Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-3 (2015) 1502–1508

Product-Driven Generation of Action Sequences for Adaptable Manufacturing Systems

Nadine Keddis Gerd Kainz Alois Zoitl

fortiss GmbH
Guerickestr. 25
80805 Munich, Germany
{keddis, kainz, zoitl}@fortiss.org

Abstract: In times of fast changing markets and short product life-cycles manufacturing systems have to be adaptable and able to support a variety in products and product volumes. Production has to be product-driven and switching between different products should be possible with little manual intervention. We suggest an action sequence generation approach that tailors the control programs of resources to product needs. The approach requires a model of the available resources with their capabilities and internal material flow, the material flow between resources as well as a product description. An action sequence can then be generated out of these models and later translated into an executable action sequence. The action sequence can be automatically downloaded and executed on a resource. The approach is evaluated on an educational production system with industrial components.

Keywords: Flexible manufacturing systems, Information technology, Machine code, Model-based control, Planning

1. INTRODUCTION

After the peak of mass production in 1955 there has been a shift towards mass customization (Iacocca Institute, 1991; Hu et al., 2008; Koren, 2010). The revolution of manufacturing is continuing with a trend towards personalized products (Koren, 2010). This revolution is driven by customer demands that vary over time. Current markets are saturated (Westkämper and Decker, 2006) with a higher supply than demand which forces suppliers to be more flexible in their processes to be economically viable. Manufacturing systems nowadays have to support heterogeneous products with low volume in order to be able to compete in highly competitive markets. Changeable manufacturing systems as described by Wiendahl et al. (2007) are necessary to cope with turbulent environments. They are characterized by being adaptable, intelligent, and versatile (Zor et al., 2010).

The role of IT in manufacturing has increased (Vogel-Heuser et al., 2009). However, current IT systems are still too inflexible (Zor et al., 2010; Sauer and Jasperneite, 2011) and require a lot of manual efforts when changes in the production are required (Zäh et al., 2010). Especially, scheduling and planning software requires a lot of manual effort to set-up (Klöpper et al., 2009). Some approaches are available that can automatically generate schedules. Nevertheless, they cannot react to dynamic changes of the scheduling problem and are only suitable for static environments (Cheeseman et al., 2005). Additionally, automatic transformation from schedule to machine code would be beneficial to complement such tool chains.

This paper proposes an approach for automatic generation of action sequences for production resources to handle current problems in the manufacturing domain. Production resources in this context refer to hardware entities that can execute production steps and are called resources throughout this paper. Resources have to provide predefined interfaces that give access to available control code implementations to execute actions (Zoitl et al., 2013). Additionally, the approach is based on models of resource capabilities and material flows. Action sequences for the production can be automatically generated using these models. The action sequence considers available material flow information to ensure that it is valid. For this, we differentiate between external and internal material flow. External material flow describes the material flow between different resources, whereas internal material flow is limited to the material flow within a resource. The first contribution of this paper is the automatic generation of action sequences that consider product requirements as well as internal and external material flow. The second contribution is an approach to automatically make action sequences executable. The benefit of this is the reduction of manual effort when switching between different products and factory setups. A factory setup refers to a set of available resources and their relations within the factory. This is demonstrated using a modular production system.

The remainder of this paper is structured as follows: Section 2 gives an overview of available work in the field of action sequence generation. In Section 3 a brief introduction of the execution of action sequences on resources is given. The required models to enable the approach are explained in Section 4. In Section 5 the generation

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.06.299

process is introduced in detail. Section 6 describes the experimental setup and the evaluation of the approach. Finally, Section 7 concludes the paper.

2. RELATED WORK

There has been much work in the past years to increase software quality and re-usability of control software, e.g., (Sünder et al., 2006; Eckert et al., 2012; Sorouri et al., 2012; Zoitl and Prähofer, 2012). The approaches mainly try to develop design patterns and improve the development of control software to achieve this. However, they focus on control software of a resource only and not on generating the equivalent action sequences. Additionally, they target rigid control software and focus on how to manage variability for different resources. In our case, we also consider that different products and their variants might be produced on the same resource. The resource must be able to support different action sequences and be able to work in different factory setups containing different resources. The goal is to allow the execution of different operations on the same resource depending on the desired product. Nevertheless, such modular concepts for control software are necessary to enable such an approach. They act as a starting point for our approach.

