Robotics and Computer-Integrated Manufacturing 27 (2011) 929–941

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

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

Coloured Petri nets and graphical simulation for the validation of a robotic cell in aircraft industry

Adriano José Cunha de Aguiar [a], Emília Villani [a,*], Fabrício Junqueira [b]

[a] Instituto Tecnológico de Aeronáutica, Praça Marechal Eduardo Gomes, 50 Vila das Acácias, São José dos Campos, São Paulo 12228-900, Brazil
[b] Escola Politécnica, University of São Paulo, São Paulo, São Paulo 05508-030, Brazil

ARTICLE INFO

Article history:
Received 9 April 2010
Received in revised form
18 February 2011
Accepted 14 March 2011
Available online 8 April 2011

Keywords:
Coloured Petri nets
Robotic cell
Graphical simulation
Validation

ABSTRACT

This paper proposes a mixed validation approach based on coloured Petri nets and 3D graphic simulation for the design of supervisory systems in manufacturing cells with multiple robots. The coloured Petri net is used to model the cell behaviour at a high level of abstraction. It models the activities of each cell component and its coordination by a supervisory system. The graphical simulation is used to analyse and validate the cell behaviour in a 3D environment, allowing the detection of collisions and the calculation of process times. The motivation for this work comes from the aeronautic industry. The automation of a fuselage assembly process requires the integration of robots with other cell components such as metrological or vision systems. In this cell, the robot trajectories are defined by the supervisory system and results from the coordination of the cell components. The paper presents the application of the approach for an aircraft assembly cell under integration in Brazil. This case study shows the feasibility of the approach and supports the discussion of its main advantages and limits.

1. Introduction

This paper approaches the problem of designing and validating supervisory systems for manufacturing cells with multiple robots, considering the integration of the robots among them and with other components of the manufacturing cell. The term "validation" is used with different meanings across the engineering areas. In this paper it is used to indicate the processes of analysing a system model by simulation in order to check that it complies with the needs and intent of the customer.

The motivation for this work comes from the aeronautics industry. Traditionally, the use of robots in aircraft structural assembly has been a challenge. The riveting operation is performed either manually or in dedicated machines. Dedicated riveting machines are used to assemble small parts, while the assembly of large fuselage sections is usually done manually. The solutions for automatic fuselage assembly may require the integration of two or more robots in the same cell. Furthermore, one major problem has to be overcome: the lack of accuracy of industrial robots. For this reason, an auxiliary system has to be used, such as a vision camera or photogrammetric equipment, and it interferes in the robots' trajectory in real-time [4,5]. As a consequence, the validation of the manufacturing cell behaviour must take into account the integration of the multiple components.

From the dynamic behaviour viewpoint, this problem is inherently hybrid. While the cell behaviour can be abstracted as a discrete event system, the behaviour of the robots must be represented in a 3D environment and modelled as continuous variable system. Although many commercial tools can easily model the robot trajectory and dynamics, such as Delmia from Dassault Systems and Robcad from Siemens, they are not adequate to represent discrete event dynamics of the other components. When the robot trajectory depends on the behaviour of the other components in the manufacturing cell, all of them must be modelled and simulated in an integrated environment in order to analyse the cell behaviour.

In this context, this paper proposes a mixed validation approach based on coloured Petri net and 3D graphic simulation for the design of supervisory systems. The coloured Petri net is used to model the cell behaviour at a high level of abstraction. It models the activities of each cell component and its coordination by a supervisory system. The graphical simulation is used to validate the cell behaviour in a 3D environment, allowing the detection of collisions and the calculation of process times. The validation approach guides the construction of the Petri net model and its conversion to a 3D simulation environment.

The main contribution of this paper is on the systematic merging of these two distinct and complementary simulation environments for robotic cells. The Petri net simulation provides the understanding of the causality among discrete actions of the

* Corresponding author. Tel.: +55 12 3947 5864; fax: +55 12 3947 6965.
E-mail addresses: adrianoaguiar@gmail.com (A.J. Cunha de Aguiar), evillani@ita.br (E. Villani), fabri@usp.br (F. Junqueira).

0736-5845/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2011.03.005

supervisor and discrete states of the cell equipment. The graphical simulation provides the understanding of the spatial behaviour of the cell robots. The systematic derivation of the graphical simulation model from the Petri net model provides the understanding of the relationship between the logic implemented in the supervisory system and behaviour in a 3D environment.

The proposed mixed approach is derived from an industrial case study: the design of a supervisory system for an aircraft structural assembly cell composed of two industrial robots. The case study illustrates the advantages and limits of the approach. This work is part of the AME Project, a Brazilian initiative supported by government and industry that aims at developing a cost effective solution for the automation of aircraft structural assembly process. AME stands for *Automação da Montagem Estrutural*, which means structural assembly automation.

This paper is organised as follows: Section 2 discusses related work; Section 3 details the steps and the tools of the verification approach; Section 4 presents the design and verification of the supervisory system for the manufacturing cell of AME Project. Finally, Section 5 draws some conclusions and discusses future work.

