FROM A MODEL OF CONCURRENCY TO A TEST MODEL: A GRAPH TRANSFORMATION BASED APPROACH

A. CHAOUI, D.SAIDOUNI, M. BOUARIOUA, S. BEKRAR, E. KERKOUCHE

MISC Laboratory, Computer Science Department, Faculty of Engineering, University Mentouri Constantine, 25000, Algeria
a_chaoui2001@yahoo.com, Saidounid@hotmail.com

ABSTRACT

Maximality-based Labeled Transition Systems (MLTS) is semantic model for true concurrency. In other hand Mixed Refusal Graphs (MRG) are models for formal testing. In this paper, we propose an approach to transform an MLTS model to an equivalent MRG model. Since the input and output models are graphs, we use graph transformation to perform this transformation automatically. So, we propose two meta-models; one for the input model and the other for the output model. Then, based on these meta-models we propose a graph grammar that deals with the transformation process. The meta-modeling tool ATOM^3 is used. Our approach is illustrated through an example.

KEYWORDS: Formal Verification, Graph transformation, MLTS, MRG, ATOM^3, Meta-models, Test model.

1 INTRODUCTION

Nowadays, technology is looking for distributed applications to develop and increase its domains (network, telecommunication…etc). This kind of applications is known by their big complexity. Formal verification method is the most used technique to deal with concurrent systems questions because of its ability to describe the system behavior without ambiguity. At the final step of the formal design trajectory of a system, the implementation under test is verified according to the specification of the system. In both specification and testing models, the hypothesis of non-atomicity of actions is considered. For this reason we will use the MLTS model [9] for representing the systems behaviors and the MRG model as a support of the formal testing verification approach. In [10], an algorithm for generating MRG structures from MLTS ones is defined. Since the MLTS and MRG models are both graphs, we are interested in this paper to use a graph transformation approach [8] to deal with transforming MLTS models to MRG Models. In previous papers [5, 6], we have proposed two graph transformation approaches to perform two model transformations using ATOM^3 (A Tool for Multi-formalism and Meta-modeling) environment [1,3] as a graph transformation tool . In [5], we have proposed two automatic steps to perform the transformation of G-Nets models to their equivalent in PROD Language. The first one deals with the transformation of G-Nets models into Prt-Nets models. The second one transforms the resulted Prt-Nets models into PROD language. In [6], we have proposed an automated approach and a tool environment that formally transform dynamic behaviors of systems expressed using UML models into their equivalent Colored Petri Nets (CPN) models for analysis purpose.

In this paper we propose an approach and a tool for transforming MLTS models to their equivalent MRG models using ATOM^3 as a graph transformation tool. The MLTS models are obtained using FOCOVE tool [4] from a LOTOS specification. The obtained MRG models will be exploited by verification tools like FOCOVE for example. Here we are concerned only by dealing with the transformation of MLTS models to MRG models.

This paper is organized as follows. In section 2, we recall some basic concepts about maximality-based labeled transitions semantic models and the refusal graphs as verification methods. Then we will explain model transformation concepts and especially graph transformation with its main tools and methods. A brief introduction of ATOM^3 will be given. In section 3, we describe our approach of transforming MLTS models [9] to MRG models [10] based on graph grammars. It consists on proposing two meta-models associated respectively to the MLTS model and the MRG model and a grammar that deals with the transformation. The meta-models are represented by UML class diagrams [2] and the constraints are expressed using Python language [7]. In section 4, we illustrate our approach through an example. The final section concludes the paper and gives some perspectives.

2 BACKGROUND

Our objective is to propose an automatic generation of refusal graphs from Maximality-Based Labeled Transitions Systems using graph transformation. In the following, we recall some basic notions about Maximality-Based Labeled Transitions Systems, Refusal Graphs, and graph transformation.

2.1 Maximality-Based Labeled Transitions Systems

maximality-based labeled transition system [9] is model of true concurrency. In a maximality-based labeled transition system, a transition represents the start of action execution. An event name is used to identify this action execution. To each state, a set of all event names, identifying actions which may in execution at this state, is associated; these event names are said maximal at this state. For each state and each transition starting from it; every maximal event name at this state which is not a cause of this transition may not be used for identifying it.

