Workflow Management for Edge Driven Manufacturing Systems

Hasan Derhamy*, Mattias Andersson†, Jens Eliasson* and Jerker Delsing*
*Dept of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå , Sweden,
Email:hasan.derhamy@ltu.se
† Volvo Trucks Corporation, Gothenberg, Sweden, Email:mattias.a.andersson@volvo.com

***Abstract*—The fourth industrial revolution is one of digitization. As manufacturers build more digital equipment into their process harnessing the potential becomes a challenge. ISA 95 based manufacturers rely on top down decision making, with redundant flow of information in order to organize Workflows on the factory floor. With advances in edge computing and smart objects, more decision are made at lower levels of the infrastructure hierarchy.**

However, how can the MES tracking and execution functions be decentralized while maintain an organized collaboration of smart objects? This paper proposes a Cyber Physical Service Oriented System of Systems operating on edge computing. There are three systems proposed here; Smart Product, Workflow Manager and Workflow Executor. These systems operate within Arrowhead Local Clouds and create a fully autonomous set of system to complete a production order. Planning, Supply chain and Quality of Service monitoring are out of scope and remain as centralized activities. The Smart Product is the core Cyber Physical System that must be context-aware in order to pass the correct Workflow information to the Workflow System responsible for current activities.

***Index Terms*—Edge Computing, Industry 4.0, Internet of Things, Arrowhead Framework**

I. Introduction

The fourth industrial revolution is distinct from the third industrial revolution in: velocity, scope, and systems impact [1]. This is because the fourth industrial revolution involves the fusion of technology from physical, digital and biological domains. This means new products will be developed and manufactured and traditional products will be manufactured in new ways. Two key challenges facing traditional manufacturers in the fourth industrial revolution involve scale and cost of digitization. Issues of scale are not limited to just the volume of devices or inputs and outputs. But also the scale of complexity is also increasing. Flexibility and adaptability are some of the key drivers of this complexity. Applying traditional centralized approaches suffers from the requirement for availability and security. As the dynamics on the factory floor increases the infrastructure demands increase, thereby increasing costs of backup systems.

New approaches such as the Reference Architecture Model for Industry 4.0 (RAMI 4.0) decentralize responsibility to individual components [2]. Figure I shows the RAMI 4.0 diagram. It shows that some enterprise logic exists even at the device level. However, how much logic should be included into decentralized systems? How to coordinate all the logic?

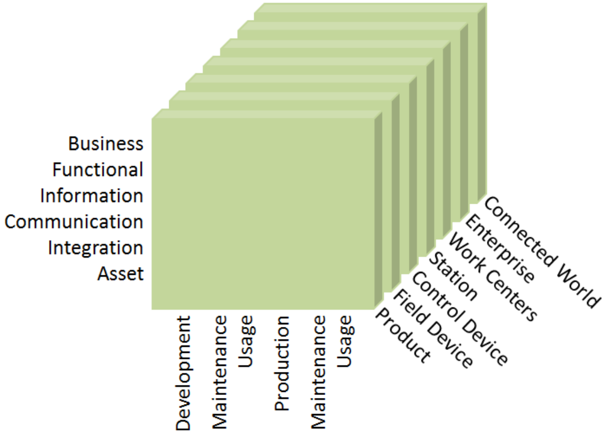

Fig. 1. The RAMI 4.0 layers vs hierarchy vs life cycle diagram

Edge computing paradigm has been suggested to help alleviate the backhaul traffic of large volumes of sensory traffic. However, edge computing when applied to a manufacturing scenario can help to reduce the decision making path. If RAMI 4.0 is applied to edge computing then a pattern for answering the question of coordinating and decentralizing logic can be made. The Far-Edge project is developing a platform that enables RAMI 4.0 based application development. The platform is building on the Arrowhead Framework, utilizing Service Oriented Architecture (SOA) thinking to divide responsibilities.

This paper proposes edge automation services that coordinate and enable productive collaboration between Cyber Physical Systems (CPS). The edge automation services consist of functionality required to plan, sequence and interconnect CPS systems. It uses the Arrowhead Framework core support systems as its basis and adds new systems, in particular for sequencing of the production process.

A. Edge automation objective

According to the ISA-95 standard, the Manufacturing Execution System (MES) layer of the factory is responsible for Workflow and recipe control. Production orders generated by Enterprise Resource Planning (ERP) and Product Lifecycle

978-1-5386-6531-2/18/$31.00 ©2018 IEEE

Management (PLM) tools will create customized workflows according to the product specification and manufacturing requirements.

