Available at www.MATHEMATICSWEB.ORG POWERED BY SCIENCE DIRECT

ELSEVIER Simulation Modelling Practice and Theory 10 (2002) 209–234

SIMULATION MODELLING PRACTICE AND THEORY

www.elsevier.com/locate/simpat

Object-oriented modeling and simulation of flexible manufacturing systems: a rule-based procedure

A. Anglani [a], A. Grieco [a], M. Pacella [a,*], T. Tolio [b]

[a] *Dipartimento di Ingegneria dell'Innovazione, Università degli Studi di Lecce, Via per Arnesano, Lecce, 73100, Italy*

[b] *Dipartimento di Meccanica, Politecnico di Milano, Via Bonardi 9, Milano, 20133, Italy*

Received 15 December 2001; received in revised form 22 May 2002

Abstract

Simulation by a software model, is one of the most frequently used techniques for the analysis and design of manufacturing systems. In the software engineering research area, the object-oriented approach has fully demonstrated to be an effective technique with respect to the design and implementation phases of complex software projects. Even if object-oriented programming has proven to be a powerful technique, a systematic design method should also be used in order to implement reliable software, in particular in the development of simulation models. This paper presents a new procedure to develop flexible manufacturing system (FMS) simulation models, based on the UML analysis/design tools and on the ARENA® simulation language. The two main features of the proposed procedure are the definition of a systematic conceptual procedure to design FMS simulation models and of a set of rules for the conceptual model translation in a simulation language. The goal is to improve the software development efficiency through a rule-based approach and to add some of the fundamental object-oriented features to the ARENA® simulation environment.

Keywords: Manufacturing discrete-event simulation; Object-oriented programming; Flexible manufacturing systems

* Corresponding author. Fax: +39-0832-320-279.

E-mail address: massimo.pacella@unile.it (M. Pacella).

1569-190X/02/$ - see front matter

PII: S1569-190X(02)00100-4

1. Introduction

A flexible manufacturing system (FMS) is an integrated production system composed by a set of independent machining centers (MCs). An automatic part handling system (PHS) interconnects the MCs to a group of part-storage locations such as loading/unloading positions and input/output buffers. An automatic tool handling system (THS) interconnects the MCs to a group of tool-storage locations as tool magazines, tool rooms, exchangers and spindles [10,16]. Either the PHS and THS mechanisms consist of one or more automated guided vehicles (AGVs) or transporters. A central supervisor (the FMS control software) monitors and manages the whole system. Three different kinds of object flows may be identified: the material flows (physical objects as parts, tools, pallets and fixtures), the information flows (abstract objects that describe the system status), and the decision flows (abstract objects that modify the system status) [12,22].

The FMSs are complex and expansive systems that require an accurate designing phase. In particular, it is important to examine closely the dynamic behavior of the different FMS components in order to predict the performance of the production system. The simulation is an essential tool for the design and the operational performance analysis of complex systems that cannot be easily described by analytical or mathematical models [19]. In particular, in the manufacturing field, the simulation can be used to configure the production system, or to select the more appropriate management rules [17].

The rapid expansion of the simulation software market accounts for the popularity of this type of tool [14]. Various discrete-event simulation languages oriented to the manufacturing environment are in use today. For example, ARENA® by System Modeling [21] and AutoMod® by AutoSimulation [19] are two of the most commonly used simulation languages at this moment both in the academic and in the industrial field. These tools consist of a collection of high-level constructs allowing the developer to achieve many of the tasks routinely required in the developing of a simulation project. Since their abstractions are focused to represent independent entity flows between processes, each of these tools is commonly referenced to as *transaction-oriented* discrete-event language. On the other hand, the *object-oriented* (O-O) simulation languages are based on a set of object classes that model the behavior of real system components. An example of O-O simulation language that has been adopted in manufacturing is eM-Plant® by Tecnomatix. An O-O software system operates by allowing the objects to pass messages. Each message represents a request to an object to perform certain functions.

Generally, an object model of a manufacturing system is designed by means of intermediate steps in which the real system components are described in an abstract mode through three different types of flows: materials, information, and decisions [6]. However, the implementation of the relative simulation model is a difficult task if the adopted approach does not guide the developers to associate their ideas of the system under study with the simulation tools. Indeed, in many of current simulation approaches, despite the existence of well-developed tools, it is often difficult to trans-