Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 56 (2015) 612 – 617

Procedia Computer Science

International Workshop on the Use of Formal Methods in Future Communication Networks (UFMFCN 2015)

A Framework for Modeling and Analysis UML Activity Diagram using Graph Transformation

Yasmina Rahmoune[a,*], Allaoua Chaoui[b], Elhillali Kerkouche[c]

[a]*MISC Laboratory, Department of Computer Science, Normal High School of Constantine, Constantine, Algeria*
[b]*MISC Laboratory, Department of Fundamental Computer Science and its Applications, Constantine2 University, Constantine, Algeria*
[c]*MISC Laboratory, Department of Computer Science, Jijel University, Jijel, Algeria*

Abstract

The most important advantage of Model Driven Engineering (MDE) is making available tools, concepts and languages to create and transform models. In this paper, we propose a framework to transform automatically UML activity diagram 2.0 (informal notation) to Petri Nets (formal notation) for the analysis purpose using INA analyzer tool. This transformation helps the software designers to analyze and verify properties. For realizing this transformation, we have proposed a meta-model for UML-AD and another one for PN. Based on these meta-models, we define a graph grammar that performs the transformation process. AToM3 is used as a tool for meta-modeling and graph transformation. We verify the resulting Petri nets with model checker INA. We illustrate our contribution with a detailed example.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Diagram UML-AD; Petri Nets; Meta-Modeling; Graph Transformation; AToM3 tool; MDE; Verification.

1. Introduction

The model transformation is an important property of any faithful approach to MDA (Model Driven Architecture) principles[1]. This transformation is performed by the application of transformation rules on a model to produce another. In our study, we provide the transformation models rules of the UML activity diagram to Petri Nets.

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: rahmoune.yasmina@gmail.com,

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2015.07.261

Unified Modeling Language UML[2] is considered as a standard adopted by the OMG with numerous platforms for modeling object oriented software systems. It is a standard widely used for modeling the dynamic behavior of systems. However, it is a semi-formal language that lacks for tools to analyze and to check models before their implementation. To verify the activity diagrams, we need to transform them to a formal specification having verification tools such as Petri Nets, automata, etc.

Petri Nets[3] are used to formalize the behavior of certain components, systems or applications, including those with complex behavior. Since Petri Nets are formal models, they do not bear any ambiguity and can be validated. In this paper, we propose an approach and a tool to achieve this transformation. Our framework is based on the use of two essential elements that are the meta-modeling and graph grammars. The proposed framework contains two parts. The first part is automatic transforming the UML-AD to PN (model 2 model) and the second part is automatic transforming PN to file with .pnt extension (model 2 code) which is an input of INA[4] analyzer tool (Integrated Net Analyser). We begin our framework with the proposal of two meta-models: one for AD and the second for PN. From these meta-models, AToM3[5] generates automatically a visual tool to build AD and PN models. Then, we propose a graph grammar consisting of a set of graph transformation rules. After, another transformation is performed to generate the code (file with .pnt extension) equivalent to PN (the output of the previous step). We use AToM3 as a tool for graph transformation and INA tool for verification properties.

The remainder of this paper is organized as follows: In section 2, we introduce some related works. In section 3, we propose and describe our approach based on a graph grammar and a meta-model for the transformation of UML Activity Diagram to Petri Nets and their verification. In section 4, we present a detailed example to show the evaluation of our contribution. Finally, a conclusion and some perspectives are given in section 5.

2. Related work

There is much research work in the area of model transformation using graph grammars in the literature.
In[6], the authors have proposed an approach that automatically generates a Communicating Sequential Processes (CSP)[7] specification from AD. They chose the CSP formalism because CSP language is a formal specification language which allows us to check certain properties. In[8], the authors proposed an approach for transforming UML Statechart and collaboration diagrams to Colored Petri Nets models. For this, they have defined an automated approach and a tool environment that formally transforms dynamic behaviors of systems described using UML models into their equivalent Colored Petri Nets (CPN) models for analysis purpose. In[9], the author presented a set of rules that allows software engineers to translate the behavior described by a UML Activity Diagram (AD) into a Petri Nets (PN). The main purpose of the mapping to Petri Nets is to use the theoretical results in the Petri Nets domain to analyze the equivalent Petri Nets and infer properties of the original workflow.

