Diagnosis with Dynamic MSC Languages

Benedikt Bollig[1] and Stefan Haar and Loïc Hélouët[2]
[1] LSV, ENS Cachan, CNRS, INRIA, France
[2] IRISA, INRIA, Rennes, France

One of the key challenges when a fault occurs in a distributesd system is to find the original causes for the failure. In some cases, a seemingly unimportant phenomenon leads to a complete network collapse, and finding the root(s) of the dysfunction, as well as the way the fault has spread through the system is a key issue to correct the faulty system. Usually, running systems are equiped with probes, i.e. software or hardware mechanisms that record occurrences of some events. These logs are then used to retrieve faults. However, logs grow rapidly, and finding explanations without automated tools is almost unfeasible. Therefore, it is preferable to base diagnosis on a behavioral model of the system. The observations are then correlated to runs of the model to find explanations, that is the set of runs that might have lead to the observation (the logged events). Usually, an explanation is characterized by the existence of an embedding from the observation to the considered run.

This problem has been studied for interleaved models such as finite state machines [5], or concurrent ones such as Petri nets [1] or scenarios [3]. However, these models describe behaviors involving only a bounded number of processes. A challenge is then to propose diagnosis for dynamic and concurrent models. This work defines a diagnosis framework for a scenario based dynamic model, close to the dynamic MSCs proposed in [4]. This model is called *MSC grammars* [2], and consists in context free grammars that compose basic MSCs (descriptions of finite and asynchronous interactions between processes). As usual, these grammars contain an axiom, a set of non-terminals and rewriting rules, but terminals are basic MSCs. We furthermore allow renaming of processes in the terminal basic MSCs, which allows for the definition of behaviors over an arbitrary number of processes. The semantics of an MSC grammar $\mathcal{G}$ is the set of MSCs that can be derived from the axiom of $\mathcal{G}$. Within this setting, a derivation of an MSC grammar can be seen as a run ρ of our model, that produces an MSC M_ρ.

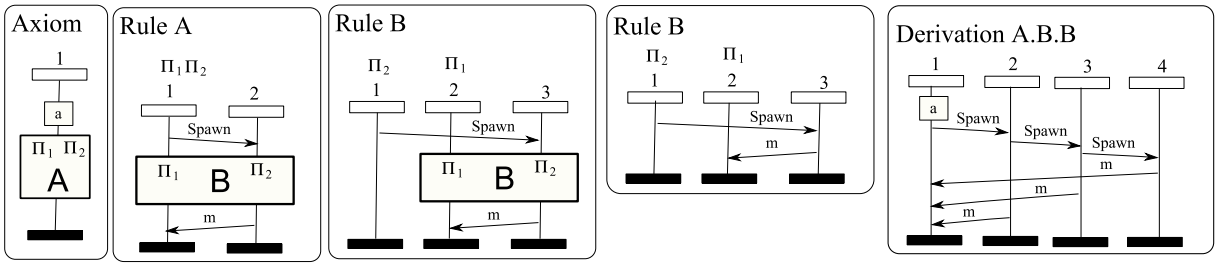

FIGURE 1. An example of MSC grammar and one of its derivations

We first show that the question whether an MSC M_ρ embeds an observation O is equivalent to the question whether M_ρ satisfies an MSO property ϕ_O computed from the observation. We then show that verifying whether an MSC grammar $\mathcal{G}$ satisfies an MSO formula ϕ (i.e. if there exists a run of the gammar satisfying ϕ) is decidable. The proof is obtained by building a tree automaton that recognizes all parse trees of $\mathcal{G}$, and then decorating the states of this tree automaton with sub-formulae of ϕ (this result relies on standard techniques, that were already used in [4]). This decorated automaton is then a *recognizer* for all MSCs in the language of the grammar that satisfy ϕ. This result provides an immediate means to perform diagnosis from MSC grammars, by construction of a recognizer for the formula ϕ_O attached to observation O. This tree automaton accepts a run of $\mathcal{G}$ if and only if M_ρ embeds the observation, and can be seen as a new grammar $\mathcal{G}'$ that generates only explanations for O.

Though the complexity of model checking MSO formulae may seem prohibitive for practical applications, this first step demonstrates clearly the feasability of diagnosis from MSC grammars. A dynamic diagnosis framework was already proposed in [6], but to the best of our knowledge, this work is the first diagnosis solution for an infinite state, dynamic and concurrent model with asynchronous communications.

Références

1. A. Benveniste, E. Fabre, C. Jard, and S. Haar. Diagnosis of asynchronous discrete event systems, a net unfolding approach. *IEEE Transactions on Automatic Control*, 48(5) :714–727, May 2003.
2. B. Bollig and L. Hélouët. Realizability of dynamic msc languages. In *Proc. of CSR 2010, Computer Science in Russia*, June 2010.
3. L. Hélouët, T. Gazagnaire, and B. Genest. Diagnosis from scenarios. In *Proc. of WODES'06*, 2006.
4. M. Leucker, P. Madhusudan, and S. Mukhopadhyay. Dynamic message sequence charts. In *FSTTCS'02*, volume 2556 of *LNCS*, pages 253–264. Springer, 2002.
5. M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D.C Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, 4(2) :105–124, 1996.
6. P. Baldan, T. Chatain, S. Haar, and B. K¨onig. Unfolding-based diagnosis of systems with an evolving topology, *Information and Computation*, 2010.