A Specification and Validation Technique Based on STATEMATE and FNLOG

Olfa Mosbahi, Leila Jemni Ben Ayed, Samir Ben Ahmed, Jacques Jaray

A specification and validation technique based on STATEMATE and FNLOG

Olfa MOSBAHI, Leila JEMNI, Samir BEN AHMED and Jacques JARAY

Département des Sciences de l'Informatique, Faculté des Sciences de Tunis, Campus universitaire, 1060 le Belvédère Tunis Tunisie. olfa.mosbahi@fst.rnu.tn, leila.jemni@fsegt.rnu.tn, samir.benahmed@fst.rnu.tn, jacques.jaray@loria.fr

Abstract. The paper presents a specification technique borrowing features from two classes of specification methods, formal and semi-formal ones. Each of the above methods have been proved to be useful in the development of real-time and critical systems and widely reported in different papers [1], [2]. Formal methods are based on mathematical notations and axiomatic which induce verification and validation. Semi-formal methods are, in the other hand, graphic, structural and user-friendly. Each method is applied on a suitable case study, that we regret some missing features we could found in the other class. This remark has motivated our work. We are interested in the integration of formal and semi-formal methods in order to lay out a specification approach which combines the advantages of theses two classes of methods. The proposed technique is based on the integration of the semi-formal method STATEMATE [3] and the temporal logic FNLOG [7]. This choice is justified by the fact that FNLOG is formal, deals with quantitative temporal properties and that these two approaches have a compatibility which simplifies their integration [7]. The proposed integration approach uses the notations of STATEMATE and FNLOG, defines a various transformations rules of a STATEMATE specification towards FNLOG and extends the axiomatic of the temporal logic FNLOG by new lemmas to deal with duration properties. The paper presents the various steps of our integration approach.

Key words. Formal methods, Integration, Real-time Systems, Semi-formal methods, Specification, Temporal logic, Validation, Verification.

1 Introduction

Critical real-time systems are complex and require a high level of safety and reliability. To reduce this complexity and to reach a necessary degree of reliability and safety, it would be quite interesting to lay out a specification approach which simplifies the requirement description, deals with mathematical notations inducing verification and validation, and allows the description of quantitative temporal properties. Thus, it comes the idea of integrating formal [1], [5] and semi-formal

approaches in order to lay out a specification approach which combines the advantages of these two classes of methods. Semi formal methods are graphic, structural and user-friendly ; Formal methods are based on mathematical notations and axiomatic inducing proofs. In this paper, we propose a specification technique integrating STATEMATE [3] as a semi-formal method and the temporal logic FNLOG [7] as a formal one. Several reasons justifies the choice of these methods. STATEMATE [3] is a graphic formalism; covers the various aspects of a complex system. The temporal logic FNLOG [7] provides a requirement specification language that allows a concise expression of properties about quantitative properties. The proposed specification and validation approach introduces an integration method using STATEMATE and FNLOG notations and proposes transformation rules, and an extention of FNLOG axiomatic to reason about duration properties.

2 General view of the proposed specification and validation method

The proposed integration method [6] comprises mainly five great steps (Fig.1.):

Step 1. Description of requirements

This step consists on the description of system requirements by using FNLOG notation [7].They are liveness and safety properties specified by the system user.

Step 2. Specification with STATEMATE

This specification reduce system complexity which is broken up into a hierarchy of activities, control and primitives activities, with statecharts and activity-charts.

Development of the context diagram. which consists on the main activity, some external processes and flows of information connecting the system to its environment.

Decomposition of the system with activity-charts. The context diagram is broken up into a series of activities and date-store as well as control activity.

Specification with statecharts. The control activities are associated with statecharts which describe the behavior of their main activity.

Step 3. Transformation of STATEMATE primitives to FNLOG.

In this step we have proposed some transformations rules from Statecharts and Activity-charts specifications to logical formulae in FNLOG.