There has been much work in the field of recipes in manufacturing as well. They are similar to action sequences but more static and defined manually for each resource.

For fixed factory setups and plant structures, standards like ISA 88 were developed by the International Standardization Association (Brandl, 2006). ISA 88 separates the description of production steps from manufacturing resources. The standard provides means to coordinate different resources based on a predefined recipe for a product. Originally this was developed for the process manufacturing, but similar concepts for the discrete manufacturing exist, e.g., PackML guideline (Arens et al., 2006).

The NAMUR association gives a recommendation for recipes and defines general requirements of recipes in NE033 (NAMUR, 2003). It gives a recommendation on how to structure recipes for discontinuous processes and mainly focuses on the process industry. It is a good foundation for introducing recipe-based operation. Nevertheless, it does not propose how these recipes can be used from a planning and scheduling perspective.

In the field of agent-based planning and scheduling there has been plenty of work to develop suitable platforms and planning strategies, e.g., (Bussmann and Schild, 2001; Gabel and Riedmiller, 2008; Leitão, 2009; Alexakos et al., 2012; Lepuschitz et al., 2013). Holonic manufacturing tackles similar problems. They are used for coordination and sequencing of manufacturing resources (McFarlane and Bussmann, 2003; Lohse et al., 2005). Such approaches focus on negotiating schedules and allocating resources for the operations. The agents execute available code depending on the negotiated schedule.

Zäh et al. (2008) propose an approach to generate schedules using different capabilities. However, they do not discuss how the generated schedules can be translated into machine-readable control commands that can be used to automatically execute them on resources.

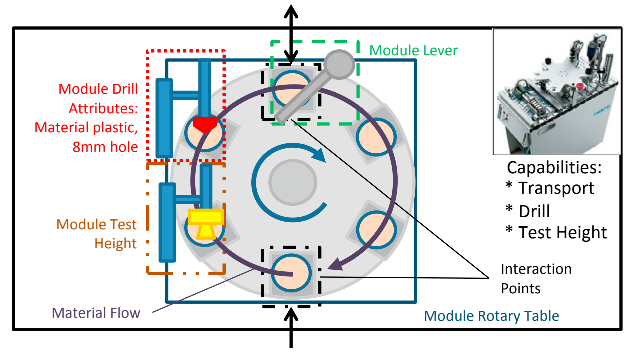

Fig. 1. Resource model for a processing station with capabilities and internal material flow information.

3. BACKGROUND

The starting point for our approach is a modular programmable logic controller (PLC) program. This is described in detail in the work of Zoitl et al. (2013). We just give a brief overview here. Every component is represented by a module comprising all functions required to control it as well as a set of interfaces to access the functions. Additionally, there is a recipe controller that is responsible for coordinating the execution of the module programs. Moreover, each program has a startup code, called program outline, which includes the basic control of a resource with its locking mechanisms. It also triggers the recipe controller that executes the currently loaded action sequence. Each command in an action sequence consists of a flag in the beginning to indicate whether this is a branch instruction or not. This is followed by a set of flags to determine the branch. Afterwards, the command is specified using an identifier for the used module, interface, and parameters. The execution of action sequences is similar to how CPUs function.

4. SYSTEM MODELING

In order to generate action sequences based on the product requirements, a factory-independent description of the product has to be provided. Additionally, a description of the factory and the available resources is necessary. As proposed in previous work (Keddis et al., 2013, 2014) the resources are modeled based on their capabilities. Capabilities of each resource express the processing steps that are supported and can be executed by this resource. We just give a brief example of the resource model, for more information see (Keddis et al., 2013). In addition to modeling capabilities, the internal material flow of the resource has to be modeled. The internal material flow describes the internal structure of a resource and how it is actually built to be able to support the processing steps. Moreover, a factory model is provided that represents how the different resources interact within the factory. Combining this information with a model for the control software of the resource can then enable an automatic generation of a product-based action sequence that can be directly executed on the resource. In the following, the different required models are presented in detail.