After a comparative study of these approaches, we find that the transformation of AD to PN that was done before is not automatic. The interest of our proposition is to automate the transforming of activity diagrams to Petri Nets. It is the strong point of our approach compared to neighboring works and represents the first advantage. The second advantage consists of the creation of a tool to manipulate an AD model (create and modify an AD). The third advantage is the creation of a tool to manipulate a PN (create and modify a PN model). The fourth advantage is the verification of certain properties of the resulted PN using the model checker INA and give a feed back on the properties of the system modeled the input AD.

3. The proposed framework

Our main goal in the proposed framework consists of transforming automatically UML activity diagrams that are semi-formal (informal notation) to the Petri Nets specification (formal notation). Then we verify the resulting PN with the model checker INA. To achieve this objective, two steps are defined:

- In the first step, we have proposed two meta-models, the first meta-model for UML activity diagram and the second meta-model for Petri Nets. After, we have proposed a graph grammar that performs automatically the transformation of activity diagram.
- In the second step, we have integrated our transformation of the result PN to input of the model checker INA proposed in[10] and we launch the operation of the verification.

In the following, we explain these steps in detail.

3.1. Meta-Modeling Activity Diagrams

For the activity diagrams, the proposed meta-model including eight classes linked by fourteen associations is shown in figure 1. It is our meta-model published in[11].

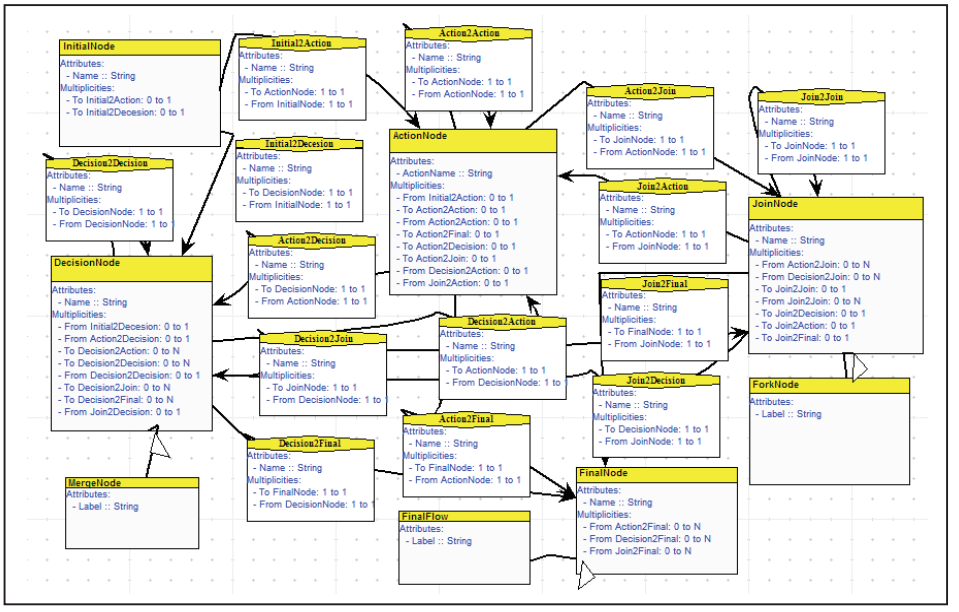

Fig. 1. Activity diagram Meta-Model.

3.2. Meta-Modeling Petri Nets

For the Petri Nets, the proposed meta-model contains two classes linked by two associations such as shown in figure 2.

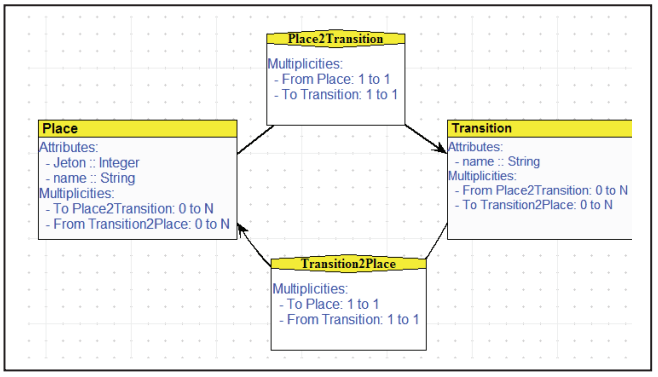

Fig. 2. Petri Nets Meta-Model.