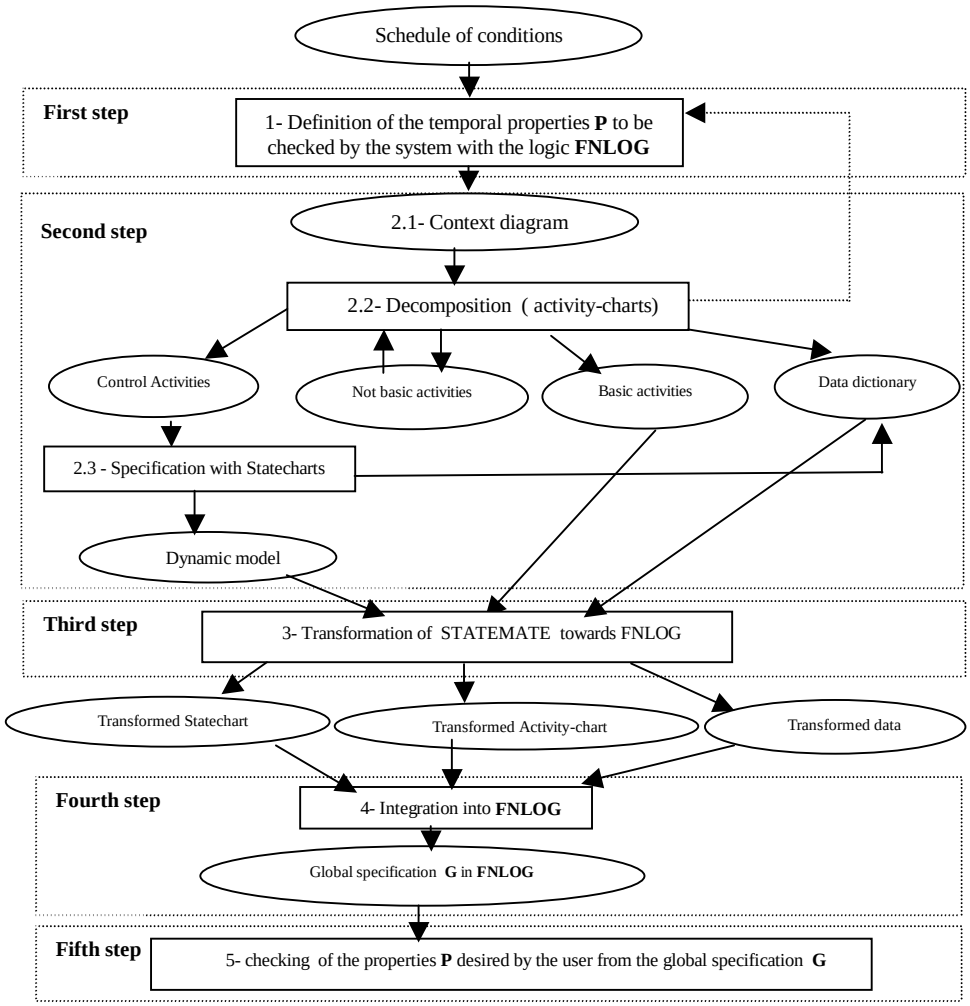


Fig. 1. Method of integration proposed using STATEMATE and FNLOG

Transformation from Statechart to FNLOG. The transformation of a Statecharts specification to an FNLOG specification is based on primitive's and on composition's transformations given in Table 1.

Statecharts	FNLOG
A state	An activity
An action	An event
An event	An event
Duration of an activity	Duration of an activity
Basic statecharts	Functions FNLOG
OR of two statecharts	Disjunction of two specifications FNLOG
AND of two statecharts	Conjunction of two specifications FNLOG

Table. 1. Transformation of statechart's primitives and structures to logic FNLOG

a- The event Timeout tm (E, T): This expression defines a new event which will be generated T units of time after the last occurrence of the event E.

b- The action Scheduled Sc! (G, T): This expression defines the execution of the action G, T units of time after the execution of the primitive Sc.
The transformation of these expressions is given in Fig. 2.

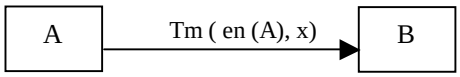
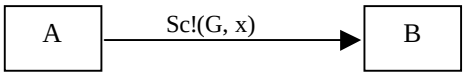
Statecharts	FNLOG
	$\odot_{t-x}$(init-A) $\rightarrow \odot_t$ (init-B)
	$\odot_{t-x}$(init-A) $\rightarrow \odot_t$ (G)

Fig.2. Transformation of time expressions from statecharts to FNLOG

Transformation of the activity-charts to FNLOG. The transformation of the activity-chart elements is illustrated in Table 2.

