I. Troch, F. Breitenecker, eds.   ISBN 978-3-901608-35-3

On service-orientation as a new approach for automation environments

M. Schlütter[1], U. Epple[1], T. Edelmann[2]
[1]RWTH Aachen University, Germany, [2]Siemens AG, Germany

Corresponding author: Markus Schlütter
Chair of process control engineering
RWTH Aachen University – 52064 Aachen, Turmstraße 46 – Germany
Email: m.schluetter@plt.rwth-aachen.de

Abstract. In the field of automation environments software has to fulfil many standards especially according to security and safety. One way to make standardized functions applicable to engineers of automation systems was the usage of the IEC 61131. Using such a standard and an engineering tool, an engineer is able to build up an application for controlling a process within a plant. This results in two limitations. The first one are the fixed (engineered) connections between the parts of the application. So a change of the plant would result in a complete new program, which has to be compiled and loaded. The second limitation is that an information, which has been evolved at the level of automation systems, has to be forwarded via different interfaces to the higher management and enterprise levels, where it will be needed also.

This paper will show a service-oriented approach, to make applications of the automation environment in the whole enterprise-infrastructure available. The two mentioned limitations of the fixed connections and the accessibility of informations in just one special area of automation environments will become obsolete by using the service-oriented approach. So the plant, the engineering and at least the company will become more flexible.

1 Introduction

Enterprises have to become more flexible to present innovative products in even shorter time to the market. This is the most promising way for a company to get success.

This fact has more and more influence to the lower production level. The production is already highly automated to reduce the production costs, but the plants and explicitly the automation environment is not as flexible as it should be. So the costs to reconfigure a plant for producing new products are still high and it takes time to realize the changes.

In this paper we first present the architecture of automation environments as it is seen today. The next section shows the future demands on automation environments. Then the idea of service-orientation is presented. After that a solution to fulfil these demands using the approach of service-orientation is shown. The paper concludes with a summary of the given points.

2 Todays automation architecture

During the past in the field of automation the costs for a reconfiguration of a plant are reduced by using a high standardization. This was also important to get the needed reliability and safety of a plant. This lead to the *level model of a corporation* as introduced by [12] and further work like IEC 62264 [6]. Which results in the representation of the *automation pyramid* shown in figure 1.

On the top of figure 1 you see the level of Enterprise Resource Planning (ERP). This level is representing the enterprise management which decides what products will be produced generally, what plants are built up and how the human resources are distributed in the enterprise.

The next level is the Manufacturing Execution System (MES). The main tasks are the detailed planning of personnel, raw materials and the production schedule. Also key performance indicators (KPIs) could be determined in this level.

The role of the Control Level is to control and monitor the process. The actual state of the process will be determined by the actual sensor values and the result is the control value given to the actuating elements of the plant.

The Field Level contains the sensors and actuators. The sensors determine dedicated process values and the actuators manipulate the behaviour of the process.

The bottom level is the process itself. More or less it has interfaces to the higher levels just by the sensors and actors. The state of the process will be determined in the Control Level using a model of the process and the actual

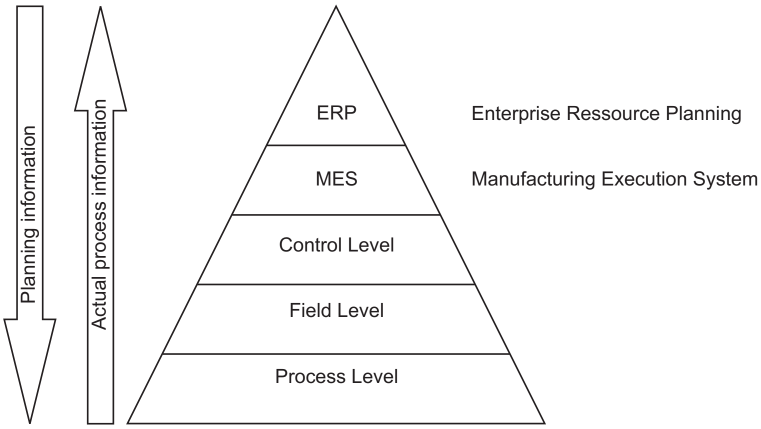

Figure 1: The automation pyramid with the main levels.

process values measured by the sensors.

These are the main levels and there could be more levels to get a more detailed description. Generally the information of the actual process state is given from bottom to top and the planning information is given from top to bottom. From bottom to top the informations are more and more concentrated so this explains the illustration as a pyramid.

