second model is an instancely marked Petri net over ontological graphs. In this model, tokens are instances of concepts from ontological graphs associated with places of a Petri net. Basic definitions covering structures and dynamics of Petri nets over ontological graphs are given.

Key words: Petri nets, Ontological graphs, Semantics, OWL ontologies

1 Introduction

In the paper, we show formal basics of new simple models of high-level Petri nets, called generally Petri nets over ontological graphs. On the one hand, Petri nets [10] provide a powerful graphical and formal tool to model dynamic systems. On the other hand, ontologies [8] specify the concepts and semantic relationships among them comprising the vocabulary of particular areas. In our research, we try to combine these important properties of both methodologies used for specifying and modelling selected domains of interest. On the basis of such specifications and models, software systems can be created. Integration of ontologies with Petri nets bridges the gap between the data specification layer and the process layer. New models of Petri nets enable us to use tokens being concepts or instances of concepts present in ontologies. To our knowledge, none of the Petri net models use concepts as tokens in their markings and perform actions on the basis of linguistic semantics of tokens directly derived from ontologies. Some of the best known Petri net models and their markings are collected in Table 1. The proposed models are high-level Petri nets enabling us to obtain much more succinct and expressive descriptions than can be obtained by means of low-level Petri nets (cf. [6]). The presented models enable us to define the conditions for firing transitions in a coherent way on the basis of linguistic semantics of tokens. Our approach tries to make Petri nets fit into a general trend

Table 1. Different types of Petri nets and their markings

Petri Net Type	Markings
Place-Transition (P/T) nets [11]	Indiscernible tokens
Fuzzy Petri nets [7]	Real values between zero and one
Coloured Petri nets [5]	Multisets over colour sets associated with places
Predicate-Transition (Pr/T) nets [4]	Formal sums of n-tuples of individuals (items)
OBJSA nets [3]	Tokens of abstract data types defined using the language OBJ2

in computations proposed by L. Zadeh and called computing with words [13]. The main idea is that words and concepts are used in place of numbers for computing and reasoning. This is justified by the fact that the ability of a human is to perform many tasks without any measurements and calculations on numbers. Special attention can be given to ontologies built in accordance with the OWL 2 Web Ontology Language (shortly OWL 2). OWL 2 is the most recent development in standard languages defined by the World Wide Web Consortium (W3C) [1]. An OWL ontology consists of three components: classes, individuals, and properties. Classes are representations of concepts in a given domain of interest. Classes are interpreted as sets that contain individuals. Individuals (also known as instances) represent objects in the domain of interest. Individuals can be referred to as being instances of classes. Properties (also known as roles or attributes) are binary relations on individuals. The first model, originally proposed in [12], is a conceptually marked Petri net over ontological graphs. In this model, tokens are concepts from ontological graphs associated with places of a Petri net. Dynamics of such Petri nets determines a flow of concepts. The second model is an instancely marked Petri net over ontological graphs. In this model, tokens are instances of concepts from ontological graphs associated with places of a Petri net. Dynamics of such Petri nets determines a flow of instances. The second model can be especially considered in the context of OWL ontologies.

2 Formal Basics

In this section, we present formal basics of new simple models of Petri nets over ontological graphs. Basic definitions covering structures and dynamics of Petri nets over ontological graphs are given.

An ontology specifies the concepts and relationships among them comprising the vocabulary of a given area (cf. [8]). Formally, the ontology can be represented by means of graph structures (cf. [9]). The graph representing the ontology $\mathcal{O}$ is called the ontological graph. Let $\mathcal{O}$ be a given ontology. An ontological graph is defined as $OG = (\mathcal{C}, E, \mathcal{R}, \rho)$, where $\mathcal{C}$ is the nonempty, finite set of nodes representing concepts in the ontology $\mathcal{O}$, $E \subseteq \mathcal{C} \times \mathcal{C}$ is the finite set of edges representing semantic relations between concepts from $\mathcal{C}$, $\mathcal{R}$ is the family of semantic descriptions (in a natural language) of types of relations (represented by edges) between concepts, and $\rho : E \rightarrow \mathcal{R}$ is the function assigning a semantic description of the relation to each edge. Semantic relations are very important components in ontology modeling as they describe the relationships that can be established between concepts and/or their instances. Our attention is focused on four basic semantic relations, namely:

- **EQUIV-TO** (synonymy). If c EQUIV-TO c, then c is a synonym of c.
- **SUBCLASS-OF** (hyponymy), also known as IS-A. If c SUBCLASS-OF c, then c is a kind of c.
- **PART-OF** (meronymy). If c PART-OF c, then c is a part of c.
- **INSTANCE-OF**. If i INSTANCE-OF c, then i is an instance of c.