Edge Automation Services (EAS) role is to automate the execution of a production order at the edge tier. Meaning that the edge tier (edge gateway and smart objects) are able to complete the production order (or part of it) without reliance on the centralized MES layer. Alternatively, it means that the MES layer has been decentralized across the edge tier, with the edge automation services as the enabler.

It stands to benefit to define some of the core operations of the MES layer; Planning, Supply chain, Inventory, Quality of Service management, Tracking and Execution.

This paper focuses on the tracking and execution aspects of the MES. Tracking is made through the intelligent product unit that holds the planned workflow and the current state of activities. Execution is conducted through the Workflow manager, Workflow executor and smart objects such as intelligent equipment.

Once the MES data is available at the edge tier, EAS are able to complete the required production steps with very little or no connectivity to any cloud based systems. Moreover, Production Orders are also updated by EAS as a consequence of production steps being completed. A working copy of the production order is interacted with at the edge, this is a disconnected approach. The work copy is then propagated to the master copy once connectivity is available.

Overall, edge automation services are expected to have the following advantages over a traditional centralized architectures:

1) Shopfloor more tolerant to network and central system faults.
2) Shopfloor more modular and flexible, due to individual workstation (workcell) being autonomous.
3) Automation more reactive to shopfloor events, due to the shorter path between business logic and sensing/actuation hardware.

There are two situations where these advantages become very apparent. Where a workstation must be operated within an environment with no factory wide MES is available. This "autonomous" workstation only requires that the smart product it will operate on has an interface to communicate the workflow at the target station. This information that the smart product contains can be loaded to the product prior to shipping from the origin factory. Or can be transfered through some secure means to the workstation.

The second situation is that because the workflow does not specify execution details (this is the responsibility of the workflow executor orchestration), "plug-in-produce" is supported for equipment installation to the work station. Meaning that at runtime an "intelligent nutrunner" can be plugged into the workstation and utilized immediately.

1) Cyber Physical Systems: Edge Automation Services are directly related to Cyber Physical Systems. CPS are defined as networked embedded systems [3]. Embedded systems are computer systems that interact, sense and actuate, on their environment. The information carried by CPS and the services provided by CPS mean that MES functions that are traditionally centralized can now be decentralized.

This paper proposes a method of co-ordinating the efforts of CPSs within the framework of edge MES functionality.

II. Related work

Workflow, also known as business process, management has been a topic of research since the start of industrialization [4]. A workflow, or business process, is the combination of activities and information toward the creation of new value [5]. In [6], many reusable workflow patterns are documented and existing workflow frameworks are evaluated for support of the pattern. Workflow patterns are split into four core categories; Control, Resource, Data and Exception Handling workflows.

In [7], Sardis et al. describe a distributed workflow and monitoring system. They propose an architecture that combines rule based alarm detection, fault planning through preplanned behaviors, decision support and workflow monitoring. They suggest that a distributed architecture such as Multi-Agent Systems (MAS) or SOA be used for realization of the workflow manager and monitoring. However, they do not propose how the workflow system should be realized.

Quintanilla et al. propose a holonic approach toward workflow execution using SOA [8]. In this work, they propose 3 holons, the Product (PH), Order (OH) and Resource holons (RH). The PH is responsible for exploring workflow solutions, meaning that it performs planning and scheduling. The OH is responsible for execution of the workflow tasks specified by the PH. While the RH provides the manufacturing services utilized by the OH to complete the workflow. Petri-nets are used model the workflow and to evaluate alternatives.

REST and hypermedia has been proposed for work flow automation. For example in [9], Balis proposes a hypermedia driven workflow for data-driven scientific experiments. In this context a data driven workflow is implemented using an Hypermedia As The Engine Of Application State (HATEOAS) approach. Data and Tasks are addressed as resources and a client "executor" is responsible for following the hyper media links. The links have embedded an appropriate state machine. Clients and resources must have agreement on link relation meanings and resource types. The client executor knows the overall work flow state machine required to be executed. Starting by preparing data, the client moves through servers according to the link and relations returned by the servers.

This work closely resembles the work by Quintanilla et al. in [8]. There are clear differences with seperation of concerns and centralization of the Order holon. Manufacturing systems that require segregated work centers require decentralized task execution. Here, the proposed approach has a decentralized, stateless Workflow manager that implements a blackboard architecture pattern. The blackboard is available to all equipment systems operating in the workcenter. Meaning that, concurrent workflow description available can be acted upon by individual systems as needed.

