INFORMATION
Volume 16, Number 1(B), pp.821-826

ISSN 1343-4500

Formal Verification of Basic DEV&DESS Formalism Using HyTech

Han Choi*, Sungdeok Cha*, Jae Yeon Jo**, Junbeom Yoo** *,
Hae Young Lee*** and Won-Tae Kim***

* *Korea University, Seoul, Republic of Korea*

** *Konkuk University, Seoul, Republic of Korea*

*** *CPS Research Team, ETRI, Daejeon, Republic of Korea*

Abstract

A hybrid system is a dynamical system reacting to continuous and discrete changes simultaneously. Many researchers have proposed modeling and verification formalisms for hybrid systems, but algorithmic verification of important properties such as safety and reachability is still an on-going research area. This paper demonstrates that a basic modeling formalism for hybrid systems, DEV&DESS, is an easy-to-use input front-end of a widely-used formal verification tool, HyTech. This paper transforms basic DEV&DESS models into linear hybrid automata and performs the HyTech model checking. We are now developing translation rules from DEV&DESS models into linear hybrid automata through various case studies.

Keywords : Hybrid system, DEV&DESS, Linear hybrid automata, HyTech, Model checking, Parametric analysis

1 Introduction

A hybrid system is a dynamical system whose behavior is a combination of continuous and discrete dynamics [3]. The discrete parts naturally model modes of operation of the system while the continuous dynamics model physical interactions with themselves or environment. Most embedded systems, consisting of digital control programs interacting with analog devices and environment, are examples of hybrid systems. Many researches[1] have been proposed to specify and verify hybrid systems [7, 6, 5, 9, 4]. Efficient application of these techniques, however, is rather limited to specific scopes, because of their inherent complexity.

DEV&DESS have been typically used as a fundamental formalism of other extended ones (e.g. CHARON [2] and ECML [8]), and verification of the hybrid system modeled in the DEV&DESS formalism using HyTech [7] is a good starting-point of developing modeling and verification tools for hybrid systems,

*Corresponding author: Junbeom Yoo, Email: jbyoo@konkuk.ac.kr

[1] See the website http://wiki.grasp.upenn.edu/hst/index.php?n=Main.HomePage managed by U. Penn.

which a research institute ETRI (Electronics and Telecommunications Research Institute) does. We transform a model of DEV&DESS formalism into a form of linear hybrid automata, which is an input front-end of the HyTech model checker, and perform the HyTech model checking such as safety verification and parametric analysis. The outline of the paper is as follows. Section 2 explains a hybrid system modeled with DEV&DESS. Transformation into linear hybrid automata and formal verification using HyTech are explained in Section 3. Section 4 concludes the paper.

2 Hybrid System Modeling

We modeled an atomic DEV&DESS model for a simple barrel filler system originated in [11] in order to explain our transformation and verification approach efficiently. The DEV&DESS formalism has no official visual representation and we used a graphical notation proposed by [10] for better understanding. In graphical notations, states mean the phases which are mutually exclusive state space [11], while solid and dotted transitions symbolize external and internal events, respectively. ⟨Fig.1⟩ denotes the DEV&DESS model for a barrel filler system. The formal definition of DEV&DESS model is skipped for the limit of space.

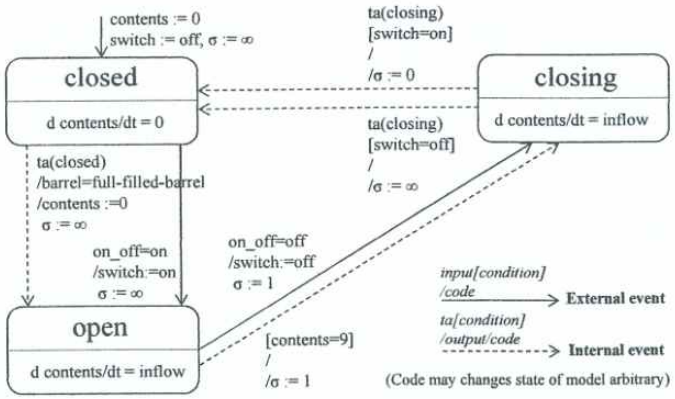

Figure 1: An atomic DEV&DESS model for the Barrel Filler

The system fills a barrel with a certain inflow rate while the valve of pipe is opened. When the barrel is filled up, it stops filling and puts the full-filled barrel out. This process involves changing the barrel with a new empty one. The valve of the system can be changed by the signal from the input port '*on_off*,' the external event, or by the state event occurring when '*contents*' becomes a cutoff value while the valve is opened. The cutoff value is set as 9 in order to prevent it from overflowing. The continuous variable '*contents*' represents the level of water, increasing its value continuously with a specific derivative. Its

derivative is the same as the continuous input '*inflow*' if the valve is opened. We also assumed the inflow of water is decreased as half, from 0.5 to 0.25, when the valve is closing. The variable '*switch*' is used for aiding understanding. 'σ' is renewed at every phase change, and it represents results of time advanced function.