For each level there have been developed effective tools to perform the desired tasks. Interfaces are defined to exchange the informations between two adjacent levels. So the tools implemented these interfaces and the information can be exchanged in both directions.

Due to the history and the different requirements on communication there have been established different communication channels in each level in the past. This was one of the most hindering point to get an easy information exchange.

In particular the last point has changed in the recent past. Inside the different levels more and more communication channels based on Ethernet and TCP/IP are established. So in theory it is possible for the chairman of an enterprise to get access to a dedicated process value. For safety reasons this is not intended.

There are two different sights to a plant and thus to the automation pyramid. One sight is the user as the operator of the plant. The other sight is the vendor of control systems and other tools to give the user the functionalities needed to run a plant. Both have sometimes different interests, e.g. the vendor wants to sell his own system and the user wants or needs interfaces between existing and new system elements.

Within the Control Level standards like the IEC 61131 [3] are developed to make it easy for an engineer building reliable and safe programs to control the process. Especially the idea of function blocks is based on the idea, that a small number of controllers is reliable for the execution of the tasks.

To become more flexible in this field the IEC 61499 [4] was founded. It defines function blocks in distributed Industrial-Process Measurement and Control Systems. So the production system can become more modular and thus the controllers can now distributed over the plant.

A further approach is Fieldbus Foundation (FF), which is based on further standards like the IEC 61804 [5]. It has in common to the idea of the IEC 61499 that the controlling part of a plant moves a bit more from the Control Level to the Field Level. So now it should be possible to build up more modular systems.

The basic idea of the last inventions is that an automation system is a distributed and decentralized system. But nevertheless in all ideas it is not possible to make changes to a running plant. So if you just want to add a new function to the process you have to stop the process, turn off the controller, load the new program and start the process again. Often this is from the sight of the process not necessary and so the user did not want this due to the upcoming costs.

3 Future demands on automation environments

One of the motivations behind the data exchange language AutomationML™for production planning systems [2] is that engineering takes up to 55% of the costs of automotive manufacturers for investing in their automation environment for new plant elements. Another motivation is to modularize the manufacturing systems to get a higher flexibility. So a plant consists of components using defined interfaces. These interfaces are derived from the interfaces in the real world like the material transport interface, the communication interface of the fieldbus and others. The intention is to get the configuration of a plant consisting of such components more easily.

So components fulfil a functionality within a plant and they may have each an own controller to monitor their functionality. But a component needs access to all levels of the automation pyramid, because it has influence on e.g. the KPIs derived in the ERP or MES levels.

Another point is that components will be created by other contractors than the vendor of the field instrumentation or the vendor of a control system. So the contractor produces a high-performance component and did not want that his know-how goes to the user of the component. For the user the transfer in the other direction is also an important point. So knowledge has to be hidden in front of both parties.

In automation technologies client-server architectures are used for reliability and safety reasons, but in the future it is desired to organize them distributed and flexible. This could not be reached by the common used techniques.

The goal is to reduce the time consumption and thus the costs for engineering by using flexible, easy re-configurable and agile production systems. This should be achieved by giving an add-on to existing systems.

4 Approach of service-orientation

First time Gartner [10][1] introduced the term 'service oriented architecture'. The wording *architecture* shows already, that it is a concept and it is independent from a special implementation.

Today service-oriented is often associated with Web services. Also Gartner does so nowadays. That is not the idea presented here. Here the understanding is like it is presented by OASIS [11] as a paradigm for organizing and utilizing distributed interoperable systems.

The original idea for service-orientation is the view on business processes within a company. A company is a system and each business process is a service which may also be composed by other services. So more generally a system can be divided into different functionalities. Each functionality is represented by a service. Such a service has a description and if used, it promises some requirements to the user of the service.

The platform for realizing a service is negligible and each service can be realized in a different way by each provider. The behaviour of a service is defined by its functionality.

In order to find a needed service it must be published to possible service requesters. Therefore it is necessary to have a common infrastructure for communication. So one functionality is the search function for demanded services. A service could be offered by more than one provider. In this case one provider able to fulfil the requirements of the requester has to be chosen.

A service can be accessed via exact defined interfaces. The interface description has to be published and the details of the implementation are not visible to others. By this services are loose-coupled. If a service needs another service to provide its functionality, the requester will not see the dependencies and thus has not to handle the internal states. The service provider just returns an answer to the service requester. By this the communication is asynchronous in service-oriented systems. So the requester executes a service and can continue with the execution of further operations.

A service has to be described. The description contains in addition to the interface some capabilities and requirements. So a service can be evaluated and the requester will be bound to the best provider. That is for example a requester who needs a service frequently and thus he will not bind to a service provider executing the service just once. Here reliability and security criteria are also involved.

Another survey on service-orientation can be found in [9]. It focuses on the future internet architecture, but the related work and the service-oriented approach are clearly represented.

One way to implement the management capabilities is the usage of Web services as introduced in [7]. Due to the overhead of such systems by using WSDL and other XML-oriented techniques the usage of Peer-to-Peer systems as presented in [13] may also a good approach. Peer-to-Peer systems are developed for distributed systems with the following requirements:

- Scalability
- Security and reliability
- Flexibility and Quality of Service

These requirements meet the requirements of the service-oriented architecture. A service provider can be added at runtime to scale one service. Also a service itself can be added at runtime. If a service provider fails, another one will be able to do the task. And due to changes in the capability of a service of one service provider it will be also possible to get another adequate service provider.

[1]The original paper is not free available. The cited paper from one of the authors also shows this statement.

5 Service-orientation in automation environments

The idea to use service-orientation in automation environments emerged recently. Mendes presented in [8] already a full functional architecture to design such systems. The main disadvantage of this approach is, that it does not consider the existing architecture of automation systems. As pronounced in section 3 this paper has the goal to enhance the existing architecture. By this the standardized well-known functions can be used further more. The aim is to reduce the engineering costs by offering basic functionalities of the automation environment and even other levels of the automation pyramid in a service-oriented manner.

The model of the architecture used here was introduced by Epple on the NAMUR Workshop 2008 [1]. It is shown in figure 2. On the left side the classic automation pyramid is placed. The right side shows the new service system providing functionalities which can be used by tools of the different levels of the automation pyramid. Between both elements the service interfaces are placed to make the access to the services possible.

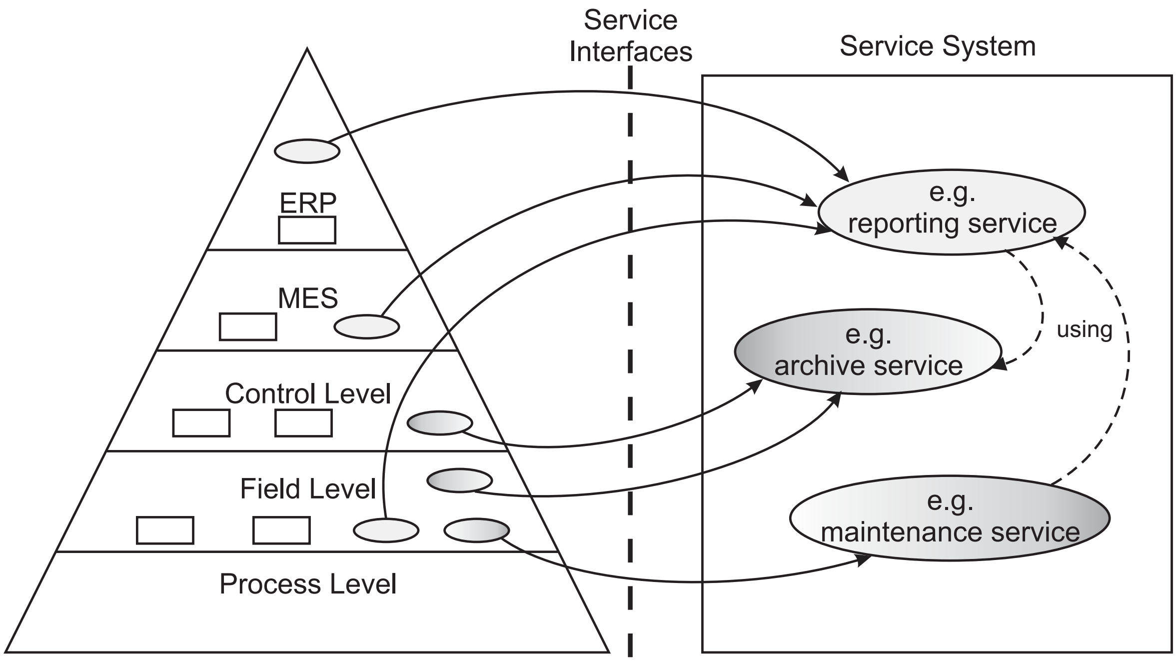

Figure 2: The automation pyramid with the service interfaces and service system [1].

In each level of the automation pyramid there are two kinds of functionalities. The first one (represented as a box) is not realizable or still not realized as a service and hence it still remains in its previous environment. The second kind (represented as an ellipse) is a functionality which is realized as a service. So first by the usage of the service interface the service has to be located in the service system. The next step is binding to the service provider complying the requirements. The last step is the usage of the service.

The service system provides the resources to manage and to execute the services. Within the service system it is allowed to the services to make usage of other services, e.g. a maintenance service may use a reporting service signalling a maintenance message. By using a service its requirements may change. So the service system is also responsible to change existing bindings between service consumer and provider.

The service interface describes how a concrete service with given requirements from the consumer will be located. Furthermore the binding is given. To execute a service the needed parameters are described in the service interface as well.

Due to the already mentioned fact that the communication in each level will be based on Ethernet and TCP/IP, an unique communication infrastructure between all levels will be feasible at least in the near future. So a service system is available on all levels of the pyramid and one service is not responsible for just one level only. The service system is an effort of the whole automation system.

A survey on possible classes of services and more concrete services follows here. First of all there are classic control services like archiving or reporting. Next theme are steering services e.g. for tuning of process parameters. Also monitoring is one possible service class to determine KPIs of a plant or an enterprise. Another often used functionality is maintenance e.g. to realize an asset management service. Other classes will follow and have to be discussed in official committees where users and vendors are both represented in order to get a widespread acceptance.

To avoid a misunderstanding by the interpretation of figure 2 it is said explicitly that the arrows in the drawing just indicate the binding to a service. The information may flow in both directions. E.g. it would be less useful to define an archive service only saving informations without the possibility to request informations.

Open tasks for the users are to describe the functionalities transfered into services and to define the service interface with its requirements. The vendors open tasks are to build up concepts for such a service system and to implement the services described by the users. Both vendors and users have the common task to define the services and the usage of the interfaces of the services.

Another perspective is to regard a plant as a system composed of components. A component brings along its own controller and may be ready engineered for the usage in the plant. Now the service system provides all functionalities as services needed to connect to other components of the plant. Especially classic control services can be realized by this way and reduce the engineering effort tremendously.

This idea is shown in figure 3. First a component will be added to the existing and still running plant components. In the next step it searches via the service interface a requested service with some requirements. The service system returns a concrete service provider fulfilling the requirements and the component will be bound to this provider. At least the service could be used by the component.

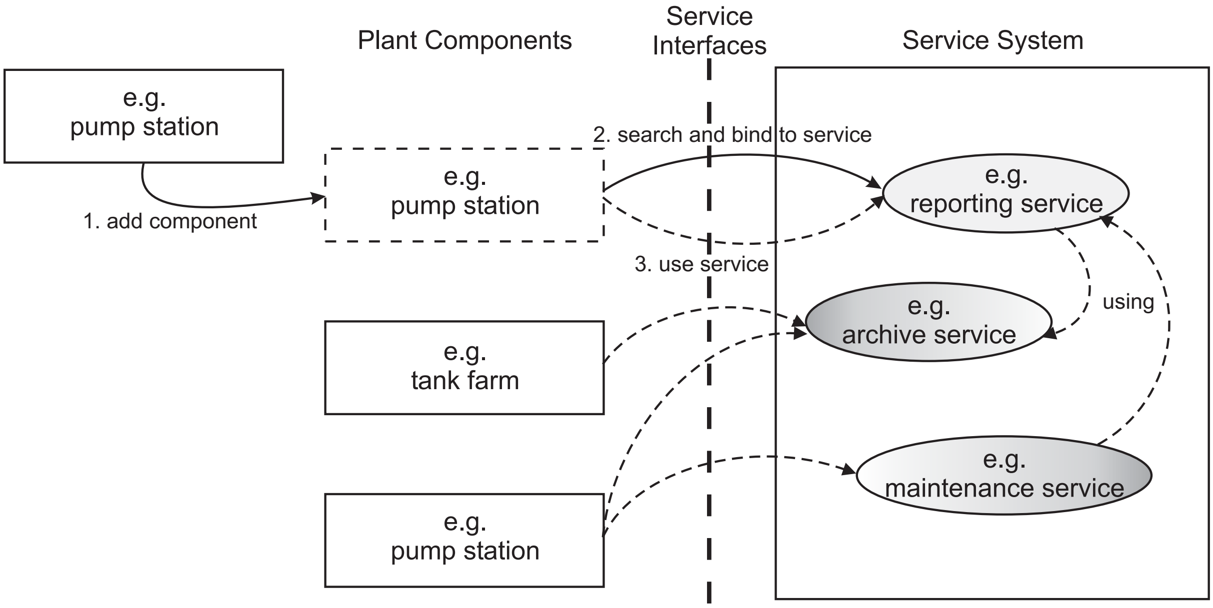

Figure 3: Components using the service system.

This shows again that the service system is a separate system provided by the user as an operator of a plant. The benefit is that the engineering effort and thus the costs will be essentially reduced by this approach.

Now it becomes visible that two goals are reached. On the one side a plant can become modular by components and changes generate less effort. The other side is that a service system gives the potential to add functionalities to a system during runtime without any loss in production time.

The introduced design of the service-orientation for automation environments also allows a parallel existence of former installed plant elements and the usage of components using the service system.

Another advantage of this approach is the easy chance to realize functionalities which make use of more than one level in the automation pyramid. E.g. to get a maintenance information from the field level to the MES level, the control level has also to provide the interfaces and the processing of this information. This is the procedure today. With the usage of a service system the control level will be no longer involved in the information processing.

The benefit by using a service system is that it will become possible to add new functionalities to a system. Also the enhancement of existing services is feasible (scalability). The realization as an add-on protects the investment on existing plant elements. The description of the tasks of user and vendor are more easy and can be clearly separated from each other. The information exchange between persons responsible for different levels become more easy.

At last the approach assures a reliable, flexible and agile system, which is easy to configure, maintain and reconfigure. Thus it leads to a reduction of engineering costs.

6 Conclusions

This paper has first shown the history of todays automation architecture. Mainly the automation pyramid of a corporation with its different levels and the responsibility of each level was introduced. It was shown that this architecture was established for reliability and security reasons. This results in high engineering costs of a plant.

The demands on future automation environments are to reduce the engineering costs. This should be achieved by an add-on to the existing architecture considering the existing architecture. So this guarantees an investment protection on existing plants.

After that the approach of service-orientation is presented. Main point is the offering of functionalities as scalable, flexible and reliable services in the field of automation.

At last the paper shows how this requirements can be fulfilled by a service-oriented approach. The approach is based on the idea to enhance the automation pyramid with a service system. Each reasonable functionality will be realized as a service. One assumption is that a plant consists of different components each reliable for one step in the production. So each component can be engineered independent from other components. The standardized functionalities of the components need not to be engineered any more due to the fact, that these functionalities are applicable by the service system. This results in more flexible systems with reduced engineering time and costs.

7 References

[1] Epple, U.: *Strukturen in der Leittechnik - mehr als das Ebenenmodell*, NAMUR, Presented at the NAMUR Workshop, 2008.

[2] Garcia, A. A. and Drath, R.: *Whitepaper <AutomationML™/>*, `www.automationml.org`, Version 5, Nov. 20, 2007.

[3] International Electrotechnical Commission: *IEC 61131-03: Programmable controllers – Part 3: Programming languages*, 2003.

[4] International Electrotechnical Commission: *IEC 61499-01: Function Blocks – Part 1: Architecture*, 2005.

[5] International Electrotechnical Commission: *IEC 61804-01: Function Blocks for Process Control – Part 1: Overview of system aspects*, 2003.

[6] International Electrotechnical Commission: *IEC 62264-01: Enterprise-control system integration – Part 1: Models and terminology*, 2003.

[7] Leitão, P., Mendes, J. M. and Colombo, A. W.: *Decision Support System in a Service-oriented Control Architecture for Industrial Automation*, In: Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2008, pages 1228 - 1235, Sept., 2008.

[8] Mendes, J. M., Leitão, P., Colombo, A. W. and Restivo, F.: *Service-oriented Control Architecture for Reconfigurable Production Systems*, In: Proceedings of the 6th IEEE International Conference on Industrial Informatics, INDIN 2008, pages 744 - 749, July, 2008.

[9] Mueller, P. and Reuther, B.: *Future Internet Architecture - A Service Oriented Approach*, In: it - Information Technology, Vol. 50 (2008) 6, pages 383 - 389, 2008.

[10] Natis, Y. V.: *Service-Oriented Architecture Scenario*, Gartner, ID Number: AV-19-6751, April 16, 2003.

[11] OASIS: *OASIS Reference Model for Service Oriented Architecture 1.0*, Official OASIS Standard, Oct. 12, 2006.

[12] Polke, M.: *Process Control Engineering*. Chemie Verlag, Weinheim, 1994.

[13] Steinmetz, R. and Wehrle, K. (Eds.): *Peer-to-Peer Systems and Applications*. In: LNCS, Vol. 3485, Springer, 2005.