In description logics used in ontological modelling, there are two distinguished concepts with useful applications (cf. [2]), namely the top concept $\top$ (i.e., a concept with every individual as an instance) and the bottom concept $\bot$ (i.e., an empty concept with no individuals as instances).

We can build some formulas over the set of concepts $\mathcal{C}$ from the ontological graph $OG = (\mathcal{C}, E, \mathcal{R}, \rho)$. In the presented approach, we will use formulas collected in Table 2. The semantics of a given formula ϕ built over the set of concepts $\mathcal{C}$ from the ontological graph OG, will be denoted by $||\phi||_{OG}$.

Table 2. Formulas built over the set of concepts from the ontological graph

Syntax	Semantics	Description
c	$c \in \mathcal{C}$	an individual concept c
$c^{=}$	$\{c\}$	a set containing just an individual concept c
$c^{\simeq}$	$\{c' \in \mathcal{C} : c' = c$ or $c' \, EQUIV-TO \, c\}$	a set containing an individual concept c and all of the synonyms of c
$c^{\leq}$	$\{c' \in \mathcal{C} : c' \neq \bot$ and $c' \, SUBCLASS-OF \, c\}$	a set containing all of the hyponyms of c (including c but excluding $\bot$)
$c^{\subseteq}$	$\{c' \in \mathcal{C} : c' \neq \bot$ and $c' \, PART-OF \, c\}$	a set containing all of the meronyms of c (including c but excluding $\bot$)

Let $\mathcal{C}_p$ be the set of concepts from the graph $OG_p = (\mathcal{C}_p, E_p, \mathcal{R}_p, \rho_p)$. A conceptually marked Petri net $CMPNOG$ over ontological graphs (cf. [12]) is a tuple

$$CMPNOG = (Pl, Tr, \{OG_p\}_{p \in Pl}, Arc_{in}, Arc_{out}, Form_{in}, Form_{out}, Mark_0),$$

where Pl is the finite set of places, Tr is the finite set of transitions, $\{OG\}_{p\in Pl}$ is the family of ontological graphs associated with places, $Arc_{in} \subseteq Pl \times Tr$ is the set of input arcs, $Arc_{out} \subseteq Tr \times Pl$ is the set of output arcs, $Form_{in}$ is the input arc formula function, such that $||Form_{in}(p,t)||_{OG_p} \subseteq \mathcal{C}_p$ for each $(p,t) \in Arc_{in}$, $Form_{out}$ is the output arc formula function, such that $||Form_{out}(t,p)||_{OG_p} \in \mathcal{C}_p$ for each $(t,p) \in Arc_{out}$, and $Mark_0$ is the initial marking function mapping each place p to an element of $\{\bot\} \cup \mathcal{C}_p$,

Let $INST(\mathcal{C}_p)$ be a set of all instances of concept from the set $\mathcal{C}_p$ in a given ontological graph $OG_p = (\mathcal{C}_p, E_p, \mathcal{R}_p, \rho_p)$ and ϵ be an instance of the bottom concept $\bot$. An instancely marked Petri net $IMPNOG$ over ontological graphs $(PNOG)$ is a tuple

$$IMPNOG = (Pl, Tr, \{OG_p\}_{p\in Pl}, Arc_{in}, Arc_{out}, Form_{in}, Form_{out}, Mark_0),$$

where Pl, Tr, $\{OG\}_{p\in Pl}$, $Arc_{in} \subseteq Pl \times Tr$, $Arc_{out} \subseteq Tr \times Pl$ have the same meaning as in case of a conceptually marked Petri net, $Form_{in}$ is the input arc formula function, such that $||Form_{in}(p,t)||_{OG_p} \in \mathcal{C}_p$ for each $(p,t) \in Arc_{in}$, $Form_{out}$ is the output arc formula function, such that $||Form_{out}(t,p)||_{OG_p} \in INST(\mathcal{C}_p)$ for each $(t,p) \in Arc_{out}$, and $Mark_0$ is the initial marking function mapping each place p to an element of $\{\epsilon\} \cup INST(\mathcal{C}_p)$.

