Mechatronics xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

On the use of model-based techniques for achieving multi-mode control architectures

Nagore Iriondo [a], Elisabet Estévez [b], Darío Orive [a], Marga Marcos [a,*]

[a] Departamento de Ingeniería de Sistemas y Automática, ETSI de Bilbao, UPV/EHU, Spain
[b] Departamento de Ingeniería Electrónica y Automática, EPS de Jaén, Jaén, Spain

ARTICLE INFO

Article history:
Received 21 January 2013
Revised 14 March 2014
Accepted 2 May 2014
Available online xxxx

Keywords:
Automation systems
Mode switched systems
Supervision
Model Driven Engineering
Meta-modeling techniques
IEC 61131-3 standard

ABSTRACT

Multi-operation mode processes are often described through a hybrid model using well-known formal descriptions such as hybrid automata. A multi-controller architecture has been proposed in the literature for coping with different dynamics. It is composed by as many control algorithms as operation modes and a supervisor that tracks the system's operation, switching to the appropriate control algorithm when a new operation mode is reached. This paper explores the advantages of model-based techniques to automate the implementation of the supervisor and its integration into the final real-time control system from the simulation dynamic model. In particular, the Model Driven Engineering approach has been followed in order to provide a framework to support the development cycle of the control system for mode switched processes. It uses information from the model of the process and from the control loops designed and using model transformation techniques the supervisor is automatically generated. In order to provide independency from the particular simulation tools used for defining the process model, meta-models are proposed for both, the process and the supervisor, as well as for the necessary transformations. A proof of concept framework has been designed for the target field of automation applications, generating the supervisor in PLCopen XML format for achieving vendor independency.

1. Introduction

In the last decades, hybrid systems have become an active area of research as a result of the rapid development and implementation of microcontrollers. Scientists from different communities like the control engineering and the computing sciences have studied this type of systems from their domain point of view, proposing different alternatives for system formulation as well as for control strategies. In general, hybrid systems could be defined as those that exhibit both continuous and discrete dynamics and they can be found in practically all type of industries. For instance, within the renewable energy field [1] proposes a hybrid system model for a solar air conditioning plant where the variable configurations due to the variations at the solar radiation are modeled as the discrete part of the hybrid system. Bhagwat et al. [2] uses a multi-linear modeling for studying operation transitions in pH neutralization reactions. Other example can be found in the position mooring for marine vessels. Nguyen and Sørensen [3] proposes the use of different controllers depending on the operational and environmental conditions.

The theory of hybrid systems offers formulation, analysis and synthesis techniques that, in general, need to describe different continuous evolution at different operation stages i.e. techniques that allows combining conventional continuous and discrete event models. A great variety of model structures has been proposed in the literature. A review of the different modeling, analysis and design techniques for hybrid systems is presented in [4].

Two main modeling approaches can be distinguished. From the discrete event systems (DES) perspective, models like automata or Petri Nets are extended with continuous state variables. These models have been mainly used in supervisory control theory in which the goal of the supervisor is to control operation transitions, such as start-up, shutdown, product or raw material changes [2,5–7,10]. On the other hand, the continuous systems perspective extends Ordinary Differential Equations (ODE) with discrete variables to include discrete phenomena [1,13].

In addition to have both continuous and discrete dynamics, many hybrid systems exhibit different operating conditions that may demand different control algorithms specially tuned to control the continuous dynamics. This could be due to nonlinearities

* Corresponding author. Tel.: +34 946014049.
E-mail addresses: nagore.iriondo@ehu.es (N. Iriondo), eestevez@ujaen.es (E. Estévez), dario.orive@ehu.es (D. Orive), marga.marcos@ehu.es (M. Marcos).

http://dx.doi.org/10.1016/j.mechatronics.2014.05.002
0957-4158/

inherent to the plant, to the very different dynamics of the process at the operating modes or to different performance demand in different operation points. In such cases, a solution may be the use of different control strategies at loop level as required by the operating modes providing a mechanism for switching the control algorithm when the operation mode changes [8,9].

Following this idea, a formal method for designing control systems for multi-modal processes using decomposition of the system in one control law per mode is presented in [9]. It provides formal methods for designing the modes including the different switch dynamics and the method is applied to a flexible manufacturing system.

Other approaches use optimal control [11] or adaptive control [4,12,13] for designing the supervisor which decides the switching strategy based on different process models, observers or state estimators. These control techniques have been commonly applied when process uncertainty or perturbation effects are considerable, using models for on-line estimation of the behavior based on process output measurements. Thus, these techniques require much on-line computation in order to estimate the most suitable control strategy.