3.3. Our proposed Graph Grammar

For transforming UML activity diagrams to Petri Nets, we have proposed a graph grammar composed of forty-five rules. These rules will be executed in ascending order.

For lack of space we only describe in the following some rules (see figure 3).

- *Rule 1: InitialNode2Place (priority 1):*

  This rule is applied to attach Initial node for UML activity diagram to a new place, and specifies that the name of the attached place is the same name as the corresponding initial node.

- *Rule 2: Action2Transition (priority 2):*

  This rule is applied to attach each Action (not previously processed) to a new Transition, and specifies that the name of the attached Transition is the same name as the corresponding Action.

- *Rule 3: Decision2Place (priority 3):*

  This rule is applied to attach Decision for UML activity diagram to a new place, and specifies that the name of the attached place is the same name as the corresponding Decision.

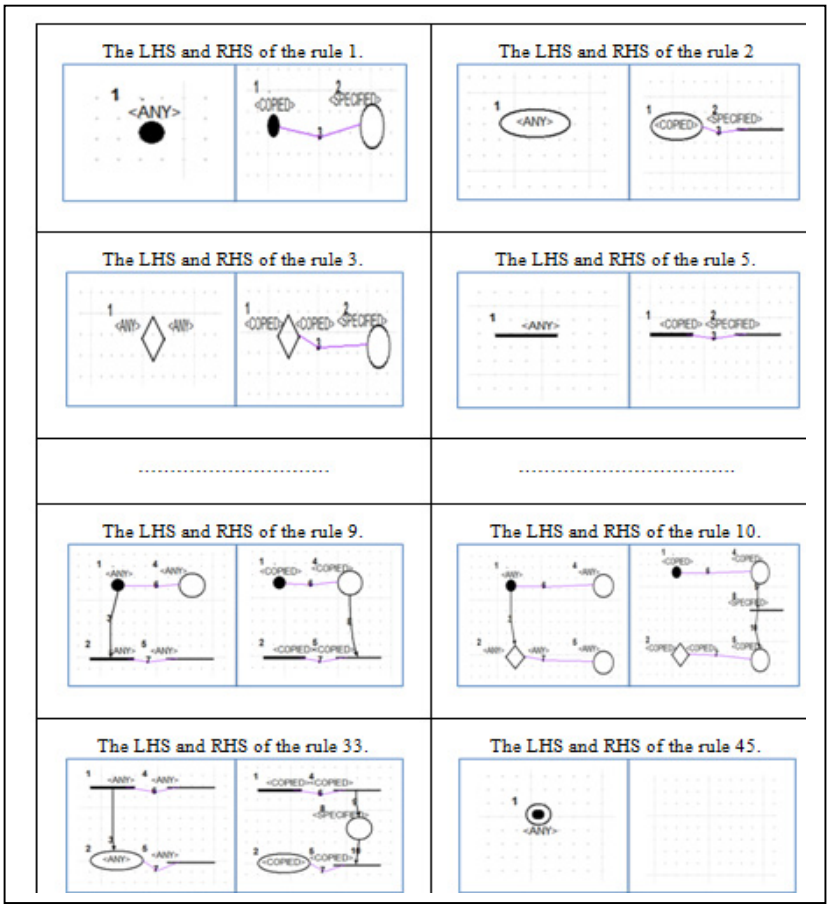

Fig. 3. Set of proposed rules in our graph grammar.

3.4. Translating Petri Nets Model to INA Specification

The objective of this part is the automatic generation INA description, for allow the user to avoid the errors when this description is done manually. Then, the INA tool is used to perform the simulation and the analysis of the resulted INA description.

In this work, we are integrate our rules proposed in[10] for automatically generate a file contains a description INA (.pnt file extension), for more detail see[10].

4. Case Study

The purpose of this section is to present a case study to evaluate the steps of the proposed approach. We have chosen calls phones as a case study. This example illustrates some situations that occur within UML-AD and how they are transformed into PN, such as sequence flow, decision, merge and join. The next figure presents the transformation of activity diagram of the calls phone created by our proposed tool. This model is transformed into Petri Nets using the graph grammar defined in section 3.3.