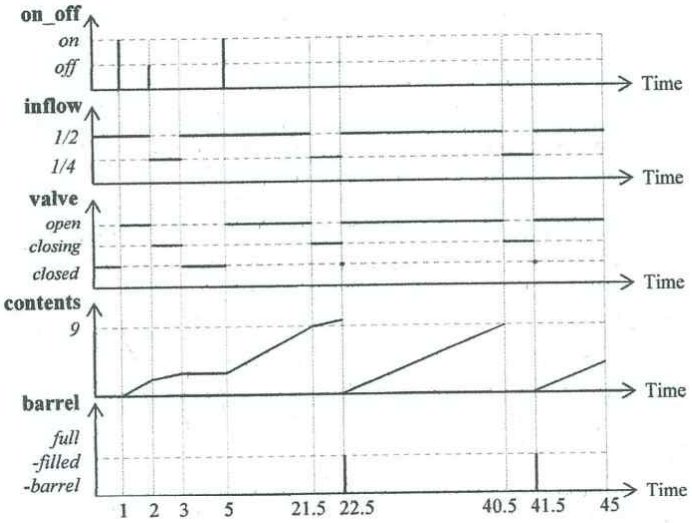

Figure 2: A trajectory for the Barrel Filling System

Fig.2 shows an example scenario for the barrel filling system, which we developed for demonstrating its correct behavior. The scenario above starts filling the first barrel throughout two '*on*' and one '*off*' signals during 22.5 time units. The state event occurs whenever the '*contents*' reaches at 9. It makes the model put a '*full-filled-barrel*' out, and the series of the same process are repeated immediately after a barrel is produced out.

3 Formal Verification Using HyTech

This section demonstrates that properties of barrel filler model can be verified with HyTech through translation into linear hybrid automata model. We introduce the translated linear hybrid automata first.

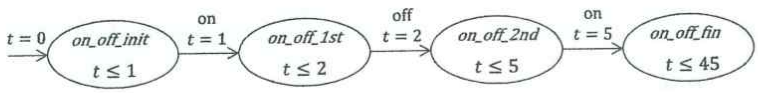

Figure 3: Automata for discrete input '*on_off*'

The main issue we had to resolve is that hybrid automata [1] does not have any notation to express I/O behavior, different from the DEV&DESS formalism. Our solution is to use an additional automaton, as described in ⟨Fig.3⟩, showing the same behavior with the input '*on_off*'. The variable 't' is a clock for global time, and the automaton coordinates with the barrel filler automaton through synchronization labels '*on*' and '*off*'.

The barrel filler model with explained scenario can be translated into a linear hybrid automaton as depicted in ⟨Fig.4⟩. The variable 'e' represents the elapsed time in DEV&DESS and compared with the variable 'σ' at every control mode to check occurrence of time events. ⟨Fig.5⟩ shows an excerption of the state trace from 0 to 22.5 time unit, simulated by HyTech.

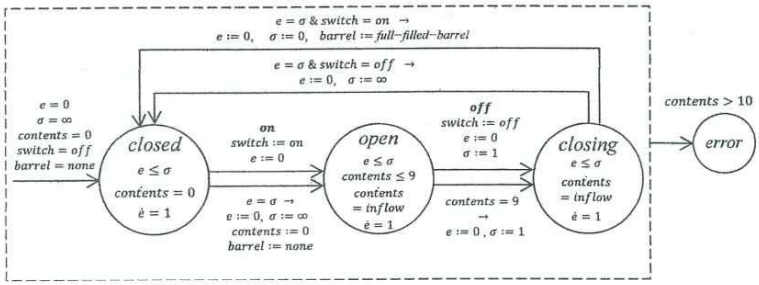

Figure 4: A linear hybrid automata model for the barrel filler model

Analysis commands for safety requirement:
(1) unsafe := loc[bf]=error;
(2) reached := reach forward from init reg endreach;
(3) if not empty(reached & unsafe)
(4) then print trace to unsafe using reached;
(5) else prints "Safety property satisfied";
(6) endif;

Verification of safety properties: The barrel filling system model has safety requirements such as "Content of a barrel should not be over 10 liter." We added an unsafe state '*error*' in Fig.4 and the safety property can be satisfied by finding paths to the unsafe '*error*' state. HyTech performs such analysis by computing all reachable states and checking whether the unsafe state can be reached or not form the initial state. A sequence of analysis commands above enables HyTech to print verification results out. By using HyTech, we were able to demonstrate that the model satisfies the safety requirement.