To illustrate the principle of the MLTS model let's consider the example of two machines M1 and M2 which can offer two cups of coffee after the introduction of a coin. Their behaviors are as follow: M1 has a single device which delivers coffee while M2 provides two devices delivering two cups of coffee at the same time. The customer must submit a coin *p* and interacts twice on the button *c* used for requesting coffee. Obviously, a customer using these two machines can observe the difference between them. In fact, after introducing a coin p, the machine M1 starts offering the first cup of coffee after the first interaction on c, the second interaction on c is refused until the first cup is completely delivered. However, in M2, after introducing the coin, the costumer may interact twice on c. No refusals on this action are observed, since the two cups of coffee are delivered in parallel. This difference between M1 and M2 has been observed because the delivery of coffee is not instantaneous.

Figure 1: Maximality-based Labeled Transition Systems of machines M1 and M2

The difference between the two systems may be captured in states S3. In the first system, the set of maximal vents associated to S3 contains only one event name x corresponding to the start of the last execution of action c; however in the second system, the set of maximal event names contains the event names x and y corresponding respectively to the first and the second executions of action c. This means that two caps of coffee may be delivered at the same time by this machine. Then, the distinction between machines M1 and M2 is captured at the semantic level.

Formally, an MLTS is defined as follows.

Definition

Let M be a countable set of event names, a maximality-based labeled transition system of support M is a quintuplet $(\Omega, \lambda, \mu, \xi, \psi)$ with:

$\Omega=(S, T, \alpha, \beta, S0)$ is a transition system such that:

- S is a non-empty set of states; it can be finite or infinite.
- T is the set of transitions indicating the change of state that the system can achieve; this set can be finite or infinite.
- α, β are two applications defined from T to S : for any transition t of T: $\alpha(t)$ is the origin of the transition and $\beta(t)$ its target.
- S0 is the initial state of the transitions system Ω.
- $(\Omega; \lambda)$ is a transitions system labeled by the function λ on a set of actions Act (Act is called the support of $(\Omega; \lambda)$) / $(\lambda: T \rightarrow A)$.
- Ψ : is a function associating to each state a finite set of maximal events names which are presents at this state.
- μ is a function associating to each transition a finite set of event names corresponding to actions that have begin their execution, and the enabling of this transition depends on the termination of these actions.
- ξ : is a function associating to each transition the name of the event that identifies its start.

Such that $\psi(s0)=\emptyset$ and for all transition t , $\mu(t) \subseteq \psi(\alpha(t))$, $\xi(t) \notin \psi(\alpha(t)) - \mu(t)$ and $\psi(\beta(t)) = (\psi(\alpha(t)) - \mu(t)) \cup \{\xi(t)\}$.

For more details, the reader is referred to [9].

2.2 Refusal Graphs

Refusal graphs [10] are models for testing reactive systems. They take into consideration more realistic hypothesis for testing such systems. In fact, mixed refusal graphs consider a new kind of refusals, named temporary refusals. They are induced by actions with non-null durations in the system. For considering actions durations, the maximality-based labeled transition systems are used as a semantic model of reactive systems.

As an illustration, consider again the example of the machines M1and M2. Their mixed refusal graphs are depicted respectively by the Figures 2.(a) and 2.(b). The difference between the two machines is captured by the temporary refusal on action c. In fact, after the trace p.c, no temporary refusal on c is observable in state 3 of Figure

2.(b), which is not the case in state 3 of Figure 2.(a).

Figure 2: Mixed refusal graphs of machines M1 and M2

Formally, refusal graphs are defined as follows.

