MCReNet: a tool for Marked-Controlled Recongurable Nets*

Marisa Llorens and Javier Oliver
Departamento de Sistemas Informáticos y Computación (DSIC)
Universidad Politécnica de Valencia (UPV)
Camino de Vera, s/n E-46022 Valencia, Spain
{mllorens,fjoliver}@dsic.upv.es

Abstract

MCReNet is a tool for the specification, modeling, simulation, and verification of concurrent systems that are subject to dynamic changes by using Marked-Controlled Recongurable Nets. In a marked-controlled recongurable net, a system conguration is described as a Petri net and a change in conguration is described as a graph rewriting rule. A change in conguration amounts to a modication in the ow relations of the places in the domain of the involved rule in accordance with this rule, independently of the context in which this rewriting applies. The enabling of a rule depends on the net topology and on the net marking according to control places.

1. Introduction

The model of marked-controlled recongurable nets (MCRNs) [5, 6] allows to describe the dynamic changes that can take place in the structure of concurrent systems. In MCRNs, a system conguration is described as a Petri net [7] and a change in conguration is described as a graph rewriting rule which consists of replacing part of the system (the part that matches the left-hand side of the rewriting rule) with another one (given by the right-hand side of the rewriting rule). A change in conguration in a MCRN is limited to the modication of the ow relations of the places in the domain of the rewriting rule involved. Most of the changes in conguration that occur in real systems depend on the state of the system (represented in a net by its marking). Therefore, for a rewriting rule to be enabled, so that a change in conguration can take place, not only the net topology is taken into account but also the net marking. Basically, the idea is to have a control mechanism so that the net only makes changes in conguration when a certain minimal marking is reached. See [5, 6] for formal denitions and proofs.

To study real systems modeled by MCRNs it will be useful to have a software tool to analyze the structure and dynamic behavior of the modeled system and so, to evaluate it and suggest improvements or changes. This is the goal of our tool MCReNet.

2. Software Tool MCReNet

The model of MCRNs is expressively equivalent to the model of Petri nets and, therefore, the decidable properties of Petri nets are still decidable for our model [5]. It is possible to obtain an automatic verication tool for MCRNs. Taking into consideration that there are a lot of automatic design and verication tools based on Petri nets, for both business and academic use [9], and the equivalence between Petri nets and MCRNs, we have developed a software tool whose main purpose is the translation of a MCRN into its equivalent Petri net. We start from the following premises: both the initial state and the rewriting rules of a MCRN are Petri nets; a MCRN is formally equivalent to a Petri net; and, we have developed an algorithm [5] that obtains an equivalent Petri net from a MCRN.

The idea is to use an existing graphical editor of Petri nets to draw the initial MCRN and, from this net, our tool must do a syntactic analysis of the generated les and store the relevant information in the suitable data structures; it must implement the algorithm to translate a MCRN into an equivalent Petri net; from the resulting Petri net, it must create a le for being visualized from the used editor.

We have chosen PIPE (Platform Independent Petri net Editor) [10, 2, 1], a free software for non business use that allows to draw, analyze and simulate Petri nets. This editor allows to create, save and load Petri nets according to the last standard XML for Petri nets, PNML (Petri Net Markup Language [8]). The user draws the initial state of the MCRN (a marked Petri net) using PIPE (see Fig. 1), and the several

*This work has been partially supported by the EU (FEDER) and the Spanish MEC under grant TIN2004-00231, by the Generalitat Valenciana GRUPOS03/025, and by the ICT for EU-India Cross-Cultural Dissemination Project ALA/95/23/2003/077-054.
rewriting rules (also marked Petri nets) as well. We obtain various XML files, one for the initial state and two for each one of the involved rewriting rules.

Figure 1. Initial state in PIPE

Figure 2 shows the initial screen of the MCReNet Tool. From this screen, the user introduces the XML files representing the initial state and the left-hand and right-hand side of all rules.

Figure 2. MCReNet Tool

When the user presses the button Create Petri net, the process of translation begins. The translator is implemented in Java [3, 4] following the algorithm developed in [5]. The previous step of this algorithm is to obtain the configuration graph of the initial MCRN. Once obtained it, we are fit to translate this net into its equivalent Petri net. According to the algorithm, the translation process consists in adding new places, new transitions and its corresponding flow relations to the places and transitions of the initial configuration. If the number of places of the original MCRN is \( n \), the number of transitions is \( m \), the number of the distinct configurations of the configuration graph is \( k + 1 \), and the number of configurations immediately reachable from each of the possible configurations is \( z = \sum_{i=0}^{k} (|\Gamma_i^*| + 2) \) (where \( |\Gamma_i^*| \) is the number of output arcs from the configuration \( \Gamma_i \) in the configuration graph of the net), the number of places of the equivalent Petri net is \( n + (k + 1) \) and the number of transitions is \( (k + 1) + m + z \).

From the resulting Petri net, we obtain a file with the same format of the PIPE files, so that the obtained Petri net can be displayed on the PIPE editor. Figure 3 shows the obtained equivalent Petri net. The PIPE itself offers the possibility of doing simulation and analysis of properties on the equivalent Petri net.

Figure 3. Equivalent Petri Net

References