Although different methods and theories have been proposed from the control design perspective, in most of the cases the implementation issues have not been addressed [1,9]. Notwithstanding this, there are interesting works in the literature that achieve code generation from simulation models. For instance [15] presents an experimental framework to test advanced controllers in heating, ventilating and air conditioning systems. It is based on MATLAB© environment and it prevents the writing of code to implement control algorithms in the physical system. It makes use of *Simulink*©, *Real-Time Workshop*© and *Windows Target*© toolboxes and standard PC hardware. Flordal et al. [16] addresses how to avoid collisions in multi-robot systems when the robots work together in a shared space. From simulation models, defined in commercial 3D robot simulation tool, they extract supervisors (finite state models) for ensuring that the deadlock situations are avoided. The finite state model is used to generate the PLC (Programmable Logic Controller)-code, although implementation issues are not detailed. Wang et al. [17] proposes a formal developing process for reliable PLC software.

All these interesting works address verification issues proposing an ad-hoc transformation process from the model to the code. However, current software engineering practices offer very powerful formalisms and methodologies that allow automating the generation of the final system from models, e.g. dynamic process model, control model, etc. In this sense, the Model Driven Engineering paradigm (MDE) relies on models for building systems [18]. In the last years, modeling technologies have experienced maturity enough as to offer significant influence in all aspects of software development, as much of the application code could be directly generate from models [18]. Meta-modeling techniques [19] allow defining modeling languages and model transformations and they provide the means for supporting MDE frameworks.

Looking at the field of mechatronic systems, the use of models is not new. Relevant works can be found in the literature that use modeling techniques in the industrial automation field. Most of them apply the Unified Modeling Language (UML) [20] and the Systems Modeling Language (SysML) [21] to the design and development of systems compliant to the IEC 61131 standard. Some of them propose the use of specific modeling UML profiles [22]. Secchi et al. [23] proposes a UML based methodology control software and physical components of mechatronics systems. Thramboulidis and Frey [24] analyzes UML, SysML and IEC 61499 function block Network (FBN) as mechanisms to define models from which the source code can be generated. Following this, UML and SysML are combined in order to model mechatronic systems using a 4-layered architecture [25]. Kernschmidt and Vogel-Heuser [26] comprises the concept for an interdisciplinary modeling approach, based on SysML, in order to analyze interdisciplinary change influences in production plants. A modeling methodology to support the design process of nontrivial mechatronic manufacturing machineries is presented in [27]. In Vyatkin [28], an analysis of the benefits of the IEC 61499 standard for different purposes is presented. These works conclude that UML and SysML are very useful during the first phases of the development cycle and the IEC 61499 FBN diagrams are relevant during the deployment and maintenance phase. Previous works of the authors have proposed a UML-based methodology for designing automation systems based on the IEC61131 standard [29]. Further work has included the use of industrial standards and guidelines to define domain modeling languages using CAEX [30].

There are other works that focus on automatic code generation describing first the system to be controlled using appropriate models that constitute the start point for the generation. Deligiannis and Manesis [31] presents a synthesis tool that guides the programming steps for generating the ST or SFC code for controlling a batch plant, entirely modeled as a global automata. The control actions associated to the states are all discrete actions over process actuators previously determined assuring specified criteria or restrictions. For the same type of batch or sequential applications, [32] generates code in Ladder Diagram (LD) language from UML-based state diagrams, taking advantage of the existence of process development software and validation tools for which UML modeling is available.

The goal of this paper is to apply model driven techniques to the field of hybrid systems. In particular, it focuses on the implementation issues of multi-mode systems, particularly when the continuous process to be controlled presents different dynamics depending on the operating conditions. The proposed supervision and control architecture follows the well-known multi-controller scheme [14] extended with multiple loops having different timing requirements. It is assumed that loop controllers are already designed and tuned offline and the goal is to generate the supervision architecture from the simulation dynamic model of the process and from the control architecture. Model driven techniques are used to automate the generation of the supervisor of mode-switched systems taking into account the timing requirements of the loop controllers actuating in every mode. The proposed approach uses model transformations and processing to generate a synchronous plan for every mode. In [33], the general formulation as well as the procedural steps for building the supervisor structure was proposed. The contribution of this paper is the definition of a model-based framework that automates the design of the supervisor for multi-loop, multi-mode switched processes. An initial transformation stores the information coming from the simulation models into a pre-defined structure following an ontological Meta-model of the process hybrid dynamics and loop control algorithms. This step makes the framework independent from the simulation tool. A proof of concept prototype has been developed targeting automation systems where the control equipment is the PLC, the most common controller used in industry applications.

The layout of the paper is as follows: Section 2 describes the general scenario to automate the supervision development, i.e. it contains the definition of meta-models and model transformation rules. Section 3 presents a prototype developed for industrial control systems having as target control system a PLC. The prototype not only generates the supervisor code but also the timing mechanisms for a correct execution of the control loops at modes. Finally, Section 4 illustrates the whole process through a simple but illustrative case study. The paper ends with some concluding remarks.

Please cite this article in press as: Iriondo N et al. On the use of model-based techniques for achieving multi-mode control architectures. Mechatronics (2014), http://dx.doi.org/10.1016/j.mechatronics.2014.05.002