Definition

a mixed refusal graph is a deterministic structure labeled both on transitions and states defined as follow:

mrg=(G, $\sum$, Δ, Ref, g_0) where:

- G is a finite set of states, g_0 the initial state.
- $L \subseteq$ Act is a finite set of observable actions: the alphabet of mrg.
- $\Delta \subseteq$ (G x L x G): is a transition relation. An element (g, a, g') $\in \Delta$ will also be noted $g \rightarrow g'$.
- Ref : is an application that associates to any $g \in G$ a set of subset of $L \cup L$ such that:

$$\sim \quad \sim \quad \approx \quad \approx$$

$$L = \{a: a \in L\} \text{ et } L = \{a: a \in L\}$$

For more details, the reader is referred to [10].

2.3 Graph Transformation

The transformation between models is a process that converts a model to another model. This task requires a set of rules that define how the source model has to be analyzed and transformed into other elements of the target model. The transformation engine takes the source model in input; execute the rules of transformation and finally generate the target model in output.

Graph Grammars [8] are used for model transformation. They are composed of production rules; each having graphs in their left and right hand sides (LHS and RHS). Rules are compared with an input graph called host graph. If a matching is found between the LHS of a rule and a subgraph in the host graph, then the rule can be applied and the matching subgraph of the host graph is replaced by the RHS of the rule. Furthermore, rules may also have a condition that must be satisfied in order for the rule to be applied, as well as actions to be performed when the rule is executed. A rewriting system iteratively applies matching rules in the grammar to the host graph, until no more rules are applicable. ATOM^3 is a graph transformation tool among others. In this paper we use ATOM^3.

Example of grammar rule in ATOM^3

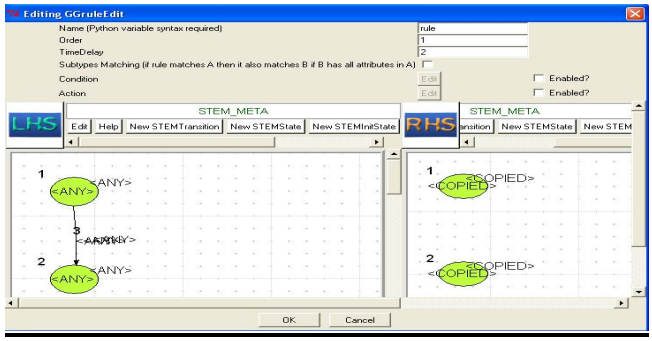

Figure 3: A grammar rule (LHS and RHS) that eliminate a transition between two states

In the next section, we will discuss how we use ATOM^3 to generate mixed refusal graphs models from maximality-based labeled transitions system models using graph transformation based on ATOM^3 tool.

3 OUR APPROACH

In our approach, we propose two meta-models associated respectively to the MLTS model and the MRG model and then we will propose a grammar for the transformation. The meta-models are represented by UML class diagrams and the constraints are expressed using Python language.

3.1 Meta-Models

The meta-models in ATOM^3 are a UML class diagrams and the constraints are expressed in python language.

3.1.1 MLTS Meta-Model and A tool for MLTS models

In this section we propose a meta-model for MLTS and from this meta-model we generate a tool for manipulating MLTS models.

3.1.1.1 MLTS Meta-Model

An MLTS consists of states (with an initial state) and transitions. So, our meta-model of MLTS is composed mainly of two classes (MLTSstate, MLTSinitstate) and an association (MLTStransition association) as shown in figure 5 and described below:

MLTSstate

This class represents the MLTS states. Every state has 3 attributes: a name (name), a maximal set of events name in this state (eventmax) and (path) which contains the executed actions from the initial state until this state. This class is connected to MLTSinitstate by inheritance link.

MLTStransition association

it describes MLTS transitions. Every transition is identified by a set of events names corresponding to actions that have started their execution (Mu), Action, and the event name that identifies its start.

MLTSinitstate

This class represents the initial state of the MLTS. It inherits its attributes from STEMstate class.

Each class has a unique graphical appearance.

