Science of Computer Programming ••• (••••) •••–•••

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

A process-theoretic approach to supervisory coordination under partial observation

Jasen Markovski [1]

Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

HIGHLIGHTS

- We propose a model-based engineering framework for coordination of complex systems.
- The framework relies on a process theory geared towards supervisory coordination.
- We defined controllability by means of the partial bisimulation preorder.
- We characterize event history-based and data observer-based supervisors.
- The framework is employed for coordination of a printing process function.

ARTICLE INFO

Article history:
Received 24 January 2013
Received in revised form 25 March 2014
Accepted 1 July 2014
Available online xxxx

Keywords:
Supervisory control theory
Controllability
Partial bisimulation
Partial observability
Process algebra

ABSTRACT

We propose a synthesis-centric model-based engineering framework for safe and nonblocking coordination of distributed components of complex high-tech systems. The framework is based on a process theory geared towards supervisory coordination and control of nondeterministic discrete-event systems under partial observation. Supervisory control theory deals with automated synthesis of provably correct models of supervisory controllers based on formal models of the system components and a formalization of the coordination requirements. Based on the obtained models, code generation can be used to implement the supervisory controllers in software, on a PLC, or an embedded (micro)processor. The proposed theory employs communication actions to distinguish between the different flows of information, i.e., observation of the system behavior and supervision by means of forwarding control signals. In addition, we consider the case of partial observation, where some behavior of the unsupervised system is hidden from the supervisory controller, e.g., due to lack of sensory information or internal inter-component communication. We revisit the notion of partial observation and treat it as a property of the supervisor, instead of conditioning the control requirements, which is a standard approach. By employing a behavior relation termed partial bisimulation, we are able to succinctly and transparently capture the notions of controllability and partial observability.

1. Introduction

Traditional software development techniques proved insufficiently flexible for development of quality control software, due to frequent changes in the control requirements and the ever-increasing demand for improved safety, performance, and ease of use [1]. This establishes the latter as an important bottleneck in design and production of complex high-tech systems,

E-mail address: j.markovski@tue.nl.

[1] Supported by Dutch NWO project ProThOS, no. 600.065.120.11N124.

http://dx.doi.org/10.1016/j.scico.2014.07.002
0167-6423/

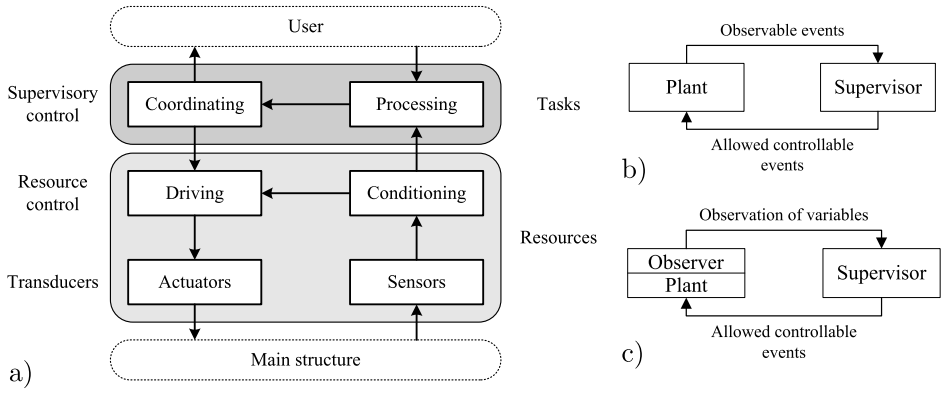

Fig. 1. (a) Supervisory control; (b) Supervisory control feedback loop with event-based (partial) observations; (c) Supervisory control feedback loop with data-based observations.

which gave rise to supervisory control theory of discrete-event systems [2,3]. This theory studies automated synthesis of models of supervisory control software, which coordinates high-level discrete-event system behavior. The supervisory controllers are synthesized based on the formal discrete-event models of the unsupervised system and a formal specification of the control requirements. Based on the synthesized models, the control software is generated automatically, or it is implemented on a PLC or an embedded (micro)processor.

Supervisory controllers ensure safe and nonblocking high-level system behavior by coordinating the components of the system. They rely on observations made regarding the discrete(-event) system behavior by using sensory information, as depicted in Fig. 1(a). Based upon the observed signals, the supervisory controllers make a decision on which activities are allowed to be carried out safely, and send back control signals to the hardware actuators. By assuming that the supervisory controller reacts sufficiently fast on machine input, this *supervisory control feedback loop* is modeled as a pair of synchronizing processes [2,3]. The model of the machine, referred to as *plant*, is restricted by the model of the controller, referred to as *supervisor*. The synchronization of the supervisor and the plant, results in the *supervised plant*, which models the behavior of the supervisory control loop as depicted in Fig. 1(b).

Traditionally, the activities of the machine are modeled as discrete events, whereas the supervisor is a process that synchronizes with the plant, enabling and disabling events by synchronizing or not synchronizing with them, respectively [2,3]. As a result, the supervisor comprises the complete history of the supervised system, i.e., it enumerates the state space of the supervised system [3,4]. The events are split into *controllable events*, which can be disabled by the supervisor in order to prevent potentially dangerous or otherwise undesired behavior, and *uncontrollable events*, which must never be disabled by the supervisor. The former model activities over which control can be exhibited, like interaction with the actuators of the machine, whereas the latter model activities beyond the control of the supervisor, like observation of sensors or interaction with the user and the environment.

The supervised plant must also satisfy the *control requirements*, which model the safe or allowed behavior of the machine. In addition, it is typically required that the supervised plant is nonblocking, meaning that it comprises no deadlock nor livelock behavior. To this end, every state is required to be able to reach a so-called *marked* or final state [2,3], which denotes that the plant can successfully complete its execution. The conditions that ensure the existence of such a supervisor are referred to as (nonblocking) *controllability* conditions [2,3]. An additional fundamental assumption is the power of observation of the supervisor, i.e., the inability to observe all events or states originating in the plant, due to lack of sensory information. The existence of the supervisor in this case is additionally conditioned on the property of *partial observability*, which intuitively ascertains that if the supervisor cannot distinguish between the observable part of two sequences of events or two states, then they should require the same control action.

The original supervisory control theory setting is language-based, employing automata with CSP-based synchronization [5] to model the supervisory control loop [2]. Partial observability necessitated treatment of nondeterminism as it introduced choices between two identical events that lead to different system states, which again was handled using traced-based notions [6]. Supervisory control theory of nondeterministic systems was studied on multiple occasions, again treated using trace-based notions, the most prominent of which is *state controllability* [7–9]. To enable a systematic algebraic study of supervisory control theory, a process theory geared towards supervisory control was proposed in [4]. This theory employs a behavioral preorder as a refinement relation between the supervised and the original plant to define controllability, inspired by the work of [10] that employs failure traces. The behavioral preorder is known as *partial bisimulation*, initially proposed in [11] as a foundation for a coalgebraic approach of supervisory control for deterministic systems.

In this paper, we extend the process theory of [4] with data and communication primitives in order to improve the modeling convenience and to model the supervisory control feedback loop more precisely. The introduction of the data elements enables us to specify the control requirements in a more compact and more transparent manner, which was inspired by analysis of specification documents in multiple industrial case studies [12–15]. The communication primitives are employed to distinguish between the separate flows of information between the plant and the supervisor in Fig. 1(b). Relying solely

Please cite this article in press as: J. Markovski, A process-theoretic approach to supervisory coordination under partial observation, Sci. Comput. Program. (2014), http://dx.doi.org/10.1016/j.scico.2014.07.002