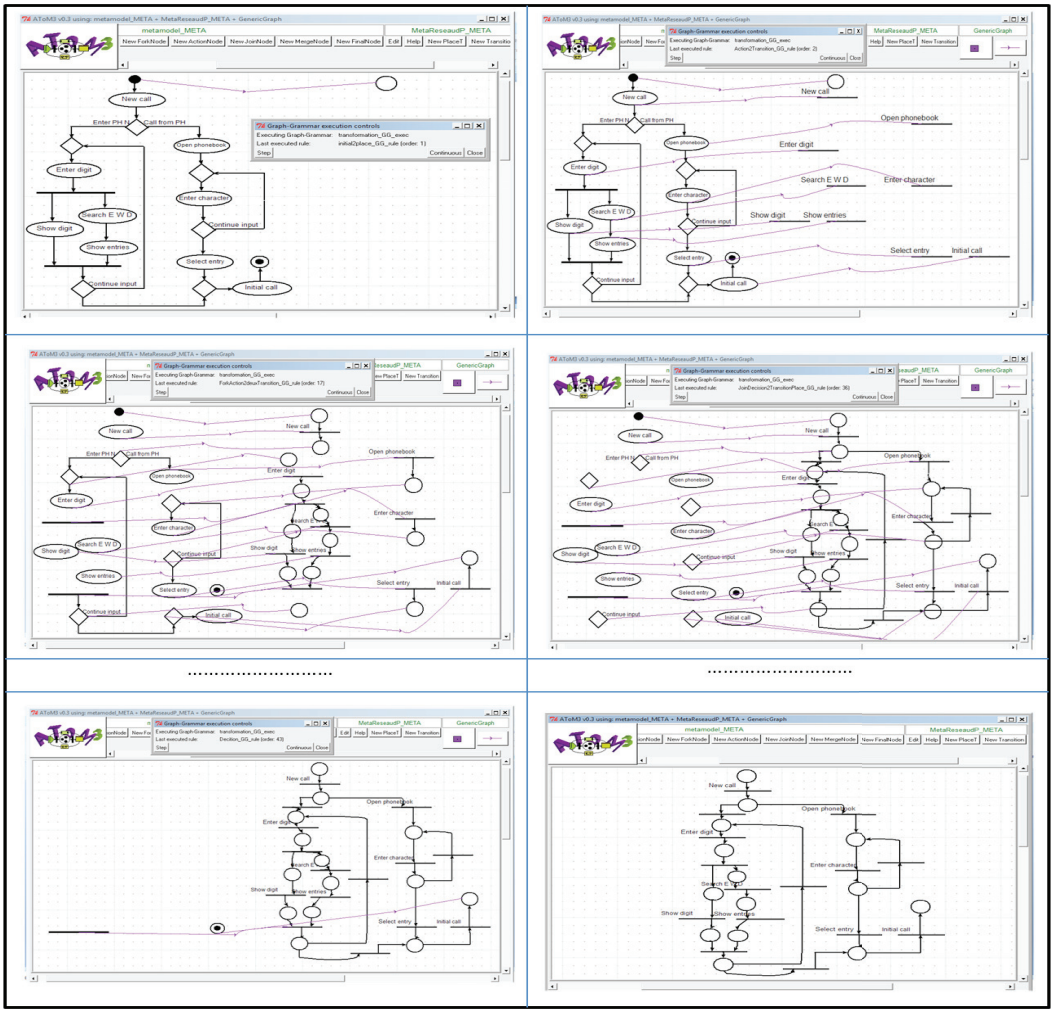

Fig. 4. Some steps of transforming AD to PN with our tool.

At the end of the first part of the transformation, the resulted PN (see figure 4) is transformed to generate INA description as shown in figure 5 (a).
The result of the model checker INA of the description presented above (figure 5 (a)) is shown in figure 5 (b).
Among the results: "The net is not live and the deadlock-trap-property is not valid" which indicates that the PN modeled in figure 5 (b) contains a deadlock situation.

