CIRP Annals - Manufacturing Technology xxx (2018) xxx–xxx

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: http://ees.elsevier.com/cirp/default.asp

Model-based design and simulation of smart factory from usage and functional aspects

Hitoshi Komoto (2)*, Keijiro Masui

Advanced Manufacturing Research Institute, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan

ARTICLE INFO

Keywords:
Computer aided design
Simulation
Smart factory

ABSTRACT

Usage and functional aspects of smart factory are used to express its design concepts in terms of the roles and activities of stakeholders and the functions of system elements. This study proposes a model-based framework for developing use cases of smart factory, which satisfies two competing requirements; easy-to-use composition of design concepts from these aspects and simulation-based analysis of design concepts. The framework can simulate prognostics services for machine tools connected with the service platform, and those for machine tools equipped with self-prognostics functions and it can analyse their differences regarding the performance and its progress of prognostics services.

1. Introduction

Manufacturing systems should be more rapid, flexible and sustainable to cope with dynamic changes in market needs and the manufacturing environment while maintaining the quality of manufactured products [1]. Various intelligent technologies for a smart factory such as digital enterprise-level planning and control [2] and cloud-enabled prognosis [3] are studied (see the overview in Ref. [4]), and some have been introduced in practice (e.g. continuous maintenance [5]). These novel technologies intensively use a digital representation of machines (i.e. prepared models in the design stage and collected data in the use stage) that interact with one another on both the physical and cyber spaces of a smart factory [6]. By adopting these technologies, the manufacturing industry pursues new forms of value propositions to increase its market opportunity. Particularly, manufacturers of machine tools are concerned with how machine tools and the digital representatives of machine tools are used to effectively satisfy the needs of users (including operators, maintenance experts and managers of production facilities).

Use cases (as descriptions of such value propositions) are prepared to disseminate such value propositions to the manufacturing industry. These use cases vary not only in content but also in the levels of abstraction (or concreteness), as they are highly dependent on the knowledge and interest of those who prepare such use cases. Recently, the design and analysis of use cases in the manufacturing industry has been an important research topic in the field of engineering design and particularly in industrial product-service systems [7]. To guide the development of use cases, classifications and aspects (viewpoints) of use cases have been proposed in the reference architectures for smart manufacturing (e.g. Refs. [4,8]). Among the proposed aspects in Ref. [8], such as business, usage, function and implementation, the usage and function aspects guide the decomposition of a use case to a set of concrete activities and the description of roles and interactions among the devices, applications and stakeholders appeared in a given activity. Such activities, roles and interactions in a use case clarify the corresponding design concept of a smart factory.

This paper proposes a modelling and simulation framework for the design and analysis of a smart factory based on the usage and functional aspects. This framework is proposed to apply the concept of model-based development, which is used in the early stage of complex mechatronics systems development (e.g. electric cars and photocopiers) [9], to the development process of a smart factory. Particularly, the framework clarifies the roles of participants and functions of the system elements in potential design concepts and evaluates the behaviour and performance of these concepts before cost-sensitive decisions (e.g. specific implementation decisions) are made in the development process.

The proposed framework is shown in Fig. 1. In the framework, an executable model of a use case for performance evaluation is built with a system-level model and activity-specific models. A system-level model consists of a set of activities, each of which specifies the roles of participants and the functions of system elements that appear in the use case. Designers can build, modify, understand and explain this model without the knowledge of specific programming languages. In contrast, modelling experts (in collaboration with designers) describe activity-specific models reusable in various use cases, which include the detailed behaviour of participants and system elements which interact in an activity using a specific language to simulate a smart factory. This study specifies and verifies the essential functions of the framework using a prototype system developed for this purpose.

The present framework is used for performance evaluation of data-intensive prognostics of machine tools connected with a data collection and analysis platform in context of various maintenance strategies (e.g. time-based and condition-based maintenance

* Corresponding author.
E-mail address: h.komoto@aist.go.jp (H. Komoto).

https://doi.org/10.1016/j.cirp.2018.04.025
0007-8506/

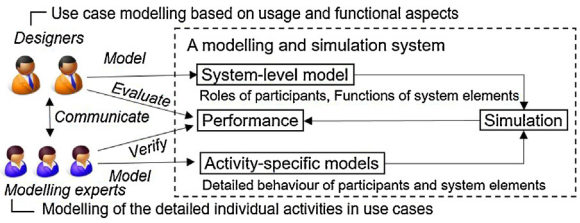

Fig. 1. Model-based development of a smart factory based on usage and functional aspects.

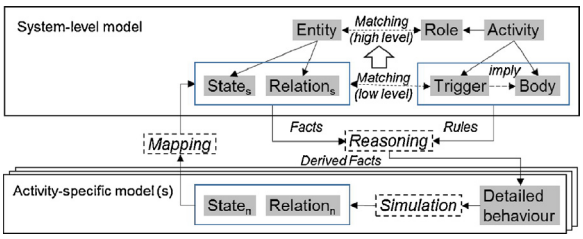

Fig. 2. The overview of the proposed model.