In case of a conceptually marked Petri net $CMPNOG$ over ontological graphs, the initial marking function $Mark_0$ assigns concepts to places. The dynamics of $CMPNOG$ is given by firing enabled transitions causing the movement of concepts through the net. A mapping $Mark : Pl \to \{\bot\} \cup \mathcal{C}_p$ is a marking of $CMPNOG$. A transition $t \in Tr$ is said to be enabled if and only if: (1) $Mark(p) \in ||Form_{in}(p,t)||_{OG_p}$ for all $p \in Pl$ such that $(p,t) \in Arc_{in}$, and (2) $Mark(p) = \bot$ for all $p \in Pl$ such that $(t,p) \in Arc_{out}$. If, for $t \in Tr$, there is no $p \in Pl$ such that $(p,t) \in Arc_{in}$, then only condition (2) must be satisfied. After firing an enabled transition t, we obtain a new marking $Mark'$ of $CMPNOG$ such that $Mark'(p) = \bot$ if $p \in Pl$ and $(p,t) \in Arc_{in}$, $Mark'(p) = ||Form_{out}(t,p)||_{OG_p}$ if $p \in Pl$ and $(t,p) \in Arc_{out}$, and $Mark'(p) = Mark(p)$, otherwise.

In case of an instancely marked Petri net $IMPNOG$ over ontological graphs, the initial marking function $Mark_0$ assigns instances to places. The dynamics of $IMPNOG$ is given by firing enabled transitions causing the movement of instances through the net. A mapping $Mark : Pl \to \{\epsilon\} \cup INST(\mathcal{C}_p)$ is a marking of $IMPNOG$. A transition $t \in Tr$ is said to be enabled if and only if: (1) $Mark(p) \in ||Form_{in}(p,t)||_{OG_p}$ for all $p \in Pl$ such that $(p,t) \in Arc_{in}$, and (2) $Mark(p) = \epsilon$ for all $p \in Pl$ such that $(t,p) \in Arc_{out}$. If, for $t \in Tr$, there is no $p \in Pl$ such that $(p,t) \in Arc_{in}$, then only condition (2) must be satisfied. After firing an enabled transition t, we obtain a new marking $Mark'$ of $IMPNOG$ such that $Mark'(p) = \epsilon$ if $p \in Pl$ and $(p,t) \in Arc_{in}$, $Mark'(p) = ||Form_{out}(t,p)||_{OG_p}$ if $p \in Pl$ and $(t,p) \in Arc_{out}$, and $Mark'(p) = Mark(p)$, otherwise. One can see that, in both of the models, we consider a strong condition for firing transitions.

The main idea of conceptually marked Petri nets over ontological graphs is shown in a simple example given in Figure 1. Let us assume that an ontological graph representing grains is assigned to place pl_1 and an ontological graph representing operations made by an agriculture machine is assigned to place pl_2. Input arcs are described by two formulas representing subclasses of the concept $Grain$. If a concept $Wheat\ grain$ (a class of the object recognized by the agriculture machine) is placed as a token in pl_1, then transition tr_1 is enabled to fire because $Wheat\ grain$ is a subclass of $Cereal\ grain$. After firing tr_1, we obtain a new marking shown in Figure 1(b). The concept $Putting$ (an operation made by the agriculture machine) is sent (as a token) to pl_2. The main idea of instancely

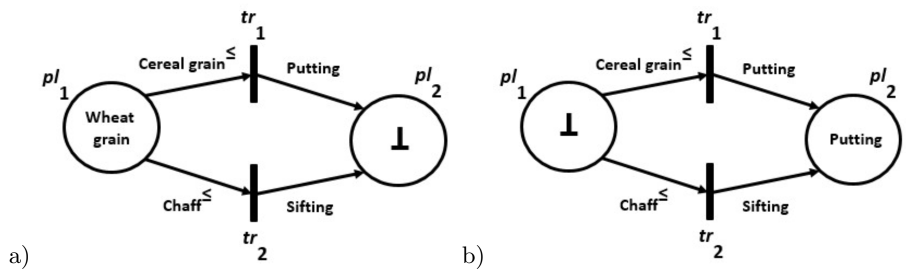

Fig. 1. An example of a conceptually marked Petri net $CMPNOG$ over ontological graphs: (a) an initial state, (b) after firing transition tr_1