Parametric analysis: We defined a statement for parametric analysis as "When should we start closing the valve to avoid water overflow of barrel?" It

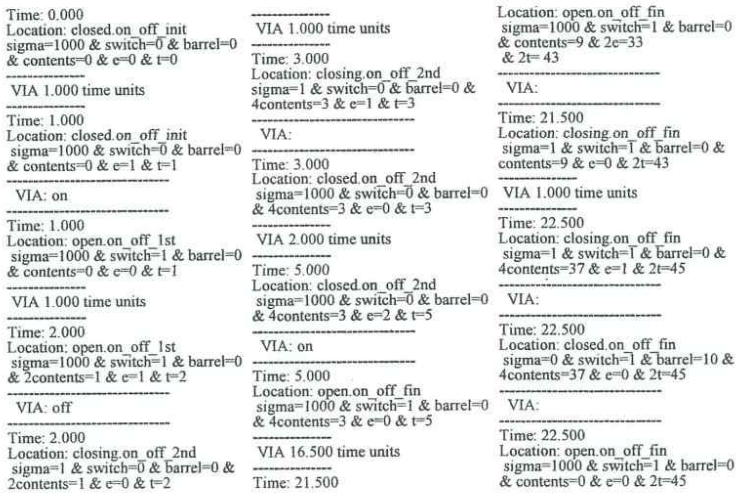

Figure 5: Reachable states for the barrel filler model (excerpted)

will find out constraints for the cutoff value which will result in water overflowing. As shown below, HyTech first searches the reachable states. Then the predicates for the necessary and sufficient conditions for visiting unsafe state are calculated by existential quantification using the reachable states and the unsafe state. The predicate '$4cutoff \leq 39$' for the safe cutoff value was produced, and we can derive the safe cutoff value 9.75.

Result of the HyTech execution:
(1) Spec. violated for parameter values
(2) Location: error.on_off_fin
(3) 4cutoff < 41 & 4cutoff > 40
(4) | 4cutoff > 39 & 4cutoff <= 81
(5) | 4utoff <= 83 & 4cutoff > 39
(6) | cutoff > 10
(7) Spec. satisfied for parameter values
(8) Location: error.on_off_fin
(9) 4cutoff <= 39

4 Conclusion and Future Work

DEV&DESS is a basic formalism for modeling hybrid system, but verification technique for DEV&DESS has not proposed yet. This paper aims to demonstrate that the HyTech model checker is a good starting point for developing efficient verification techniques for various modeling languages such as ECML.

We translated an atomic DEV&DESS model for a barrel filling system into linear hybrid automata, and then performed model checking of safety requirement and parametric analysis using the HyTech model checker, successfully. We are now developing translation rules and a mechanical translator for both single and coupled DEV&DESS models.

Acknowledgements

This work was supported by the IT R&D Program of MKE/KEIT [10035708, The Development of CPS (Cyber-Physical Systems) Core Technologies for High Confidential Autonomic Control Software]. This research was also partially supported by the National IT Industry Promotion Agency (NIPA) under the program of Software Engineering Technologies Development and the Engineering Research Center of Excellence Program of Korea Ministry of Education, Science and Technology(MEST) / National Research Foundation of Korea(NRF) (Grant 2011-0000978)

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

[2] R. Alur, T. Dang, J. Esposito, Y. Hur, F. Ivančić, I. L. Vijay Kumar, P. Mishra, G. J. Pappas, and O. Sokolsky. Hierarchical modeling and analysis of embedded systems. *Proceedings of the IEEE*, 91(1):11–28, 2003.

[3] P. J. Antsaklis, J. A. Stiver, and M. D. Lemmon. Interface and controller design for hybrid control systems. In *Hybrid Systems II, LNCS 999*, pages 462–492. Springer, 1995.

[4] E. Asarin, T. Dang, and O. Maler. The d/dt tool for verification of hybrid systems. In *Computer Aided Verification, LNCS 2404*, pages 746–770, 2002.

[5] A. Chutinan and B. H. Krogh. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In *Hybrid Systems: Computation and Control, LNCS 1569*, pages 76–90, 1999.

[6] C. Daws, A. Olivero, S. Trypakis, and S. Yovine. The tool kronos. In *Hybrid Systems III, LNCS 1066*, pages 208–219. Springer, 1996.

[7] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. Hytech: a model checker for hybrid systems. *Software Tools for Technology Transfer*, 1(1-2):110–122, 1997.

[8] H. Y. Lee, I. Chun, and W.-T. Kim. Dev&dess-based visual modeling language and tool for cyber-physical systems. (unpublished).

[9] I. Mitchell and C. J. Tomlin. Level set methods for computation in hybrid systems. In *Hybrid Systems: Computation and Control, LNCS 1790*, pages 310–323, 2000.

[10] H. Praehofer and D. Pree. Visual modeling of devs-based multiformalism systems based on higraphs. In *Simulation Conference Proceedings, 1993. Winter*, pages 595 –603, dec 1993.

[11] B. P. Zeigler, H. Praehofer, and T. G. Kim. *Theory of Modeling and Simulation*. Academic Press, 2000.