DEVELOPMENT OF A PETRI NET MODEL FOR A RECONFIGURABLE INTELLIGENT SYSTEM BASED ON EXPERIMENTAL DATA

Juan Ignacio Latorre^a, Karim El-Laithy^b, Martin Bogdan^c, Emilio Jiménez^d

^(a,d) University of La Rioja. High Technical School of Industrial Engineering.C/ Luis de Ulloa 20, 26004 Logroño, Spain ^(b,c) Leipzig University. Faculty of Mathematic and Computer Science. Postfach 10 09 20, 04009 Leipzig, Germany

^(a) juan-ignacio.latorre@unirioja.es, ^(b) kellaithy@informatik.uni-leipzig.de , ^(c) bogdan@informatik.uni-leipzig.de , ^(d) emilio.jimenez@unirioja.es

ABSTRACT

This paper proposes a methodology for solving a double problem. On the one side, it is considered the construction of a Petri net model of a system using data recovered from its behaviour. On the other side, this model should have the capacity of reconfiguration, in order to be able to modify substantially its behaviour depending on its learning experiences. As an open research line related with this methodology, an application on the biological behaviour of neural systems is envisaged

1. Introduction

Several methodologies have been proposed, by different authors, for constructing a Petri net model of a system (Recalde et al., 2004; Piera et al., 2005; Gradišar and Mušič, 2007; Jiménez et al., 2009; Latorre et al, 2011b). However, rarer are the references that deal with constructing a model from scratch by the use of experimental data, which may be incomplete (Lorenz et al., 2007).

On the other hand, the construction of a reconfigurable Petri net system has been the objective of many researchers as well.

The early work of Badouel and Darondeau (1997) introduces the stratified Petri nets as a subclass of the self-modifying Petri nets defined by Valk (1978). The stratified nets change by self-modification and can be used for modelling dynamical processes whose structure evolves along computation. Moreover, Ehrig et al. (2008) describe a theory of transformations of Petri nets derived from graph transformations or graph grammars, applied as stepwise modifications in the static structure. These ideas contribute to the formal basis of the top-down modelling, PN verification and variations in the development process. The concept generalizes the idea

of place fusion of nets by the union or gluing of subnets using interfaces.

The contribution Perez-Palacin and Merseguer (2010) develops an approach to model self-reconfigurable open-world software systems with stochastic Petri nets. The mentioned systems may automatically make decisions regarding their own behaviour, evaluating the effectiveness of the reconfiguration strategies.

A relevant topic in the analysis of the transformations of a Petri net consists of the modifications that do not vary the behaviour of the system. For example, Berthelot (1987) introduced some transformations in Petri nets that preserve certain properties and allow simplifying or refine a system. Recalde et al. (1997) deals with the topic of transformation and decomposition techniques for improving the applicability of structural techniques for liveness analysis of Petri nets. The transformations are aimed at obtaining a Petri net for which well-known results exist for the characterization of certain structural properties.

Several matrix-based operations, applied to the incidence matrices of a set of alternative Petri nets, are presented in Latorre et al. (2011a). These operations correspond to transformations of the alternative Petri nets that preserve their behaviour.

In the following section the goals of the paper will be described. Section 3 will be focused on the chosen formalism, the experimental Petri nets, developed for modelling a real system. The following section describes the procedure to construct a model of a real system, based on the formalism of the experimental Petri nets, from streams of experimental data. The last section presents a research line, focused on the analysis of neural systems, where the modelling methodology presented in this paper will be applied.

2. Objectives to be fulfilled

The methodology proposed in this paper is aimed for the construction of a Petri net model of a system, using data recovered from its behaviour. This data can be generated by means of experiments or by observation of an existing system, whose behaviour should be mimicked by the Petri net model.

Due to limitations in the observation, for example due to a constrained time of field work, or due to limitations in the available experiments, the experimental information may be incomplete for reflecting the real behaviour of the system in the model.

In the Petri net model that will be presented in this paper, the lack of knowledge in the behaviour of the system will be handled in the same way as the capacity of the model of the system to evolve and selfreconfigure as a consequence of its experience. This experience and its implications in behavioural changes of the system may be considered as a learning process of the system. Section four deals with the advantages of the use of the methodology based on the experimental Petri nets,

3. Chosen formalism: experimental Petri nets

The formalism considered to build the model of a real system is the experimental Petri nets, defined in the following way:

 $R = \langle P, T, W, W', m_0, avtf, freqf \rangle$

Where

P is the set of places.

T is the set of transitions. Notice that $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

W is the input incidence matrix.