[10,11]). Such prognostics services for the facilities in a smart factory are essential in use cases belonging to the application scenario value-based service (VBS) [12], which represents new value propositions that use operational data of facilities collected in the use stage. The performance of prognostics services may depend not only on the employed prognostics algorithms but also on the type of service providers with different data accessibility (e.g. users managing their own machine tools, platforms connected with machine tools all over the world or machine tools with self-prognostics functions). This study shows that simple modification of a system-level model is sufficient to deal with diversity of such service offerings and that their performance is quantitatively evaluated by dynamically controlling the execution of the activity-specific models connected to the system-level model. Such a performance evaluation of a variety of prognostics offerings has not been reported in related work (e.g. Refs. [10,11]).

The structure of paper is as follows. The model and simulation mechanism used in the framework is presented in Section 2. In Section 3, prognostics services of machine tools integrated in various maintenance strategies are modelled and analysed with the framework. In Section 4, the modelling and simulation techniques in engineering design relevant to the present work (e.g. Refs. [7,9,13,14]), are reviewed and compared with this work. The summary and conclusions are presented in Section 5.

2. The proposed modelling and simulation framework

2.1. Background

The model studied in the paper is based on the model proposed in the reference architecture for smart manufacturing [8]. It describes the usage aspect of a smart factory as a set of activities, each of which has mainly three elements. First, the trigger and consequence (body) of each activity define causal relations among the activities. Second, each activity is described with roles of participants, which allow flexible configurations of a smart factory in terms of how participants can assume certain roles in various activities (e.g. data analysis). Third, each activity has functional and implementation components, which indicate the allocation of functions to the elements of a smart factory (e.g. assets, applications and a platform). The details of these components are not evident from the usage aspect, but can be described in the functional and implementation aspect models.

2.2. Model formulation

Based on the model proposed in Ref. [8], the present model of a smart factory is formulated from its usage and functional aspects. As explained in Section 1, the model consists of a system-level model and activity-specific models. Furthermore, an interface is defined between the system-level model and activity-specific models. The overview of the model and examples of the model description are shown in Figs. 2 and 3, respectively.

First, the system-level model consists of a set of entities, activities, and their interrelationships. Both participants (e.g. operators of machines (as assets) and domain experts) and elements of a smart factory (e.g. machines) are regarded as entities, because of the flexible interchanges between the participants and elements of a smart factory (e.g. operators with machines as well as machines with domain experts). This can come about, for instance, owing to further automation of labour-intensive activities in a smart factory. An activity is defined by a trigger and a body. A trigger is defined by a logical statement about the state of entities and the relations among the entities, which are indicated by State$_s$ and Relation$_s$ in Fig. 2. As shown in Fig. 3, the states and relations defined here are characterized by predicates, and entities are variables in predicate logic (e.g. *connected-to* as a predicate and *?x* and *?z* as variables in Fig. 3). Similarly, functions of entities are predicates decorating entities (e.g *connectable* in Fig. 3). This means that whether an entity has a specific function or not is evaluated by the state of the entity, which may change during its life time. A body is defined by a logical statement including the name of the activity followed by variables indicating the role of entities. These roles allow flexible assignment of entities to activities (the high-level matching in Fig. 3). It is done by regarding entities as potential values of the variables in the trigger and body of activities (the low-level matching in Fig. 3). The variables will be dynamically bound with the instances of entities during simulation (see Section 2.3).

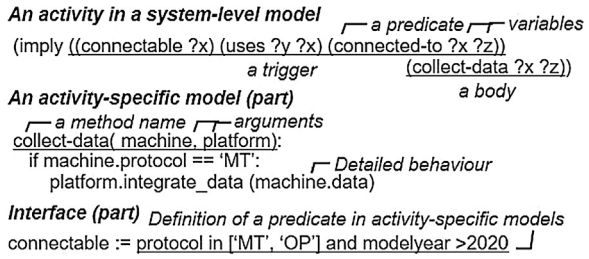

Fig. 3. Connecting a system-level model with an activity-specific model.

Second, while the system-level model specifies a set of activities and their execution conditions in terms of symbolic descriptions of the state of entities and their relations, an activity-specific model defines the detailed behaviour of entities in individual activities. Each activity-specific model is defined as a method, whose name is the name of the activity in the system-level model (e.g. collect-data in Fig. 3) and its arguments are slots of instances of entities (e.g. machine and platform in Fig. 3). The detailed behaviour of the instances of entities and their relations, which are shown as State$_n$ and Relation$_n$ in Fig. 2, is described as the contents of the method.

The interface defines the correspondences between the system-level and activity-specific models in terms of the state of entities and their relations (see the definition of *connectable* in Fig. 3).

2.3. Simulation mechanism formulation

The proposed framework uses a discrete event simulation mechanism in which the simulator generates instances of entities and simulates the detailed behaviour of entities as well as their relations, according to the definition of activity-specific models. Additionally, the present mechanism has two unique features. First, the simulator derives symbolic state of entities and their relations (State$_s$ and Relation$_s$) in the form of predicate logic statements from the corresponding numerical description (State$_n$ and Relation$_n$). Second, the derived statements and the definition of activities are used as the facts and rules of inference (reasoning), which results in new facts that indicate the activities to be executed, while the

Please cite this article in press as: Komoto H, Masui K. Model-based design and simulation of smart factory from usage and functional aspects. CIRP Annals - Manufacturing Technology (2018), https://doi.org/10.1016/j.cirp.2018.04.025