III. Arrowhead Framework

The Arrowhead Framework is based on Service Oriented Architecture (SOA) [10]. It defines Systems as the containers of Services that are exchanged between systems and clients. The concept of local clouds is defined as a means of controlling Quality of Service (QoS), increasing security and segregating functionality. Within each local cloud there are several mandatory core systems and support core systems. These services provide functionality for discovery, access control, and interconnect management. Additionally they provide interoperability, storage and configuration services. In order to describe the plant layout and capability locations the Plant Description [11] support core system is defined. Figure III shows the main building blocks of the Arrowhead Framework.

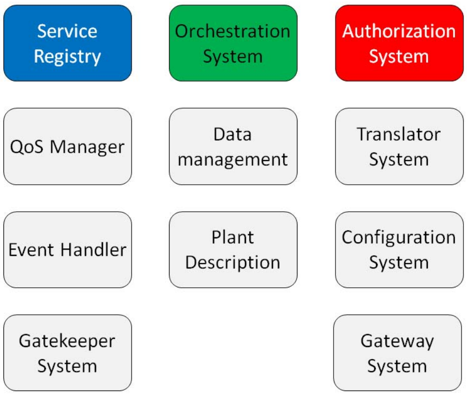

Fig. 2. The Arrowhead Framework block diagram of constituent systems.

The Arrowhead Framework is an active open source project that is being contributed to be European Union funded projects; Productive 4.0 and Far-Edge. The relevant parts of the Arrowhead Framework are described in the proposed solution.

IV. Work flow handling

The production order is the fundamental data required by the edge automation services. It is carried by CPS based smart products or a digital twin (where CPS is not available). Figure IV shows a block diagram of a smart product in a factory with edge automation services.

Therefore the edge automation services have an interaction point in the smart product unit. Also shown in the Figure IV the smart equipment at each work cell make up I4.0 components that interact with the edge automation services.

A. Production Order

The ISA95 describes a model for the connection between ERP and MES. But in this work a simpler model that uses a Product to Process approach has been taken, i.e. how to exchange information between a product and the process. Additionally a batch-size one approach is taken, where every product unit has a unique recipe that is configured directly from the specification ordered of the customer.

The production order consists of the production specification (customer requirement) and the product recipe (production requirement). The recipe contains the Bill Of Materials (BoM), List of Operations and Equipment Requirements. The recipe does not bind the operations to either equipment instances or to work stations. Shown in Figure IV-A the Entity Relation Diagram (ERD) of the production order can be seen.

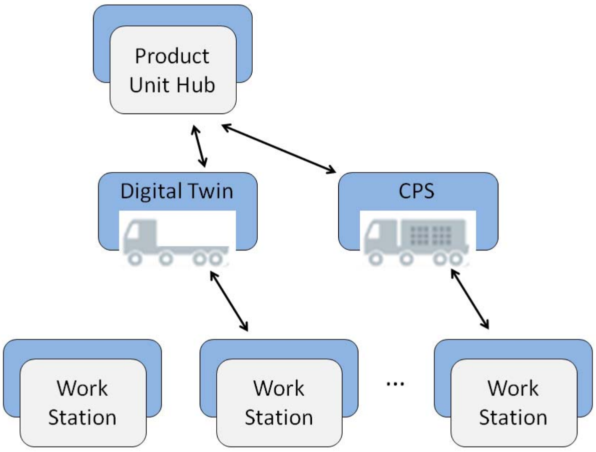

Fig. 3. High level perspective of Industry 4.0 enabled smart product and work cells

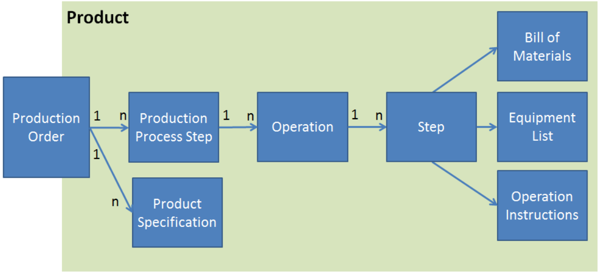

Fig. 4. High level Entity Relation Diagram of a production order

As described earlier, CPS-based products or their digital twins are containers of the production order. The synchronization between digital twin and physical product is made directly. It is not the focus of this paper and so is assumed to be available. The specification holds product variant information of the production order. The List of Operations is grouped into ordered work cells, meaning that a flow of instructions is defined and loosely grouped.

B. Production Orders to the factory edge

PLM and ERP tools generate production orders based on a product specification. The product specifications are created by engineers working with customers to identify requirements and solutions. The PLM/ERP tools maintain a direct link between customers, requirements, specifications, and the production order.