 \mathbf{W}^+ is the output incidence matrix. Both incidence matrices represent the arcs between the set of arcs and transitions.

 \mathbf{m}_0 is the initial marking.

avtf is a function that associates an average firing time to every transition in *T*.

freqf is a function that associates a frequency of appearance of a transition in a training process to every transition in *T*.

Given a set of transitions associated to a given conflict $T_{cj} = \{t_a, t_b, ..., t_q\} \in T$ a priority of firing is associated to them according to the associated value from **freqf**, being the highest priority the one which corresponds to the greater value of frequency of appearance in the training process and so on.

A model constructed using this formalism can be trained from a set of experimental data for representing

the discrete event system that produces the mentioned data.

4. Advantages of this approach

There are other formalisms that allow representing a real system by means of a quantitative model, such as the neural networks. As an improvement from that kind of formalisms that model a real system by means of "black boxes", experimental PN explicitly may show the internal structure of the real system.

This explicit representation of the internal structure of the real system may allow a researcher to know the present state of the system and the probabilities to change to a new state.

Moreover, when the internal structure of the model is known, it is possible to perform a structural analysis of the model of the system, as well as performance analysis.

Finally, this insight in the behaviour of the model might lead to a better understanding on the behaviour of the real system and maybe new results in the research on the behaviour of the system.

5. Procedure for the construction of a model of the real system

The stage of building up a timed Petri net model is performed from experimental data in the form of nsequences of states, including time information of the change of states, associated to n training events:

$$seq_{1} = \{\sigma_{a1} - [time(a_{1},b_{1})] \rightarrow \sigma_{b1} - [time(b_{1},c_{1})] \rightarrow \sigma_{c1}$$

$$...\}$$

$$seq_{2} = \{\sigma_{a2} - [time(a_{2},b_{2})] \rightarrow \sigma_{b2} - [time(b_{2},c_{2})] \rightarrow \sigma_{c2}$$

$$...\}$$

$$...$$

$$seq_{n} = \{\sigma_{an} - [time(a_{n},b_{n})] \rightarrow \sigma_{bn} - [time(b_{n},c_{n})] \rightarrow \sigma_{cn}$$

$$...\}$$

All the sequences configure the set $S_{seq}{=}\{ \ seq_1 \ , \ seq_2 \ , \ \ldots \ , \ seq_n \}.$

This data should verify some characteristics to allow the development of an accurate model of the behaviour and structure of the real system:

- a) Every possible state and change of state should appear in the experimental data for obtaining an accurate model of the behaviour of the system.
- b) Experimental data that may seem redundant might be, on the contrary, useful, for example if the same sequence appears several times in the experimental data; hence, it should not be removed, since some information to construct

the model of the system depends on the frequency of occurrence of every change of state. That is the case, for example, of the priority of firing transitions belonging to a conflict in the Petri net model.

c) As a consequence of the previous considerations, the larger is *n*, the most accurate the model may be expected to be.

Some assumptions can be made before the model is built up:

- a) Every state is characterised by a single value of energy and a given stable value of energy is associated to a single state.
- b) There are two kinds of transitions between stable states:

b.1) The ones that are produced by known causes, are driven by deterministic relationships of cause and effect.

b.2) The ones, whose occurrence cannot be modelled deterministically because of lack of knowledge and hence introduce the need of a probabilistic approach for parts of the model of the system.

Once the experimental data has been gathered, it is time for constructing the experimental Petri net:

Step 1. Count the number *m* of different states σ_i in the experimental data.

Step 2. Create two matrices of *m* rows and 1 column, called W^- and W^+ respectively, which will represent the input and output incidence matrices of the model.

Create an array **avt** with a single element that will contain the average firing time of the transition, $avt(\sigma_{aj} \rightarrow \sigma_{bj})$. Every element of this array will be associated to the columns in **W**⁻ and **W**⁺ placed in the same position (meaning to the same transition).

Create an array **freq** with a single element, which will represent the frequency of firing the transition $freq(\sigma_{aj} \rightarrow \sigma_{bj})$. Every element of this array will be associated to the columns in **W**⁻ and **W**⁺ placed in the same position (meaning to the same transition).

Let us call $\mathbf{W}_{i,j}^{-}$ a generic element of \mathbf{W}^{-} placed in the *i*th row and *j*th column, while $\mathbf{W}_{i,j}^{+}$ is the element of \mathbf{W}^{+} placed in the same position.

Let us call q the number of columns either in \mathbf{W}^{-} or \mathbf{W}^{+} .