Activity-charts	FNLOG
An event	An event
A data	An expression of a number
An activity	An activity
A condition	A boolean expression

Table 2. Transformation of the activity-chart to FNLOG

Step 4. Composition in FNLOG.

It's the conjunction of FNLOG formulae found at each level of the decomposition obtained at the steps 2 and 3.

Step 5. Validation.

The fifth step consists on proving that the behavior specification found in the fourth step implies the system's requirements specified in the first step. These requirements are in general safety or liveness properties depending on time consideration [4]. However a problem holds in the verification of such duration properties with the existing axiomatic. To simplify such verification, we extend the FNLOG axiomatic [7] by introducing two new lemmas presented in the following :

Lemma 1. Duration over state sequence. The duration of an interval associated to a state sequence is the total length of the sub-intervals associated to each state.

We consider in Fig.3.three consecutive states A, B and C. A is followed by B and B is followed by C. A lasts x time units and B lasts y time units. The duration from the begenning of A to the begenning of C is x+y.

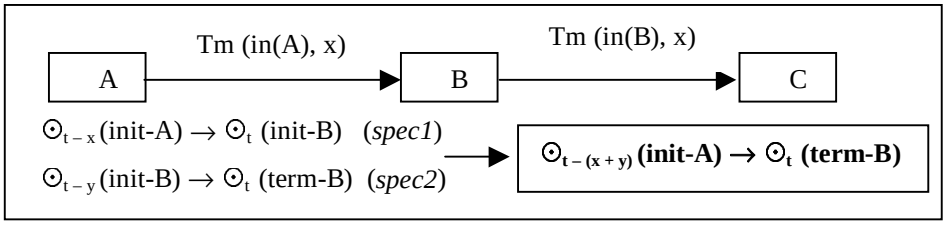


Fig. 3. Lemma 1.

Lemma 2. Atteignability. If a property ϕ holds in an interval [t-k, t] with t > k, then it holds also in the interval [t-j, t]with $j \geq k$.

$$\Diamond_t^{t-k}(\phi) \Rightarrow \Diamond_t^{t-j}(\phi) \quad \forall \; j \geq k$$

3 Conclusion

In this paper, a new technique for the specification and the validation of real-time and critical systems integrating the STATEMATE method [3] and the FNLOG logic [7] has been proposed. The most distinctive characteristic of our approach is the simple way of specifying real-time system's behavior dealing with functional and behavioral aspects. Also, the use of FNLOG has allowed the validation of specification in STATEMATE. We have illustrate our appraoch through an industrial example : a version of a computer controlled gas burner [6]. In order to develop formal technique for specifying and verifying real-time systems integrating STATEMATE and FNLOG, we have extended FNLOG axiomatic to reason about duration properties and proposed a transformation rules from STATEMATE to FNLOG [7].

References

[1] E.M.Clarke and J.M.Wing, *Formal Methods : state of the art and Future Directions*, ACMcomputing survey, Vol. 28, N0 4, P 626-643, décembre 1996.

[2] B.Cohen, *A brief history of formal methods*, *FACS Europe*, Vol. 1, N0 3, 1994.

[3] D. Harel, *Modeling Reactive systems with Statecharts*: *The statemate approach*, McGraw-Hill, USA, 1997.

[4] T.A.Henzinger, Z.Manna and A.Pnueli, *Temporal Proof Methodologies for Real-Time Systems*, 18 th Ann. Syym on Principales of Programming Languages, P 353-366, ACM Press, 1991.

[5] F.Jahanian and A.K.-L.Mok, *Safety analysis of timing properties in Real-time systems*, I.E.E.E Trans. On soft. Eng., Vol. 12, N0 9, P 890-904, 1986.

[6] O.Mosbahi, Une technique de spécification et de validation basée sur la méthode STATEMATE et la logique FNLOG, mémoire de D.E.A en informatique, FST, Tunis, Tunisie, 2002.

[7] A.Sowmya and S.Ramesh, *A Semantics-Preserving Transformation of statecharts to FNLOG*, Proc.14 th. IFAC Workshop Distributed Computer Control systems, Seoul, Korea, 1997.