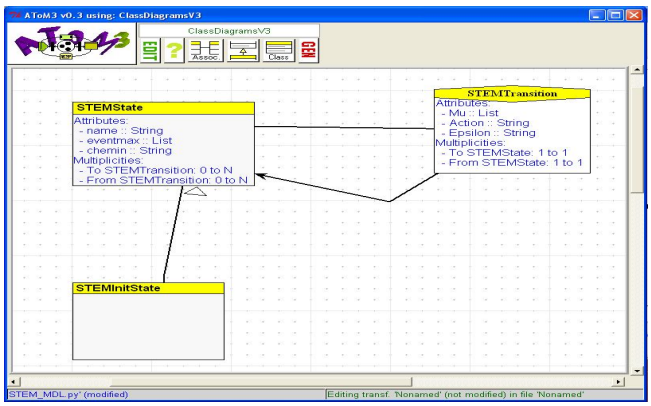

Figure 4: MLTS meta-model

3.1.1.2 A tool for MLTS models

Based on the meta-model of figure 4, we have generated using ATOM^3 a tool for MLTS models as shown in the tool bar of figure 5.

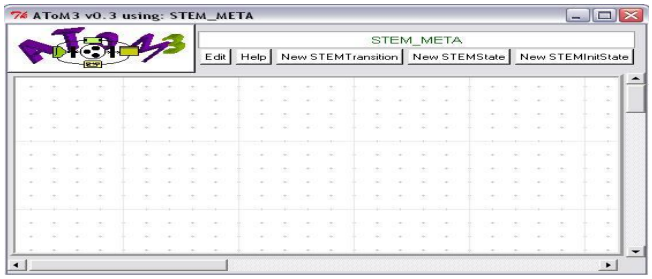

Figure 5: A tool for MLTS models generated using ATOM^3

3.1.2 MRG meta-model and a tool for MRG Models

In this section we propose a meta-model for MRG and from this meta-model we generate a tool for manipulating MRG models using ATOM^3.

3.1.2.1 MRG meta-model

An MRG consists also of states (with an initial state) and transitions. So, our meta-model of MRG is composed mainly of two classes (MRGSstate, MRGinitstate) and an association (MRGtransition association) as shown in figure 7 and described below:

MRGstate:

This class represents the MRG states. Each state is identified by its name (name), a set of refusals (ref) and (path) which contains executed actions from the initial state until this state. This class is connected to MRGinitstate by inheritance link.

MRGtransition association

It describes MRG transitions. Every transition is identified by an action.

MRGinitstate

This class represents the initial state of the MRG. It inherits attributes from MRGstate class (name, ref).

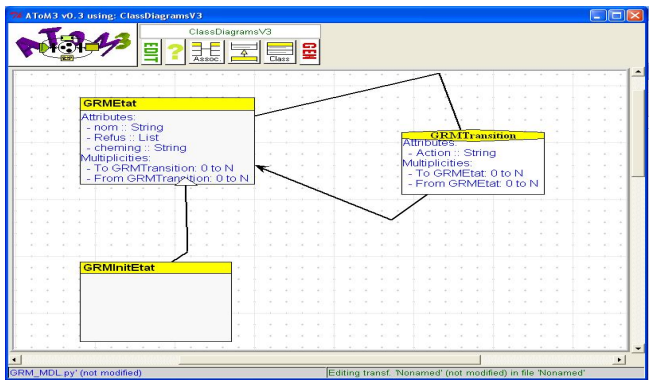

Figure 6: MRG meta-model

3.1.2.2 A tool for MRG Models

Based on the meta-model of figure 6, we have generated using ATOM^3 a tool for MRG models as shown in figure 7.

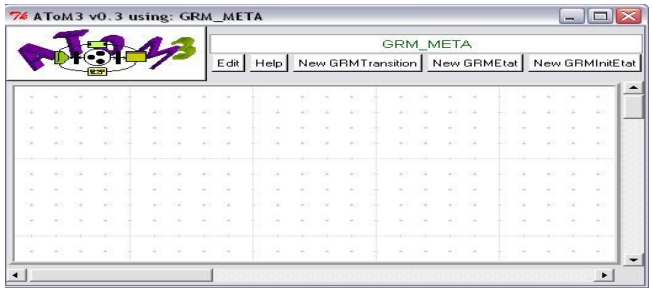

Figure 7: A tool for MRG models generated using ATOM3

3.2 Our Graph Grammar

We have proposed a graph grammar containing 13 rules organized in 5 categories.