Step 3. For every sequence of states seq_i belonging to the set S_{seq} , repeat the following steps:

Step 3.1. For every column of either in W^{-} or W^{+} (meaning for every transition already

included in the model constructed so far) repeat:

Step 3.1.1. If there is a change of state σ_{aj} -[time (a_{j,b_j})] $\rightarrow \sigma_{bj}$ in seq_i, such that it is possible to find a column k in **W** and **W**⁺ verifying **W**_{a,j,k} $\neq 0$ and **W**_b⁺_{j,k} $\neq 0$ (meaning that the transition has been already included in the model of the system) then modify the values associated to the transition in the following way:

 $avt(\sigma_{aj} \to \sigma_{bj}) \leftarrow [avt(\sigma_{aj} \to \sigma_{bj}) \cdot freq(\sigma_{aj} \to \sigma_{bj}) \cdot i + time(a_j, b_j)] / [freq(\sigma_{aj} \to \sigma_{bj}) \cdot (i+1)]$

 $\begin{array}{l} \textit{freq}(\sigma_{aj} \rightarrow \sigma_{bj}) \leftarrow [\textit{freq}(\sigma_{aj} \rightarrow \sigma_{bj}) \cdot (i\text{-}1) \\ +1] / i \end{array}$

else modify the values associated to the transition in the following way:

$$freq(\sigma_{aj} \to \sigma_{bj}) \leftarrow [freq(\sigma_{aj} \to \sigma_{bj}) \cdot (i-1)] /$$

$$i$$

Step 3.2. For every change of state σ_{aj} –[time(a_{j,b_j})] $\rightarrow \sigma_{bj}$ belonging to the sequence seq_i not considered in the previous **step 3.1.1** repeat:

Step 3.2.1. Add a new column in the matrices \mathbf{W}^- and \mathbf{W}^+ , completed with zeroes, except the element in the a_j th row of \mathbf{W}^- and the element in the b_j th row of \mathbf{W}^+ , which should be 1.

Step 3.2.2. Add a new element in the array **avt** with the following value:

$$avt(\sigma_{aj} \rightarrow \sigma_{bj}) \leftarrow time(a_i, b_j)$$

Step 3.2.2. Add a new element in the array **avt** with the following value:

$$freq(\sigma_{aj} \rightarrow \sigma_{bj}) \leftarrow 1 / i$$

The Petri net model, obtained from this procedure can evolve if more streams of data are used for update the parameters of the model.

Once the model has been updated, it can be used with constant values for the parameters for different purposes such as structural analysis, performance evaluation, and behavioural analysis, just to give a few examples.

6. Application of the methodology to develop a Petri net model of a neural system.

Considering a brain as a biological neural computing system, (El-Laithy and Bogdan, 2011) proposes an interpretation of the behaviour of different features and components of neural systems, such as synapsis, spiking activity and neural states under an energy-based framework. According to this approach, a hypothetical energy function is proposed for dynamic synaptic models.

El-Laithy (2011) shows that a synapse exposes stable operating points and proposes that synapses in a network operating at these stable points can drive this network to an internal state belonging to discrete sets of neural states. The neural states exist as long as the network sustains the internal synaptic energy.

La figura 1 ilustra los estados y las transiciones entre estados que han sido extraídos de la actividad de la red y representa cada estado por medio de la legra griega El eje de ordenadas representa el nivel de estado definido como la distancia de la actividad a una actividad de referencia. Conforme pasa el tiempo, el comportamiento de la red muestra una diversidad de estados transitorios que se han representado por la letra griega σ . Las transiciones entre estados se han ilustrado utilizando flechas. La secuencia de transiciones de estados ilustrada en la figura es el rendimiento del modelo computacional novedoso propuesto para la máquina de estado finita temporal. Estos estados y las transiciones de estados reflejan, de una manera abstracta, los algoritmos generales disponibles para las computaciones en la red como en una máquina de estados. Los estados que duran más señalizan más situaciones de procesamiento final que aquéllos que se mantienen menos tiempo.

Figure 1 illustrates the states and transitions between states that have been extracted from the network activity, and represents each state by the Greek letter σ . The vertical axis represents the level of state defined as the distance from the activity to a reference. As time goes on, the behavior of the network shows a variety of transient states. Transitions between states are illustrated using arrows. The sequence of state transitions shown in the figure is the performance of the novel computational model proposed for the temporary finite state machine. These states and state transitions presents, in an abstract way, the general algorithms available for the computations in the network as a state machine. The states that last longer represent more situations of final processing than those remaining less time.



Figura 1: Result of the previous basic analisys developed (El-Laithy and Bogdan, 2011b).

