Simulation Modelling Practice and Theory 16 (2008) 1077–1090

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

An admissible-behaviour-based analysis of the deadlock in Petri-net controllers

G. Mušič *, D. Matko

University of Ljubljana, Faculty of Electrical Engineering, 1000 Ljubljana, Tržaška 25, Slovenia

ARTICLE INFO

Article history:
Received 3 January 2008
Received in revised form 22 April 2008
Accepted 22 April 2008
Available online 4 May 2008

Keywords:
Supervisory control
Petri-nets
Manufacturing systems
Logic controllers

ABSTRACT

This paper addresses the problem of verifying the discrete control logic that is typically implemented by programmable controllers. Not only are the logical properties of the controller studied during verification, the behaviour of the overall controlled system is also examined. An approach that combines the calculation of the safety-oriented interlock controllers in terms of supervisory control theory (SCT), the corresponding calculation of the admissible behaviour of the system, and the specification of the desired system operation by Petri nets is proposed. A potential deadlock in the controlled system is then verified by taking the admissible-behaviour model as a process model. The analysis of the simultaneously operated supervisory-control-based interlock controller and the Petri-net-based sequential controller is performed with a C-reachability graph. The paper focuses on the calculation of the graph, and the approach is illustrated with an example of a simple manufacturing cell.

1. Introduction

While the functionality of programmable logic controllers (PLCs) is continuously expanding, discrete control logic remains the core of their operation. For a long time, PLCs have been programmed in a rather intuitive way, using specialised graphical programming languages, such as a ladder diagram [13]. Recently, much attention has been given to formal methods and their application in the design and verification of PLC programs. This is motivated by the growing complexity of control problems, the demands for reduced development times and the need to reuse existing software modules, on the one hand, and the increasing demands of society for a better control of technological risks, on the other [4,8].

Verification-based approaches deal with the formalization of the specifications and verification of the program against the formal specification [7]. The program passes the verification when the behaviour specified by the designer satisfies a set of properties. The properties can be checked on the control model only, or by considering a model (possibly a partial model) of the process. The latter is a more realistic approach to verification, called model-based verification [4].

To make the results of such a verification approach useful for the control, an appropriate model of the process under control is needed; however, this is not readily available in many cases. Different aspects of plant modelling for the purpose of controller verification have been extensively studied in [5,6,14]. The approach presented there enables detailed and systematic modelling of the controlled processes by employing a special modelling formalism.

In special cases, however, a suitable model for the verification can be obtained by considering a multilevel control structure and adopting a partially controlled plant on the lower level as a plant model for the verification of the upper level. Such a

* Corresponding author. Tel.: +386 1 4768 208; fax: +386 1 42 64631.
E-mail addresses: gasper.music@fe.uni-lj.si (G. Mušič), drago.matko@fe.uni-lj.si (D. Matko).

1569-190X/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2008.04.014

two-level approach is proposed in our previous work [10], and is further elaborated in this paper. In particular, such an approach can be used in applications involving PLCs, where a large portion of the control code is dedicated to safety measures, also called interlocks, and the corresponding part of the logic is sometimes referred to as the locking controller [18]. Assuming a two-stage approach, where the interlock logic is designed first and the sequential part is then added on top of that, the admissible behaviour of the plant imposed by the interlock logic can be adopted as a plant model for the verification of the sequential part.

In the presented approach the interlock part of the control logic is synthesized using supervisory control theory (SCT) [1,16]. The synthesis also gives a model of the admissible behaviour of the process, i.e., the behaviour of the process that complies with the given interlock specifications. The sequential part is then designed using Petri nets [9], which are used in the sense of a formal specification that is verified against the admissible model derived during the interlock synthesis. The basic property of interest is the absence of deadlock. A corresponding reachability-based analysis technique is proposed, which builds a C-reachability graph and enables the detection of any potential deadlock in the system that is controlled by a simultaneously operated supervisory-control-based interlock controller and a Petri-net-based sequential controller.

The motivation for the use of two modelling formalisms is twofold; firstly, the supervisory control theory is well suited to the interlock design. SCT is essentially safety-oriented, i.e., it enables the synthesis of a control policy that prevents any undesired behaviour of the controlled plant. In most applications, however, there are also requirements about the desired behaviour of the plant that should be enforced by the controller. The SCT-based synthesis and implementation of controllers that force the system to exhibit desired behaviour is difficult, although some related results are reported in the literature [2,12]. Secondly, the Petri-net framework provides an intuitive way of modelling the operation sequences, while the Petri-net-based supervisory control methods are less elaborate, especially in terms of event feedback, and few synthesis tools are available. The proposed combined approach exploits the advantages of both frameworks. Compared to other model-based verification approaches that are described in the literature (e.g., [3,6,17], see also survey papers [4,7], and the references therein), the main advantage of the combined approach is that it eliminates the need for an additional plant model for the purpose of verifying the sequential controller. The corresponding model is derived automatically during the interlock design stage.

The remainder of the paper is structured as follows. The proposed combined synthesis/verification approach is introduced in Section 2. The relation between admissible behaviour and the firing of transitions in the Petri-net model of operational procedures is explained in detail. The proposed deadlock-analysis technique is described in Section 3. The C-reachability graph is introduced, the required properties of the graph are examined, and a corresponding graph-calculation algorithm is presented. A simple example is given in Section 4 to illustrate the approach.

2. Combined synthesis/verification appproach

The aim of the verification is to answer the question as to whether a specification model is correct. This is done by examining various properties of the model, such as the stability, the absence of deadlocks, etc. In the presented case, the investigation is limited to the study of a single property, i.e., a check to ensure the Petri-net specification is not blocking the system operation.

The non-blocking property of a Petri net is traditionally regarded as the absence of deadlocks and closely related to the concept of liveness. A Petri net is said to be *live* when it is possible to ultimately fire any transition of the net by progressing through some firing sequence, starting from any marking that is reachable from a given initial marking [9]. A live Petri net guarantees deadlock-free operation.

When examining the Petri-net specification of a logic controller, the property of liveness is insufficient to ensure the non-blocking operation due to external inputs and outputs and their interrelations. The liveness of a Petri net is a necessary, but not sufficient, condition for the non-blocking operation of a related controller.

To examine the possible blocking of the controller the relation between the inputs and outputs must be taken into account. In other words, instead of analysing the 'open-loop' model of the controller a 'closed-loop' model of the control system has to be studied. Such an approach can be considered as a model-based approach to verification, according to the classification in [4].

2.1. Modelling a process under control

The key to the success of such a verification approach is a suitable model of the process under control. However, building such a model can be a difficult and cumbersome task. But in certain cases models developed during the initial stage of the control logic design can be used.

A two-stage approach to the design of the control logic is schematically shown in Fig. 1. It is a simplified version of the multistage approach proposed in [12]. One of the key points of the approach is that the specifications are split into two parts. The first part involves the prevention of undesired behaviour. It is composed of the so-called interlocks that implement measures to ensure safety, co-ordinate sub-processes, etc. The second part deals with the sequential specification and defines the prescribed order of tasks.