The product specification defines the production capabilities required to manufacture the product. The production process step of the the production order presented in Figure IV-A is mapped to a work station (cell). The mapping is decided based on known capabilities available at the work station. This can be done dynamically meaning that scheduling and routing must be performed by local systems. The planning function is not addressed in this paper. It can be assumed that the plan is statically created and that the product unit is the container. Factory, work center, station and device capabilities must be known when creating the production order.

V. Edge Automation Services

The proposed solution involves SOA-based systems running in the edge tier. In this section the individual systems which are used to achieve the automation objectives.

A. Arrowhead local cloud

The Arrowhead Framework local cloud is running on an edge gateway. It has all prerequisite components for Service Oriented applications to run on the edge tier. It is expected that individual edge gateways may have customized local clouds according to requirement. With support core systems only present and operating when required. Full authorization security is supported at the edge automation layer, meaning that edge gateway is protected by the Arrowhead security mechanisms. Readers can also refer to the Arrowhead Wiki for further documentation and notes [12].

B. Workflow Manager

The Workflow Manager is a SOA-based system developed for the Far-Edge platform [13] and operating within the Arrowhead local cloud paradigm. It is a form of book keeper that is responsible for managing progress of the production process/recipe and reporting back to the Factory Control System (FCS).

Here the FCS is a Volvo term, however it can also be seen as a function. The function it provides is for overall coordination of the factory. The FCS is not responsible for coordinating equipment, rather it is an information source for products and equipment to refer when requiring information outside their direct scope. Here it can be seen that classical MES vs. RAMI4.0 come into contact. The Control logic: Quality, Process, Logistics, etc. are placed inside the smart product/virtual twin rather than as a centralized service. However, the FCS remains as a migration point where not all information can be located within the I4.0 Component.

1) Interfaces: The Workflow Manager has 3 interfaces used to synchronize with smart product units (CPS/Digital Twin) and 2 interfaces toward smart equipment. The Workflow Manager consumes services provided by the product unit, making pull requests for production recipe information in a RESTful manner.

1) RetrieveProcessStep
2) StoreProcessStepResult
3) StoreOperationResult

The Workflow Manager only need identify itself (workstation id) to the product unit, as the recipe has operations allocated at a workstation level. It is assumed that this ID is set when the local cloud instance (edge gateway) is being setup. It must be completed before the Workflow Manager is able to take part in the edge automation activities. It is possible that these service interfaces will evolve to allow further parametrization as required. Toward the edge, the Workflow Manager is a service provider and will listen for service invocation requests in a RESTful manner.

1) RetrieveOperation
2) StoreOperationResult

The workflow manager services are designed such that they can be used by both Push and Pull communications patterns, such as MQTT or CoAP/HTTP. CoAP has been selected for the prototype implementation due to convenience of application libraries and simple integration with application code. Additionally CoAP, has advantages for RESTful communication with a push notification channel available. The usage of the interfaces are described in the next sections.

2) Operation: A Workflow Manager will remain passive awaiting notification that a product unit has arrived at the work station. On arrival, of a new product to the work station, the Workflow Manager will query the smart product (or its digital equivalent) for the recipe related to the the work station. The production order will be checked to make sure any dependencies between work station flows have been met. Only relevant steps of the recipe will then be passed to the work station (Workflow Manager).

With the new recipe loaded, the Workflow will begin with edge systems pulling work instructions. Once work instructions are completed, the results are sent back to the Workflow Manager. Overall completion of the workflow is decided by the Workflow Manager once all results are compiled. The final results are then sent back to the product unit with confirmation. A simplified version of this is shown diagrammatically in Figure V-B2. It illustrates a typical interaction between the Edge Automation Services. Here the third item Workstation equipment groups together any individual smart objects that can be coordinated through the Workflow Manager.

The Workflow Manager holds the digital instruction list as a form of state machine. This state machine is passed to an executor system within the Workstation. The executor system is customized and configured for a particular Workstation. The customizations are both in terms of configuration parameters and capability interactions. The capability interactions are defined by Orchestration rules, described under orchestration section. Figure V-B2 shows a block diagram of this interaction. The Workflow Manager passes the instruction list as a hypermedia enabled state machine. Using IETF RFC 5980 and 6690, namely, Web Linking and Core Link Format, it is possible to describe the state machine as a set of related resources.