- Rules 1 and 2 of figure 8 allow allocating paths to MLTS states; the paths will be used for performing the matching between MLTS and MRG states.

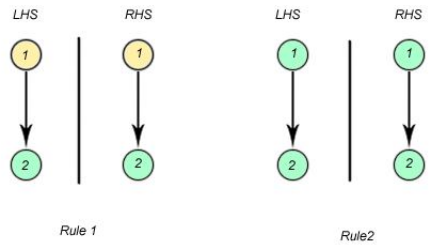

Figure 8: Rule1: Allocating paths to the first level states of MLTS.

Rule2: Allocating paths to other states of the MLTS model.

- Rules 3, 4, and 5 of figure 9 allow generating a nondeterministic graph which will be transformed later in a MRG.

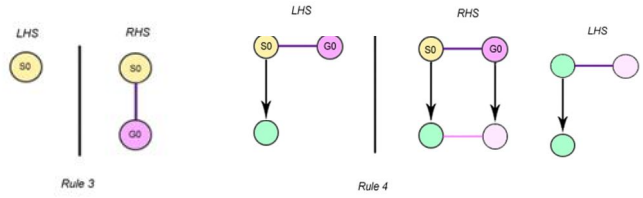

Figure 9: Rules generating a nondeterministic graph

- Rules 6, 7, and 8 of figure 10 allow generating determinist graph from the nondeterministic precedent graph.

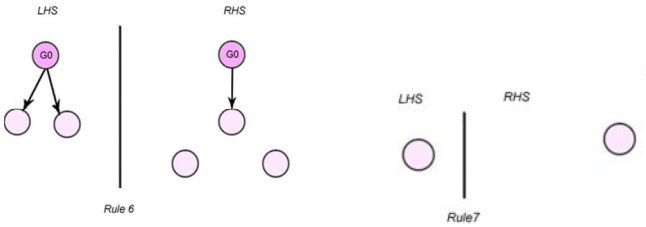

Figure 10: Rules generating a deterministic graph

Rules 9 and 10 of figure 11 attribute refusals to the states of the generated MRG.

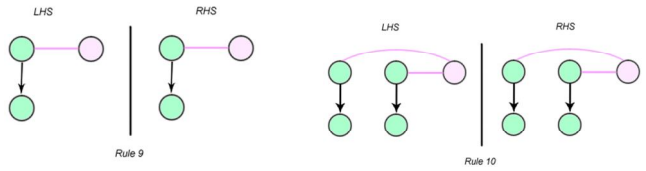

Figure 11: Rules for attributing refusals to the states of the generated MRG

- Rules 11, 12, and 13 of figure 12 allow deleting MLTS after the generation of MRG.

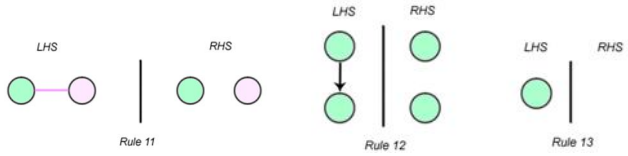

Figure 12: deleting MLTS after the generation of MRG

4 EXAMPLE

To illustrate our approach, we have applied our tool on a machine H whose behavior expression is H = a; b; STOP ||| (c; STOP [] a; STOP) and represented by the MLTS of figure 13 [11]. The mapping of the behavior expression to the equivalent MLTS model is performed using FOCOVE tool. More precisely, we have applied our tool on the MLTS of figure 13 and obtained automatically the MRG of figure 14.