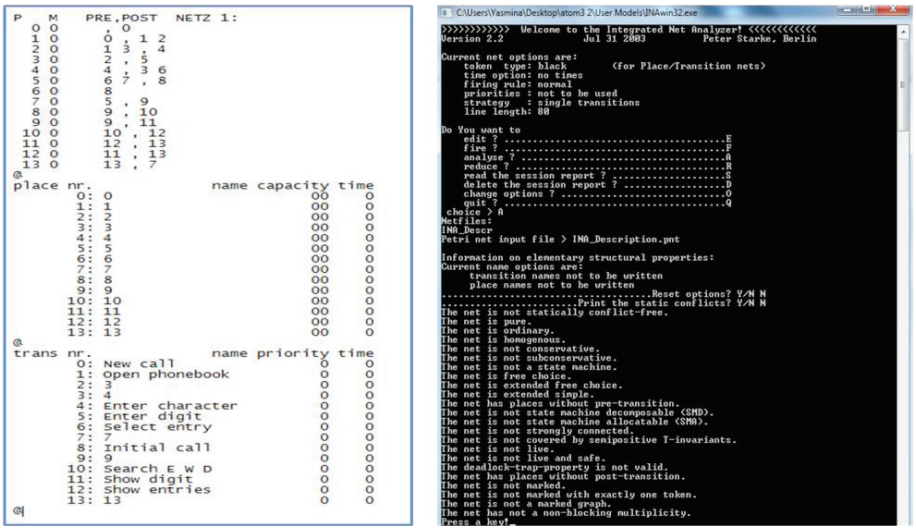

Fig. 5. (a) Generated INA specification of the PN given in Fig 4; (b) Result of the application of the INA tool.

5. Conclusion

In this paper, we have proposed a framework containing a visual modeling tool based on the use of graph grammars and AToM3 tool. Our proposal framework allows the transformation of UML activity diagrams to Petri Nets and the verification of some properties of the resulting PN. To achieve this transformation, we have used UML Class diagram formalism as meta-formalism and proposed two meta-models for the input models (UML activity diagrams) and output models (Petri Nets); we have also proposed a graph grammar to realize the translation process. The objective of this transformation is to benefit from the power of Petri Nets in terms of verification of properties like deadlocks, termination, etc. We are used the model checker of INA tool to verify the properties of the resulted PN models and give a feed back on the properties of the system modeled the input AD. In a future work, we plan to use another specification and we verify other diagrams of UML.

References

1. Soley et al., MDA (Model-Driven Architecture), White Paper, Draft 3.2, OMG Staff Strategy Group, November 27-th 2000.
2. Booch G, Ivar Rumbaugh, Jim Jacobson. The Unified Modeling Language User Guide, Addison-Wesley, 1999.
3. Murata T. Petri Nets: Properties, analysis and applications. Proceedings of the IEEE, vol. 77, no. 4, April 1989. p. 541–580.
4. INA home page. Available: http://www2.informatik.hu-berlin.de/~starke/ina.html
5. AToM3 home page. Available: http://moncs.cs.mcgill.calMSDLlresearch/projects/ ATOM3.html
6. Elmansouri R, Hamrouche H, Chaoui A. From UML Activity Diagrams to CSP Expressions: A Graph Transformation Approach using AToM3 Tool. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011. ISSN (Online). 2011. p.1694-814
7. C.A.R. Hoare. Communicating Sequential Processes. Prentice Hall International Series in Computer Science. Prentice Hall. April 1985.
8. Kerkouche E, Chaoui A, Bourennane E, Labbani O. A UML and Colored Petri Nets Integrated Modeling and Analysis Approach using Graph Transformation. In Journal of Object Technology, vol. 9, no. 4, 2010. p. 25–43. Available at http://www.jot.fm/contents/issue_2010_07/article2.html
9. Bhawana Agarwal. Transformation of UML Activity Diagrams into Petri Nets for Verification Purposes. International Journal Of Engineering And Computer Science ISSN: 2319-7242, Volume 2 Issue 3 March 2013. p. 798-805.
10. El Mansouri R, Kerkouche E, Chaoui A. A Graphical Environment for Petri Nets INA Tool Based on Meta-Modelling and Graph Grammars. Proceedings of World Academy of Science. Engineering and Technology Volume 34 October 2008 ISSN 2070-3740, Venice, Italy.
11. Yasmina Rahmoune and Allaoua Chaoui.: An Approach Based on Graph Grammar and Meta-Modeling for Transforming Automatically Business Process Models to UML Activity Diagrams. International Conference on Advanced Technology & Sciences, ICAT'14, Antalya, Turkey, August 12-15, 2014. p. 1116-1122.