PROBABILISTIC HYBRID AUTOMATA WITH VARIABLE STEP WIDTH APPLIED TO THE ANAYLSIS OF NETWORKED AUTOMATION SYSTEMS

Jürgen Greifeneder and Georg Frey

Electrical and Computer Engineering Department
University of Kaiserslautern, Germany
Erwin-Schrödinger-Str. 12, 67663 Kaiserslautern
e-mail: greifeneder@eit.uni-kl.de

Abstract: Probabilistic Timed Automata (PTA) have successfully been applied to discuss problems specific to the field of Networked Automation Systems (NAS). This paper shows the transition from PTA towards Probabilistic Hybrid Automata (PHA) by introducing an event triggered variable time version of PTAs. The main idea is to increase accuracy and decrease state space of the underlying Discrete Time Markov Chains (DTMC) which are input for those probabilistic model checking algorithms for which these models should be used. For a better illustration of the advancement, the approach is applied to a typical example from the field of NAS.

Keywords: Time varying systems, Probabilistic Models, Quality, Delay analysis, Discrete-time models, Markov models, Networks, Modeling, Automata.

1. INTRODUCTION

In the last decade, the field of automation has been interspersed with apparently simple components borrowed from other technical fields, particularly those dealing with distributed systems. One of the driving forces thereby can be found in the rapid development of Ethernet or TCP/IP-based technologies resulting in a collapse of prices for the corresponding hard- and software. Apparently to that, the miniaturization went on as well as distribution of components and the need to communicate (not at least triggered by a still growing complexity of automation control algorithms). The hence composed systems aggregate control (i.e. automation) systems and network technologies. This fusion is called Networked Automation Systems (NAS) and represents a structure consisting of a number of programmable logic controllers (PLCs), several sensors and actuators, the network itself, and various input-output-Network-Devices.

The use of open communication platforms entails an increased and not previously predictable data flow, which may induce delays not present in classical structures. Furthermore, while the use of wireless networks (e.g. WLAN, Zigbee) offers a great opportunity in terms of variability and miniaturization, it adds a potential risk of data loss (loss of data in a NAS being defined as the non-arrival of data within a given time frame). All in all, this leads to a system structure which contains delay times, hard time bounds, stochastic distributions, deterministic choice connecting physical and computation processes.

There already is some work done to tackle the challenges yielded by NAS (Dingle et al., 2002 ; Marsal et al., 2005). However, these approaches are based on simulation, which implies that there is no guarantee for a full account of all possible evolutions. Yet, such a guarantee is essential for the analysis of NAS. Additionally, there are several formal approaches, but those are unfortunately unable to deal with the system structures described above or are restricted to producing worst case analysis, which can only lead to infeasible demands on the system's hard- and software. Therefore, a new modeling approach is proposed which can deal with time, stochastic distributions and probabilistic choice, namely probabilistic model

checking (PMC). When PMC is used for a NAS analysis, several advantages as well as disadvantages have to be considered. One of the advantages is that PMC is certain to cover all possible evolutions of a system instead of only a subset as is the case with simulation and testing methods. The disadvantages are best described by the price to be paid in terms of modeling abstractions, assumptions, and computational limits.

For this work, the considered systems are modeled on a discrete version of Probabilistic Timed Automata (PTA) and projected to Discrete Time Markov Chains (DTMC) by the use of an event triggered variable time step (which will be introduced in the next section). These DTMCs are integrated for the sake of the probabilistic model checking algorithms. The results presented within this work are calculated using PRISM, a model checker from the University of Birmingham (Kwiatkowska et al., 2002). For further illustration of design and consequences, a small, but typical NAS-example is given in section 3. Section 4 furnishes an in-depth explanation of some details of the time concept introduced in section 2; a short discussion follows in section 5. Finally, the concluding section 6 gives an outlook on further work.

2. MODELING OF TIME

Since physical processes are concerned, the correct functioning of the control system depends crucially upon real-time considerations. For that reason Alur and Dill developed the theory of timed automata (Alur and Dill, 1994). Therein, they used a dense-time representation, where times are represented by real numbers with an arbitrary accuracy tolerance up to which two numbers are considered as equal. However, using this approach it is still necessary to integrate different times to result in the same characteristic. This is accomplished through the use of clock regions or region graphs (actually the same thing). The method of digital clocks suggested by (Henzinger et al., 1992) could be used as an alternative to clock regions for a successful discretisation. Yet, the question arises why the discretisation is not carried out earlier in the process. This idea leads to a discrete model, which has each state attached with time information built of discrete and monotonically increasing (mostly integer) numbers. The Analysis of a timed system with discrete time creates certain phenomena, such as the problem of finding an appropriate time step and a set of initial states which would at least be able to cope with constant time drifts among non-synchronized subsystems (Greifeneder and Frey, 2006b). While in systems where signal changes are considered to occur synchronously with a clock signal (e.g. digital circuits) the problem of finding an appropriate time step is inherent, it still provides a potential source for state explosion. When it comes to continuous time physical interfaces (sensors), there is neither a lower nor an upper limit to this decision: if, on the one hand, a time step which is shorter than necessary to exactly model the fastest system change is used, the size of the model has the tendency to increase exponentially; if, on the other hand, a time step which is longer than necessary is applied, information loss will occur. One of the most important advantages of this modeling approach is that it can be transformed easily into an ordinary formal language via the addition of a new event to the set: a synchronize tick.

Note: It is not necessary to explicitly add the absolute clock information to each state as this information can be regenerated through a count of the number of states on a given path. A discrete time model usually requires a strong monotony of the integers representing the time, but in this case, the use of a fictitious time which only needs to be non-decreasing eases this restriction.

The time delay between two events is measured as the sum of all time steps elapsed between the two of them, that is: the point of time of each event is the same as its position in a series of states leading from an initial to a final state (this is to be called a path). To do the discretisation, it is impossible to state precisely certain about delays. On the other hand, the exact occurrence time of most of the signals is of lower interest, especially if any digital device forces the incidence towards a discrete time axis. Furthermore, it is irrelevant for most purposes whether an event A (e.g. the change of a sensors' value) occurred before or after another independent event B (e.g. the arrival of a package at a switch). In these cases, it really is an advantage (in terms of an enormous decrease in state space and calculation time) that only the fact that an event occurred within the last time step is recorded instead of the exact point of time within this time step. The best interpretation of this approach is to say that events occur in the specified order at real value times, but only the (integer) readings of the time value are recorded. Hence, by the way, the system model eliminates the occurrence of non-deterministic choice. All in all, this leads to the claim that the maximum length of a time step must be shorter than the time passing in between two consecutive significant events (a significant event is an event which has an influence on other processes in the system). From thereon, a modeling approach, which has already been successfully used for several case studies (Greifeneder and Frey, 2006a; 2006b), was derived. Yet, all of them suffered from the state explosion problem which raised exponentially to the chosen time accuracy. Therefore, the originally fixed time steps are this time replaced by an event triggered time step. This fuses the advantages of the authors' previous discrete time approach and the work done by (Alur et al., 1991). As in a DTMC the consecutive path only depends on the actual state (that is: the momentary state relies solely on the previous one), the information about the next time step's length must be determined before this state transition is executed. This is the real challenge to be taken up and changes the modeling task in a quite fundamental way (as will be demonstrated in chapter 4).