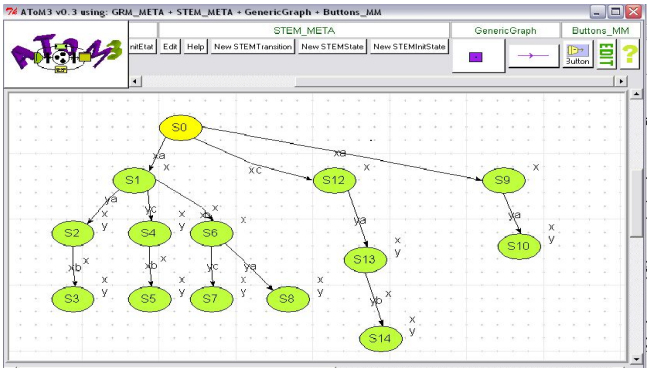

Figure 13: MLTS of the machine H in ATOM^3

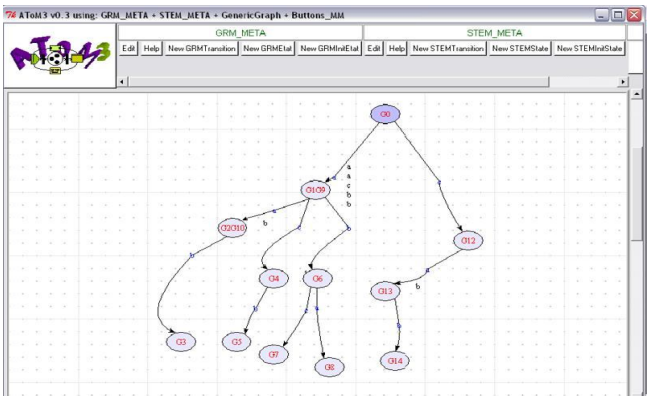

Figure 14: The final generated MRG using our tool

5 CONCLUSION

In this paper, we have proposed an approach and a tool for transforming MLTS models to MRG models using graph grammar since the input and output models are graphs. To

perform this transformation, we have proposed two meta-models; one for the input model and the other for the output model. Then based on these two meta-models, we have proposed a graph grammar that deals with the transformation process. The meta-modeling tool ATOM3 is used. We have illustrated our approach through an example. This work is concerned only with the transformation of MLTS models to MRG models. The MLTS models are obtained using FOCOVE tool from a LOTOS specification. In a future work, we plan to integrate the generation of MLTS models from LOTOs specification in our tool. The reduction of MRG models using graph grammars is also planned.

BIBLIOGRAPHIE

[1] AToM3 Home page, version 3.00, http://atom3.cs.mcgill.ca/

[2] Booch,G., J. Rumbaugh, and I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, 1999.

[3] De Lara J., Vangheluwe H.: AToM3: A Tool for Multi-Formalism Modelling and Meta-Modelling. Proc. Fundamental Approaches to Software Engineering, FASE'02, Vol. 2306. LNCS. Grenoble, France, 2002, 174-188.

[4] FOCOVE tool, http://focove.awardspace.co.uk/focove.html

[5] Kerkouche E., Chaoui A.: A Formal Framework and a Tool for the Specification and Analysis of G-Nets Models Based on Graph Transformation. Proc. International Conference on Distributed Computing and Networking ICDCN'09, LNCS, Vol. 5408, Springer-Verlag Berlin Heidelberg, India, 2009, 206–211.

[6] Kerkouche E., Chaoui A., Bourennane E., Labbani O.: Modelling and verification of Dynamic behaviour in UML models, a graph transformation based approach. Proc. Software Engineering and Data Engineering SEDE'2009, Las Vegas, Nevada, USA, 2009.

[7] Python Home page, htpp://www.python.org

[8] Rozenberg G.: Handbook of Graph Grammars and Computing by Graph Transformation, World Scientific, 1999.

[9] 9. Saidouni D.E: Sémantique de maximalité: application au raffinement d'actions dans LOTOS, (in French), PhD Thesis, University of Toulouse, France (1996).

[10] 10. Saidouni D.E., Ghenaï A., *Intégration des refus temporaires dans les graphes de refus*, In proceeding of NOTERE2006, ENSICA, Toulouse, France, June 6-9, 2006, Edition Lavoisier.

[11] Saidouni D.E. and Belala N.; Using Maximality-Based Labeled Transition System Model for Concurrency Logic Verification; The International Arab Journal of Information Technology, vol 2, No 3, July 2005.