This system, whose behaviour might be approximated by a discrete event system, constitutes a perfect objective for the application of the methodology based on experimental Petri nets. As a consequence, an open research line that the authors of this paper have undertaken consists in the development of a model of the mentioned biological system, which may allow analysing and understanding better the behaviour of the human brain.

This extended abstract presents the main line of the paper that is proposed, based on the results obtained up to now in the application of the methodology to develop a Petri net model of a neural system based on experimental and simulated data.

References

- Badouel, E., Darondeau, P. "Stratified Petri nets". Research Report No. 3128, INRIA (Institut national de recherche en informatique et en automatique), 1997.
- Berthelot, G., "Transformations and decompositions of nets" in "PetriNets: Central Models and Their Properties, Advances in Petri Nets". Lecture Notes in Computer Science, Brauer, W., Reisig, W., and Rozenberg, G. (Eds.), vol. 254-I, pp. 359–376. Springer, 1987.
- Ehrig, H., Hoffmann, K., Padberg, J., Ermel, C., Prange, U., Biermann, E. and Modica, T., "Petri Net Transformations", in Vedran Kordic (eds) "Petri Net. Theory and Applications", I-TECH Education and Publishing, Vienna, 2008.
- El-laithy, K. "Towards a Brain-inspired information processing system: Modeling and analysis of

synaptic dynamics". PhD thesis, Leipzig University, 2011.

- El-Laithy, K. and Bogdan, M., "A hypothetical free synaptic energy function and related states of synchrony". In Artificial Neural Networks and Machine Learning - ICANN 2011, volume 6792 of Lecture Notes in Computer Science, pages 40-47. Springer, 2011.
- Gradišar, D. and G. Mušič. "Production-process modelling based on production-management data: a Petri-net approach". International Journal of Computer Integrated Manufacturing, Vol. 20, No. 8, pp. 794-810, 2007.
- Jiménez, E., Pérez, M., Latorre, J. I. "Modelling and Simulation with Discrete and Continuous PN: Semantics and Delays", Proceedings of the 21st European Modelling and Simulation Symposium (EMSS 09). Puerto de la Cruz, Spain, vol. 2, pp. 14-19, September 2009.
- Latorre, J.I., Jiménez, E., Pérez, M., "Matrix-based operations and equivalence classes in alternative Petri nets" Proceedings of the 23rd European Modelling and Simulation Symposium (EMSS 11). Rome, Italy, pp. 587-592, September 2011.
- Latorre, J.I., Jiménez, E., Pérez, M., "Simulation-Based Optimization for the Design of Discrete Event Systems Modeled by Parametric Petri Nets". Proceedings of the UKSim 5th European Modelling Symposium on Mathematical Modelling and Computer Simulation (EMS2011), pp. 150-155. Madrid, November 2011.
- Lorenz, R.; Bergenthum, R.; Mauser, S.; and Desel, J. "Synthesis of petri nets from finite partial languages". Proceedings of the 7th International Conference on Application of Concurrency to System Design (ACSD), pages 157–166, 2007.
- Perez-Palacin, D. and Merseguer, J. "Performance evaluation of self-reconfigurable service-oriented software with stochastic Petri nets". Electronic Notes in Theoretical Computer Science (ENTCS). Volume 261, pp. 181-201. 2010.
- Piera, M. A.; Guasch, T.; Porcar, V.; and Bacardit, M. "Constraint Satisfaction in the Automobile Industry: A Coloured Petri Net Simulation Model Approach". In Bruzzone, A.G. and Williams, E. (Eds.): Proceedings of the 2005 Summer Computer Simulation Conference, pp. 267-273. 2005.
- Recalde, L., Teruel, E., Silva, M. "Transformation and Decomposition Techniques to Improve the Decision Power of the Rank Theorems". Proceedings of the Workshop on Petri Nets in System Engineering (PNSE'97), Hamburg, September 25-26, 1997 / Farwer, B.; Moldt, D.; Stehr, M.-O.: Report FBI-HH-B-205, pages 111-122. Universität Hamburg, September 1997.

- Recalde, L.; Silva, M.; Ezpeleta, J.; and Teruel, E. "Petri Nets and Manufacturing Systems: An Examples-Driven Tour". In: Lectures on Concurrency and Petri Nets: Advances in Petri Nets, pages 742-788. Volume 3098 of Lecture Notes in Computer Science / Desel, J.; Reisig, W.; and Rozenberg, G. (editors). Springer-Verlag, June 2004.
- Valk, R. "Self-Modifying Nets, a Natural Extension of Petri nets". Proceedings of Icalp'78. Lecture Notes in Computer Science. Vol 62. Pages 464-476. 1978