Synthesis of Control Software in a Layered Architecture from Hybrid Automata

Man Lin
Department of Computer and Information Science
Linköping University
S–581 83 Linköping, Sweden
linma@ida.liu.se
URL: http://www.ida.liu.se/~linma

Abstract. This paper deals with the synthesis of control software for hybrid systems specified as hybrid automata. Instead of generating the software from scratch, the synthesis is based on a generic layered software architecture which supports both periodic and event-triggered computations. The use of the layered software architecture as the framework for implementing hybrid controllers is motivated in the paper.

An automatic code generator HA2LS (from Hybrid Automata to Layered Systems) is introduced. HA2LS reads a specification in terms of hybrid automata and generates intermediate code that can be processed by the tools provided by the layered architecture. The generated software provides a clean interface for a control engineer to plug in the control algorithms. With externally supplied control algorithms and IO procedures, the synthesis of executable hybrid controllers can be completed.

The generated code can also be used for simulation purposes if it is generated from the specification of the complete system including the plant. The code generator HA2LS together with the software architecture substantially shorten the time to implement hybrid controllers from hybrid automata.

1 Introduction

By the term “embedded system” one usually refers to a computer program embedded in a physical system. Such an embedded system usually be used to control a mechanical, chemical or other kind of plant. The design and implementation of embedded systems usually involve contributions from both control and computer engineers.

In a complex embedded system, the plant can not be simply described as a set of differential equations. Instead, it can be described by a family of modes:
Synthesis of Control Software

153

each mode given as a set of differential equations. The control is switched among
modes depending on the state of the plant. Such a systems is called hybrid system
since it consists of continuous subsystems which are governed by the equations
and discrete subsystems (the mode switches). The controller for a hybrid system
is called hybrid controller. To implement this, the control engineers need to
provide the algorithms for each control mode and the computer engineers need to
implement the continuous control algorithms, the discrete controllers to control
the mode switches, and the interaction between the continuous and the discrete
control.

Although there exist many modeling languages [2,3,4,10] for modeling the
behaviour of hybrid systems, little work has been done to bridge the gap of high
level specification and implementation of hybrid controllers. One of the most
popular specification languages for hybrid systems is hybrid automata [2]. The
problem to be solved in this paper is: given hybrid automata as the specification
of a hybrid system, can the software for a hybrid controller be systematically
generated? The aim is to reduce or even eliminate the work of the computer
engineer in the process of developing a hybrid controller.

It is not easy to implement hybrid controllers using languages like C or ADA
since they give very poor support for algorithms expressed in a state machine
fashion. Software support and computer aided tools are therefore needed to assist
the implementation process. In this paper, we motivate a generic layered software
architecture [9] for implementing hybrid controllers. The languages and tools
developed within the architecture support periodic computations and discrete
computations needed by the hybrid controllers. We provide a systematic method
to synthesize hybrid controllers in the generic layered architecture from high level
specifications in terms of hybrid automata. The synthesis procedure is semi-
automatic. It includes automatic code generation with a tool called HA2LS and
the integration of the generated code with the user-supplied procedures. A user-
supplied procedure is either a control procedure which implements the control
algorithm for the continuous dynamics or an interface procedure which deals
with communication from sensors or to actuators. There are two uses for the
methodology. One is to generate code for the controller. Another is to generate
software for the complete system (including the plant) for simulation purposes.

The paper is organized as follows. First we provide a very brief description
of hybrid automata in section 2. Then, we introduce the generic layered architecture
and motivate its use for the implementation of hybrid controllers in section 3.
A method to synthesize the hybrid controller is provided in section 4. Finally,
the conclusions are given. The railway crossing system is used as an example
throughout the paper to illustrate the methodology.

2 Hybrid automata

A hybrid automata specification consists of a set of hybrid automata. Basically, a
hybrid automaton contains a set of locations and transitions among them. Each
location contains a set of linear differential equations to describe one mode of the