Methodology for the model driven development of service oriented plant controls

Jan-Philipp Schmidt*, Timo Müller, Michael Weyrich

*University of Stuttgart, Institute of Industrial Automation and Software Systems, Pfaffenwaldring 47, 70550 Stuttgart, Germany

Abstract

In the age of the Industrial Internet of Things, distributed subsystems are established in many domains, which bring added value through their networking. In the field of industrial automation this means a distribution of the control system. As the distribution involves an increase in complexity, new approaches are needed that address this problem. This paper presents a model-driven development approach that involves a Service Oriented Architecture.

1. Motivation

In contrast to centralized control systems, distributed controllers offer greater flexibility and reconfigurability [1]. The relevance for these properties is constantly increasing in the context of the Industrial Internet of Things (IIoT). The versatility and volume flexibility of an automated system makes it possible to operate increasingly volatile markets. Ever shorter production cycles, changing requirements for products and production plants, and individual production, which is a benefit for the end customer, make it increasingly difficult to economically use the automation technology applied to mass production [2].

In plant operation, a high degree of flexibility means that changes in plant control, which are necessary due to changed requirements, can be carried out with little effort. The reconfiguration refers to changes in the plant structure. New customer requirements, laws, standards or optimizations in the plant can be the cause for these changes.

There are already approaches that address these problems. The model-driven development goes further than the model-based development. In the design phase, models of the technical system are developed which are to be supported by the developers during implementation. In model-driven development, functional models are used to generate the code for the software system. This development process reduces the development effort both in the initial development of the plant control and in the change [3].

In addition, concepts for the coordination of communication are necessary. Due to their flexible structure, service-oriented architectures (SOA) are increasingly viewed as a communication paradigm between machines for future production automation [4]. The distributed structure of so-called SOA networks leads to decentralized coordination in automation systems [5]. This requires a higher degree of autonomy of the individual components, which integrate into the network ad-hoc and thereby interact with a large number of cooperation partners via communication interfaces. This, on the one hand, ensures the required flexibility of the control network, which can be flexibly adapted to changed framework conditions. On the other hand, compared to a central system with comparable control tasks, however, the complexity within the individual components increases, since comprehensive communication and coordination tasks are then given to the network nodes.

There are several approaches addressing the complexity of increase of automation. Design patterns support the developer...
with templates in the development and distribution of automation functions [6].

In this work, a shell model is presented, which supports developers in the modeling of new automation functions for a service-oriented plant control and thus reduces the effort for development and reconfiguration or modification of control systems in the field.

2. State of the Art

2.1 Distributed systems

Distributed systems have been in use for several years in the fields of multicore controllers, cloud and grid computing. In these areas, the availability of computational resources and the distribution of tasks for the computers, as well as concurrent problem solutions, are in the foreground. A distributed system is defined as "A collection of independent computers that appears to its users as a single coherent system." [7]

According to this definition, outsourcing of control into the field leads to a distributed system and thus to a new application field in the area of production automation. In this application field the properties of the distributed systems are used to achieve a higher flexibility and reconfigurability of the automated systems. For the user, the modular control of the system does not differ from a central automation system. Nevertheless, the controllers are independent in the sense that they do not have shared memories and therefore have to exchange information via communication channels.

Instead of a central controller, for example a PLC, which controls a chained process sequence as a central node, distributed control networks are used to distribute the control tasks to various independent control components, which lead to a subprocess. At the same time, control systems from different manufacturers, different domains and different requirements for real-time, reliability etc. will cooperate. Depending on the data volume, response time and real-time behavior, a large number of bus systems are available for communication [8].

Two types of complexity are distinguished. The complexity caused by heterogeneous systems, the compositional structure and the complexity caused by a decentralized communication between the control components, the communication architecture.

2.2 Service Oriented Architecture

As mentioned, a coordination concept for the distributed systems is necessary. In the area of Internet technologies, dynamically distributed IT systems have been structured over the paradigm of service-oriented architecture (SOA) for more than 15 years [9], which is increasingly being discussed in the field of industrial automation technology as the communication architecture of the future. For example, the VDI / VDE GMA status report of the industry-standard reference architecture (RAMI4.0) describes the industry 4.0-compliant communication of an industrial 4.0 component based on SOA - e.g. OPC UA [10]. The flexibility of the communication architecture results from the encapsulation of the functionality in a service, standardized communication interfaces and a loose coupling of the services. In this way, heterogeneous systems can be dynamically bound at runtime and orchestrated into higher services. The definition of a service in industrial automation as a “delimited range of functions offered by an entity or organization via interfaces” [11] differs from the definition used in Internet technologies, where services are described as the mechanism of their needs and abilities of software components [12].

This is usually followed by the Paradigm [13] server client, as in OPC UA. Servers provide services that the client can use.

There are already service-oriented protocols, such as OPC UA, which are used in automation technology. Since these protocols come from Internet technologies, they rely on the Ethernet protocol [14]. In the field of industrial automation, there are a wide variety of bus systems due to the different requirements regarding data throughput and real time.

2.3 Model Driven Development

The Model Driven Development decreases the engineering effort in the development of software systems. The increased complexity, which results from the use of distributed controls, primarily means an increased engineering effort. A reduction in the complexity within the components is not possible, since both the algorithms for the functionality of the controllers as well as the algorithms for the coordination of the controls have to be realized. What is quite possible, however, is a reduction of the engineering effort by methods of complexity control.

The automotive industry faced similar problems at the beginning of the 2000 years, which led to the consolidation of leading automotive manufacturers and suppliers to the AUTOSAR consortium in 2003, addressing the challenges of increasing complexity [15]. Software-technological innovations in the automobile were first implemented in new control units. Looking at the vehicle as an automated system, the control device network can be interpreted as a distributed automation system. The complexity meant that OEMs and suppliers agreed on the reference architecture AUTOSAR. This new established architecture serves the complexity management, which is now necessary in automation technology, by enabling model-driven development and virtualization by standardization of basic software modules and interfaces. Despite different requirements within the two domains automotive and plant automation the approach of the model-driven development can be transferred to the complexity control and thus reduce the engineering effort when using distributed systems.

The most important aspect for the transfer of the development methodology is a consideration of the AUTOSAR architecture. The layer architecture represents an abstraction of the microcontroller platform in several levels. The runtime environment on which the applications of the control units run have for hardware access and communication. Few standardized interfaces are available. This standardization makes it possible to automatically generate the code for the applications from functional models. Models available from the design phase describing the structure or behavior of a component, such as Petri nets and state machines, contain the information about the program flow of the application. Thus, the generation of code from these models for the application eliminates the manual implementation of the models in the implementation phase. Furthermore, the completeness of the
دریافت فوری
متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات