Autonomous Production Workstation Operation, Reconfiguration and Synchronization

Jaime Garcia Represa, Jerker Delsing
Luleå University of Technology, Luleå 97187, Sweden

Abstract

The decoupling of production line’s workstations and its equipment from the higher layers in an automation architecture has the potential to provide the following benefits: 1. Dynamic workstation reconfiguration; 2. Autonomous synchronization of workstation equipment; 3. Autonomous workstation operation.

Each of these benefits will improve the automation flexibility at each workstation and for the whole production line.

Introducing IoT equipment into workstations and making use of Service Oriented Architecture (SOA) fundamentals such as look-up, late binding and loose coupling are shown here to provide the above benefits. In addition, the cost of such an implementation and deployment seems to be significantly less than the comparable cost for legacy technology.

An implementation, using the SOA approach on a workstation, at the Volvo Trucks production facility in Gothenburg was made possible thanks to the Arrowhead framework, which has been shown to provide all three listed improvements.

The analysis of the above-mentioned demonstration clearly shows how the workstation reconfiguration is made simple using the arrowhead ServiceRegistry system. Autonomous synchronization is achieved through the look-up and late binding capabilities provided by the arrowhead Orchestration system in cooperation with the ServiceRegistry system. Finally, autonomous workstation operation is provided through the support of the arrowhead Workflow Manager and Workflow Executor systems.

All these benefits were achieved at a significantly reduced cost compared to comparable legacy implementation.
1. Introduction

Competition is the driving force behind industry development. Customers are pushing for more personalized products, with high quality and low delivery times. This is a struggle for industries following the ISA-95 [14] standard for factory organization, so new models have arisen promising factories with increased flexibility, higher productivity and lower cost.

To comply with these new requirements, organizations have proposed different architectures to drive the change, such as the Industrial IoT [8] or the Reference Architecture Model Industry 4.0 (RAMI 4.0) and the I4.0 Component [16]. However, there is still no clear implementation of these architectures.

Manufacturing systems need to change and adapt to these new architectures. For this purpose there are emerging technologies that will enhance the automation systems and fill the gaps [17]. Some of the key technologies that are currently in the spotlight are: Cyber-Physical Systems (CPS), Internet of Things (IoT), cloud computing, big data analytics, and information technology, with focus on Service Oriented Architecture (SOA). These technologies do not come from the automation field, so there is a need for integration such that they meet the requirements of the area. Several roadmaps for this have been published by, e.g., EFFRA, ECSEL and ProcessIT.EU [5, 4, 9].

One of the promises that these new technologies try to fulfill is the concept of a smart factory. In the past, the Enterprise Resource Planning (ERP) and the Manufacturing Execution Systems (MES) were introduced into factories to increase their productivity and establish a clear task organization. Currently, industry is looking for vertical integration and broad communications, having all the components of the factory connected. This path will have to pull down the barriers imposed by the ERP and MES systems while providing the same functionalities. A smart factory needs to be context-aware, and have a system that operates with information from the physical and the virtual world [10].

Smart factories can be divided into components, machines and production systems according to [7]. Working between the machines and production system levels, the production lines are often grouped in workstations where a set of operations are performed with different machines by manual workers. For a smooth transition from ISA-95 to smart factories, autonomous production workstations will be a desirable evolution. In addition to smart products, these workstations will provide easier and faster adaptation of shop floors, enabling plug & play tools, and more reliability, as the workstations become autonomous with the ability to operate and execute even if they are disconnected from the higher level control system, such as an MES, PLM or ERP.

An autonomous workstation has, as the main difference to today’s workstation, the ability to make decisions based on information from the physical and virtual worlds. The properties necessary for these autonomous workstations are dynamic reconfiguration, autonomous synchronization of the equipment, and autonomous operation at least. Smart products will have the computational power to be actors in the new architectures. They shall carry their own production orders, which are currently created by the ERP systems and stored in the factory network. The smart products will be responsible for the operations performed by the autonomous workstations, as they will store and communicate the information to command the operations. In this way, many tasks currently provided by ERP and MES systems can begin to be distributed in the new paradigms.

Smart factories are currently in development. To clarify the concept, research centers are working on their vision of a smart factory, as in [10], together with initiatives such as Factories of the Future [5]. Different theoretical solutions have been proposed to create flexible manufacturing applications, but an integrated solution for network decoupling, reconfiguration and synchronization of production workstations has not yet been found. Clear and concise requirements for autonomous production workstations are still to be agreed upon.

In this paper we proposed a technical and practical solution to integrate autonomous workstation into factories using the Arrowhead framework, which uses the System of Systems (SoS) and SOA to create an autonomous workstation. The presented solution has been tested in an industrial use case, providing interesting characteristics that can be seen as fundamental for autonomous workstations, though their requirements are not yet precisely defined:

- Dynamic workstation reconfiguration
- Autonomous synchronization of workstation equipment
- Autonomous workstation operation

The proposal has been tested in a use case at the Volvo Truck factory in Gothenburg. An autonomous workstation has been implemented with a legacy tool, that is used in the regular process of assembly, an adapter and a hardware
device that expose the operations of the tool as services. This setup has proven the functionalities that may be achieved when introducing the Arrowhead framework in a workstation.

The remainder of this paper is structured as follows: Chapter 2 defines the concepts which enable autonomous workstations, compares these concepts to other research in the field, and explains the basic ideas of the framework used. Chapter 3 relates the SOA fundamentals, as enablers of the concepts used to create an autonomous workstation, to the systems of the Arrowhead framework implementation. Chapter 4 outlines the use case and its architecture, and explains the benefits achieved at the factory with this setup. Chapter 5 contains the contributions of this paper with a summary of the work performed and proposes future enhancements for autonomous workstations.

2. Related Work and Background

It is important to clearly define the concepts of interest in this work, as such the benefits it will provide as a closer approach to the smart factory concept, in the context of Industry 4.0. The goal of having an autonomous workstation cannot be achieved without a series of requirements to be met.

In the Far-Edge project [6], an analysis of the challenges for the adoption of decentralized automation systems shows that implementation of IoT/CPS is not standard in industry, and that distributing the tasks assigned in the ISA-95 architecture to the new components of smart factories still needs further study. The proposed approach is based on key concepts, such as self-contained local clouds, SOA, service discovery, orchestration and autonomous execution. These are implemented using the Arrowhead framework [1], which has been tested in a number of use cases. The aim of the use cases can be summarized by decoupling the ISA-95 architecture, and showing the benefits of including business logic at lower levels with edge computing. Assigned to each use case are concrete properties that together should achieve the high-level goals. These properties match the benefits claimed in the abstract, and are further detailed in the next paragraphs.

- Dynamic workstation reconfiguration refers to the process of dynamically loading a new configuration when a new tool or machine is added to the workstation, or when the operations that have been performed have to be readjusted due to product requirement.
- Autonomous synchronization of workstation equipment is the process in which the physical functionalities of the equipment are matched through virtualization, and a digital twin is achieved with synchronization capabilities.
- Autonomous workstation operation consists of operation independent of a connection to an upper layer of the factory network, in an ISA-95 architecture this would be the MES layer in charge tracking, execution, planning, etc.

Together, these benefits will solve many of the shortcomings shown by the state of the art solutions for flexible automation systems.

In the literature, different strategies have been pursued to achieve results for flexible manufacturing that could populate smart factories [15]. One of these strategies is based on autonomous industrial mobile manipulator (AIMM). In [11], the introduction of autonomous mobile manipulators has been tested in an industrial environment. The study has shown that for complex operations in an assembly line this type of robot is well suited, offering more flexibility than fixed ones. At the moment, the technology is not ready yet for mass implementation because of errors in communication with the central network. It identifies the tasks of workstation reconfiguration and integration as a big drawback for commercial success. For this case, the autonomous workstation operation and the dynamic workstation reconfiguration can be seen as desired properties.

When looking at how problems with similar requirements have been solved in the data domain, the use of SOA is broadly accepted [12], and the property of loose coupling has been identified as a key ingredient in [3].

To enable autonomous workstations and their benefits, the Arrowhead framework has been used [1]. This framework uses the principles of SOA with the objective to enhance interoperability between IoT, and build SoS combining these IoT components. The property of loose coupling is part of the Arrowhead framework set of contributions as will be explained in section 3.
To accomplish interoperability in the Arrowhead framework, IoT components are abstracted into devices, systems and services. In this context, the hardware layer of the IoT will be referred to as devices, the software applications will be systems, and the exchanges of information will be the services. Not all IoT hardware is an Arrowhead device; for it to be compliant with Arrowhead it must be able to store, load and execute an Arrowhead system. An Arrowhead System is a software that is capable of providing and/or consuming services. It must also interact with the Arrowhead core systems, to publish the services it provides and allow the Orchestration system to offer the services it produces. Finally, a service is a software component used to exchange information between systems, consisting of a service interface and an implementation. It is at the service exchange level where the information assurance is accomplished.

Inside the Arrowhead framework, the IoT components are grouped together in local clouds to form SoS. A local cloud is defined as a protected network that provides communication and computation environment suitable for automation, meeting the key automation requirements of real-time, security, interoperability, scalability and engineering [2]. The local cloud concept is based on the assumption that automation is physically and geographically local, so it matches well to the characteristics of a production workstation. Following this idea, the future structure of intelligent factories will be composed of local clouds, thus mapping the actual distribution of a workstations. When a new machine is introduced to a workstation, in Arrowhead it will be understood as a new device with new systems and services.

Arrowhead local clouds must include the three mandatory core systems: ServiceRegistry, Orchestration, and Authorization systems. They serve as a support for the creation of SoS between application systems and provide fundamental properties to a local cloud. It is inside the local clouds that application systems tailored to perform the SoS goals are deployed. There is another set of core systems that can be included in a local cloud when their functionality is required; they provide support for specific use cases and are being expanded continuously.

Briefly, the Arrowhead framework provides an answer for the question that arises in IoT. When everything is connected and communication between devices is easily available, a thing needs to know who can provide the information that it requires to fulfill its goal and how is the information being sent. By means of the mandatory core systems, the Arrowhead framework can provide an answer. The response will come from the Orchestration system, that will check the available services and its properties in the ServiceRegistry and, then, verify if they are allow to communicate between them according to the Authorization system rules.

3. SOA fundamentals and implementation

There are a set of properties in the Service Oriented Architecture paradigm that are the key to the autonomous workstation presented in this work. Our focus is on the look-up, late binding and loose coupling properties of the services. These properties are available in the Arrowhead framework implementation by means of itself being service oriented or by the presence of the mandatory core systems.

Look-up, also called dynamic discovery, refers to the functionality of a system to publish its services in a public register, when it is present in the environment, and to discover other services that could be compatible with its application. This can be done at design time, looking at the documentation of the services and its contract, or at run-time, if the system can access the current data of the services deployed. This enhances the independence of the services with respect to the predeployed infrastructure and requires less prior information for the systems to work.

In the Arrowhead framework, look-up is achieved by means of the Service Registry, a system that provides a database that includes the currently available services in that local cloud/workstation. It provides the core service, “ServiceDiscovery”, that the application systems can use to register a new service, modify the entry or submit a query for a service. The service information that can be accessed includes the endpoint of the system that provides the service, which lowers dependencies when creating the systems, and its metadata if available, which may help to query for a compatible service. This opens the possibility for dynamic discovery of services at run-time.

Late binding, or dynamic binding, is the process of aggregating services at run-time, as opposed to early binding. In this way, service consumers can choose from the available services that can bind at any time, considering the optimal option at the moment, so that they can access the information when they need it. Another way late binding can be achieved is by introducing mediation in the process, to release decision making from the consumers. This property can also enable the self-healing of applications, as consumers can find replacement services that meet their requirements if their provider becomes unreliable or stops offering the service.
The core system Orchestration, in Arrowhead is the one in charge of the mediation process to provide late binding. It offers the "Orchestration Service", by means of which it receives requests from service consumers with the requirements of providers, and it responds with a list of suitable services that match the needs of the particular consumer, including the information on how and where to connect. The Orchestration introduces safety and security mechanisms in the process by calling the Authorization system. Only in some cases can there be late binding without the orchestration process in Arrowhead, when security, safety, quality of service, inter-cloud connections or accountability are not concerns.

The last property is loose coupling. Coupling is understood as the relationship between service consumers and providers in terms of the number of dependencies they share [13]. In a tightly coupled system, the number of dependencies is high and therefore flexibility decreases, changes in one system affect the other. With SOA, designers pursue loose coupling, systems that share few dependencies making them independent of one another. In this way, systems can have different life cycles, and reuse of services becomes easier.

This property is not enforced by any specific system; instead, it is dependent on the implementation of every system. It is achieved in Arrowhead by following the SOA principles, guidelines and best practices when designing systems and services. In its definition, the Arrowhead framework has as objective to keep the advantage of loose coupling by following the SOA paradigm [1].

Any framework or architecture following these properties, should be a valid candidate to be used in a scenario where workstations need to adapt to shop floor changes. They support the reconfiguration of workstations and the synchronization of new equipment.

4. Autonomous Workstation implementation

A workstation in a factory is commonly composed of a number of machines and operators. We proposed to transform it into an Arrowhead local cloud, inside of which are included different application systems corresponding to the machines present physically, with the mandatory core systems. In addition, to adapt the general concept of local cloud for a production workflow and increased the amount of logic at the edge level, the Workflow Manager and Workflow Executor core systems were added to the workstation cloud. These core systems add support for the functionalities of tracking and execution, which are commonly provided by the MES.

In the use case, the machine used was a common assembly tool with no previous service-oriented design. Instead of modifying the legacy software, an adapter was connected to the machine. The local cloud was deployed in an embedded platform, and the different devices were connected, including the adapter and display, through a wired connection, Ethernet, to not disturb the production networks.
To obtain a holistic view of the whole manufacturing workflow in a factory, the smart product and the MES need to be included in the setup, although it is different from their normal implementation due to the need to have their functionality configured as services. In the case of the MES, this adaptation is a step toward smart factories that has been researched in industry. On the other hand, smart products are more commonly seen as passive elements, carriers of information through the use of RFID tags, for example. As technology advances, the inclusion of more computational power and logic at the product level is foreseen, and the expectation is that they will take an active role in the manufacturing workflow. The final architecture is shown in Fig. 2.

The production workflow will start with the exchange of information between the ERP/PLM system, in charge of creating the production order, and the smart product, which will carry this information through the factory. This part of the workflow has not yet been tested with a real ERP/PLM system; instead, the production order has been taken and introduced in the local cloud offline prior to the creation of the smart product. Once the production order is transferred, there is no need to connect again to the factory network for the execution of the operations on the product. However, before the product leaves the factory, it will try to connect so that information about executed operations and their details for each workstation can be stored and analyzed by the factory network. Additionally, for back-up purposes, the smart product could establish connections whenever the production order changes, as is done for version control.

As the product moves along the assembly line and reaches a workstation, communication is established between the product and the workstation. In this use case, a manual scanner and a bar code was used to start the communication process. In the future, the product will be an IoT device on its own able to start this process automatically.
The production order is then transferred to the workstation. A Workflow Manager system at the workstation transforms the production order to a state machine. This state machine is consumed by a Workflow Executor system, that will command the machine and/or operator task to be executed by the workstation.

The results of the operations are returned to the Workflow Manager. It modifies the production order accordingly, and in case of an error, the smart product will decide if it must go to a repair station or continue in the assembly line. This decision is hard coded for this use case, although in the future, mitigation actions could be introduced to the product depending on the error detected.

In this workflow, a connection to the factory network is necessary at two points: when the smart product is online for the first time, before the assembly process starts, and at the end, to store the production order and the operations results when the assembly process is finished. The decoupling of the production workstations is achieved thanks to the smart product and the autonomous workstation, as their only dependency is the production order used by the Workflow Manager. This clearly shows that autonomous workstation operation is possible with this workflow, and demonstrates the way loose coupling is used when designing the interaction between systems.

![Fig. 3. Communications to the factory network](image)

The process of reconfiguration is made simpler in autonomous workstations. When a new machine is added to a workstation, it will register its services in the Service Registry, which provides the real-time data of the machines available using the look-up property. With this information, a new configuration can be stored in this local cloud or in any local cloud with a connection to this one, and can be downloaded into the machine upon setup. The only effort needed when incorporating a new machine into a local cloud is to go through the secure bootstrapping procedure established by the Arrowhead framework, although this procedure can be performed once when deploying the local cloud. The Configuration core system can provide additional functionality, as a place to store the configurations that can later be uploaded to the application systems with a push-pull communication protocol. The Configuration system can be deployed in any Arrowhead compliant local cloud, where it can provide its support services.

Finally, the autonomous synchronization of the workstation with the production order is achieved mainly with the Orchestration system. Looking at the situation from the point of view of the Workflow Executor, which receives a state machine with the order of the operations to be performed, and needs to find application systems able to fulfill the job. It will send a request to the Orchestration asking for a list of available services that offer the operations it needs. The Orchestration in turn will ask the ServiceRegistry for the services available at the moment, so that it can match those services to the needs of the Workflow Executor and answer the request with the endpoints of the service providers. These interactions are made possible due to the late binding property of the framework, as the machines can be bound at run-time to the system in charge of executing the workflow.
The above solution has been implemented at a pilot production line at the Volvo Tuve factory, Gothenburg Sweden. The implementation was made for a wheel alignment workstation with a connected wheel alignment measurement system and connected nut runners for adjustments. To create the local clouds needed in the setup, it was necessary to include additional hardware to adapt the wheel alignment equipment and product to the Arrowhead framework. At the end of the demonstration, the result of the measurements was stored in the smart product, which in this use case is a truck, showing the whole cycle complete for a production workstation.

The solution showed improvements compared to the current setup at the factory. The process of updating a workstation can nowadays take up several months, and it requires the support of the operations technology (OT) department together with the information technology (IT) department. In our solution, a workstation can be modified in a few weeks and the work of the IT is greatly reduced, although some training will be required in order to change the established procedures. Other benefits can be observed at the information management level, as the ServiceRegistry of the Arrowhead framework can help tracking the tools connected to each workstation, keeping an up-to-date version of the factory layout.

The operation of the workstation on real production trucks has been reviewed by the production staff and executives. Their feedback very clearly indicates that several of the current production system shortcomings can be solved by the experimental solution. Thus, showing the feasibility of the proposed approach.

5. Conclusion

The use of new technologies such as CPS and IoT is accelerating the path to smart factories, and the traditional structure of shop floors has to change accordingly. Many assembly lines are organized as workstations that are specialized in a particular set of operations, and when the recipe of the product changes do to customization they are hard to rearrange and suffer from rigidity.

This paper proposed a solution for implementing autonomous production workstations, exploiting the benefits of decoupling the production workstations from the factory wide networks and promoting the addition of business logic at the edge level through the use of smart products. In an autonomous workstation, tools can be added and configured as plug & play, thereby reducing engineering effort and time. In addition, networks problems will not cause a stop of the assembly line, as the production order information contains all the necessary information for the autonomous workstations to operate.

The solution has been tested on an assembly line in an industrial environment as part of an use case. The results were promising as a proof of concept, with the necessary adaptations to a real shop floor. The systems developed will be further polished to be included in the official repository for other projects to use.

The SOA fundamentals and the Arrowhead implementation used are not fully exploited yet. If they have the potential to create digital twins, as important tools to manage and optimize production, they will be further investigated. To smooth the path of adoption of these new strategies, a better integration with the ERP/PLM systems or IoT platforms is in the scope of future work.

Acknowledgements

The authors would like to thank the Horizon 2020 projects FAR-EDGE (grant agreement 723094) and the Productive 4.0 project (grant agreement 737459) for funding and sharing thoughts through discussion and debate.

References