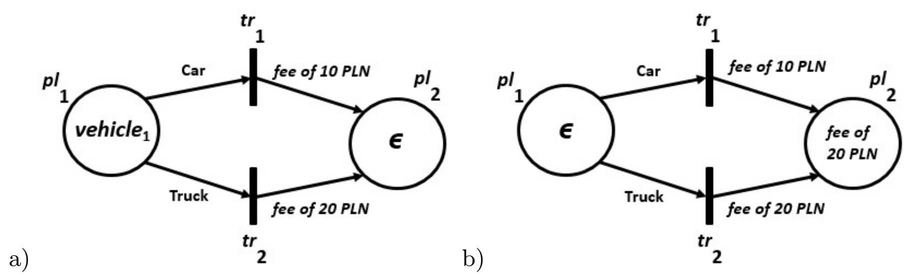

Fig. 2. An example of an instancely marked Petri net $IMPNOG$ over ontological graphs: (a) an initial state, (b) after firing transition tr_2 (if $vehicle_1$ is an instance of $Truck$)

marked Petri nets over ontological graphs is shown in a simple example given in Figure 2. Let us assume that an ontological graph representing vehicles is assigned to place pl_1 and an ontological graph representing fees collected for using a motorway is assigned to place pl_2. $fee\ of\ 10\ PLN$ and $fee\ of\ 20\ PLN$ are instances of the concept $Fee\ for\ a\ motorway$. Input arcs are described by two formulas which are hyponyms of the concept $Vehicle$, namely, Car and $Truck$, respectively. If $vehicle_1$ (a token in pl_1) is an instance of $Truck$, then transition tr_2 is enabled to fire. After firing tr_2, we obtain a new marking shown in Figure 2(b). The instance $fee\ of\ 20\ PLN$ is sent (as a token) to pl_2. Simple examples show the main difference between conceptually marked Petri nets over ontological graphs and instancely marked Petri nets over ontological graphs. We can say that, in the first case, a Petri net operates on concepts, whereas in the second case a Petri net operates on instances of concepts.

3 Conclusions

We have described two simple models of Petri nets over ontological graphs. Such models can be used to describe business processes, reasoning processes, control processes, etc. The main goal of the further research is to propose a complex model of Petri nets over ontological graphs based on OWL ontologies in which object and data properties will be used to control firing transitions. Moreover, we also plan to develop fuzzy Petri nets over ontological graphs.

References

1. OWL 2 Web Ontology Language structural specification and functional-style syntax. W3C Recommendation. Tech. rep., W3C (2012)
2. Baader, F., Horrocks, I., Sattler, U.: Description logics. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 3–28. Springer, Berlin Heidelberg (2004)
3. Battiston, E., De Cindio, F., Mauri, G.: OBJSA nets: A class of high-level nets having objects as domains. In: Jensen and Rozenberg [6], pp. 189–212
4. Genrich, H., Lautenbach, K.: System modelling with high-level Petri nets. Theoretical Computer Science 13(1), 109–135 (1981)
5. Jensen, K., Kristensen, L.M.: Coloured Petri Nets: Modelling and Validation of Concurrent Systems. Springer, Berlin Heidelberg (2009)
6. Jensen, K., Rozenberg, G. (eds.): High-level Petri Nets: Theory and Application. Springer, Berlin Heidelberg (1991)
7. Looney, C.G.: Fuzzy Petri nets and applications. In: Fuzzy Reasoning in Information, Decision and Control Systems, pp. 511–527. Springer Netherlands, Dordrecht (1994)
8. Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., Swartout, W.: Enabling technology for knowledge sharing. AI Magazine 12(3), 36–56 (1991)
9. Pancerz, K.: Toward information systems over ontological graphs. In: Yao, J., et al. (eds.) Rough Sets and Current Trends in Computing, LNAI, vol. 7413, pp. 243–248. Springer, Berlin Heidelberg (2012)
10. Petri, C.: Kommunikation mit automaten. Schriften des IIM nr. 2, Institut für Instrumentelle Mathematik, Bonn (1962)
11. Reisig, W.: Petri Nets. Springer, Berlin (1985)
12. Szkoła, J., Pancerz, K.: Petri nets over ontological graphs: Conception and application for modelling tasks of robots. In: Polkowski, L., et al. (eds.) Rough Sets, LNAI, vol. 10313, pp. 207–214. Springer International Publishing, Cham (2017)
13. Zadeh, L.: Fuzzy logic = computing with words. IEEE Transactions on Fuzzy Systems 4(2), 103–111 (1996)