2. Related work

In order to contextualise the contribution of this paper, this section summarises and discusses published related work.

Flordal et al. [7] approach the design of supervisory systems for multiple robots. The paper presents a method to extract finite state model from the information in a 3D simulation environment. The workspace is divided into zones and the supervisory system coordinates the access of each robot to the zone. Only one robot may occupy a zone at a time, to guarantee absence of collision. The finite state model generates supervisors that avoid deadlock situations, based on the widespread approach of Ramadge and Wonham [18]. Comparing the work of [7] with the proposal of this paper, in our approach the discrete event model is generated using coloured Petri net instead of automata. The resulted model contains the data exchanged among the robots and supervisory system. This information is important because the robot trajectories are not previously known and arises from the interaction among the robots, the supervisory system and other equipment in the cell. Another difference is that the collision issues are detected in the 3D environment by simulation of different scenarios and considering the integration with the supervisory system programme. Although the absence of collision is not formally proven, our proposal provides more flexibility to model the robot trajectory and static objects such as the fuselage.

Richardsson and Fabian [19] advocate the importance of off-line verification for shortening product introduction time in manufacturing systems. According to the authors, off-line verification should combine formal verification and simulation in a 3D environment. The formal method guarantees that there is no deadlock in the system, while 3D simulation verifies collision issues. The authors define the information that is necessary for generating control programs and propose a method for the automatic generation of PLC programs from this information. The main concern is on the reuse of information. Comparing this paper, with the proposal of [19] does not deal with the problem of integrating the control programs in the 3D simulation environment.

Branicky and Chhatpar [3] propose hybrid automata to simulate force-guided robotic assembly. They investigate the viability of using three generic simulators for hybrid system: HyTech, Hybrid cc and CEtool. A problem in 2D is used as an example. The model of the robot behaviour is limited to movements in fixed directions. The control strategy is simple and results in a model with a few discrete states. The application of this kind of approach to represent the behaviour of an industrial manufacturing cell in a 3D environment is not feasible. Instead of using a unified language and tool to model all the components of the cell, our way of approaching complexity is to integrate simulation environments.

Lemaire and Maowhe [14] propose a multi agent approach to design robotised cells and apply it to an industrial measuring cell with a six revolute joint manipulator. The proposed approach is tested in RobCad with C++ algorithms. Dong et al. [6] and Ma et al. [15] approach the problem of using off-line programming for a spot-weld robot. Both works are based on RobCad. Some common problems, such as motion simulation, collision detection, calibration and path optimisation are discussed in their work. However, the problem of integrating multiple robots is not approached in these works.

Another group of related works aim at associating the execution of a Petri net to a graphical animation specific to a particular application, in order to make it understandable by experts in the application area. 'Playing the token-game' is not enough for understanding the behaviour of a complex system [13,23]. Some works in this field are presented below.

The work of Kindler and Páles [13] discusses some concepts to associate a Petri net with a 3D-visualisation. Basically, the proposal consists of associating some places with physical objects, which are called animation places. The animation requires the definition of two pieces of information: the object shape and its behaviour. When a token is produced on an animation place, an object with the corresponding shape appears and behaves according to a predefined animation function.

The Basic Real-time Interactive Tool for Net-based animation (BRITNeY) suite focuses on providing an animation based on the simulation of a CPN. It allows implementers to easily implement domain-specific animations using standard Java classes. It supports the CPN animation by annotating transitions with function call, which are executed whenever the transition occurs [23]. Comms/CPN is a CPN ML library, which makes it possible for CPN Tools to communicate based on TCP/IP with external application and processes. It allows the user to develop the graphical interface in any language and use Remote Procedure Call (RPC) to communicate with CPN [13,23]. ExSpect is another tool for modelling based on CPN. It allows the user to view the state by associating widgets with the state of the model, and to asynchronously interact with the model, also using simple widgets. In this way, it is easy to create simple user interfaces that support displaying information, but support for creating more elaborate animations is not readily available [13,23].

The work of Gomes and Lourenço [9] goes beyond the association of a graphical animation with a Petri net. It proposes a tool framework allowing the rapid prototyping of an animated synoptic application associated with an embedded system controller. The tool framework includes a Petri net graphical editor, an automatic code generator that converts Petri nets to C language, an animator that allows the association of the Petri net with a graphical representation and a synoptic application that integrates the execution of the C code associated with the Petri nets to the graphical presentation status of the process.

Although closely related to this paper, there are some important differences between these tools and the problem approached in this paper. These tools provide an alternative animation to the Petri net. The information usually flows in a single direction: from Petri net simulator to the 3D animator. The 3D animation does not interfere in the Petri net simulation, i.e., the 3D animation does not contain any information that is relevant for the Petri net execution. For the problem approached in this paper, the 3D animator contains a dynamic model of the physical system that