C. Plant Description

The Arrowhead Plant Description system is responsible for maintaining a functional and topological map of the systems available in the local cloud. This means that the plant description has knowledge regarding the edge nodes connected to the edge gateway. It is kept up to date through the Arrowhead Device and System Registries operating within

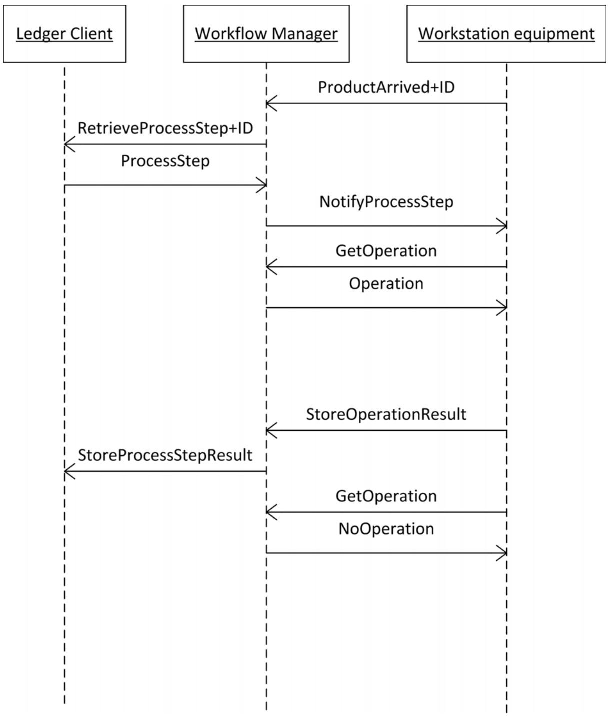

Fig. 5. The Workflow Manager interacts with its peers

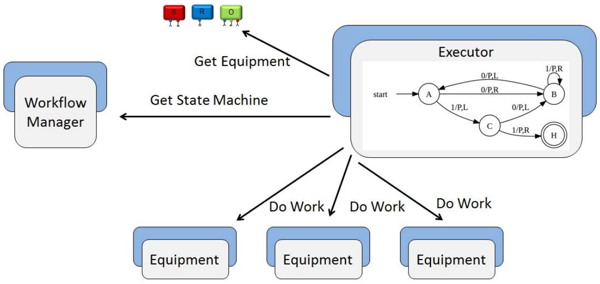

Fig. 6. The Executor is a HATEOAS enabled application state machine.

the Field Abstraction component. The information stored in the plant description is used for engineering the edge tier gateway and nodes [11].

D. Orchestration System

The Arrowhead Orchestration System enables central control of system coordination. It holds information on how systems should be composed into Systems of Systems. So when the workflow executor needs to make use of equipment of a certain capability it will request connection information from the Orchestration System and thereby locate the correct service instance representing the equipment.

E. Service Registry

The Arrowhead Service Registry holds information on active services available at the edge gateway. This information is used to locate service provider instances. Edge nodes will register (advertise) their capabilities through services registered here. Edge automation services such as the executor system will find the end-point address of the capability required to complete the operations in the process step.

F. Authorization System

Edge nodes and edge automation services are protected from un-authorized access through the Arrowhead Authorization system. This system is providing the PIP and PDP according to the NGAC terminology. A system providing services will enforce the policy decision made by the Arrowhead Authorization system.

VI. Use case implementation

The prototype implementation of the edge automation services presented in this paper is validated in an industrial use case. In this section a high level description of the use case is described. It is intended to run a small scale pilot of the use case in a parallel production environment (using non-production data).

The use case involves two key validation criteria: Plug and Produce Equipment and Autonomous Workstation. In the first case, an Industrial Nutrunner is enhanced with an administrative shell. It is shown that the smart Nutrunner can be moved between work stations and its capabilities made use of without manual configuration of the work station.

In the second case the entire workstation is self contained so that it can be ”moved” to an external site and still work. This means that it has operational independence from the factory control system. A production order may be delivered to it digitally or manually in a secure format.

The business benefits of the use case can be sumarized as:

1) Simplify or eliminate configuration and engineering of equipment.
2) Reduce IT integration costs and reliance on IT competence resources.
3) Flexibility of resource location.
4) Enabling continuous improvement through reduction in effort to realize change.

Figure VI shows the system block diagram of the use case solution. In the figure the FCS Adapter is a temporary system that serves as a migration path. It serves the same function as the Product Unit Hub described in the original platform.

The Workflow manager and executor systems can be seen in Figures V-B2 and VI. These systems create the link between the CPS Smart Product and Smart Equipment. The Service links are discovered through the Arrowhead Service Registry and Orchestration Systems.

In this prototype implementation, an Atlas Copco P4000 Nutrunner is interacted with through an administration shell implemented as a SOA-based System. It is a service provider that registers its services with the Arrowhead Service Registry. The I4.0 Nutrunner is shown in Figure VI.

A multi-protocol approach is taken to demonstrate the interoperability between different systems. This means that the Nutrunner uses MQTT for its service provision, while CoAP is used by the Workflow Manager and HTTP is utilized by the Core Systems.

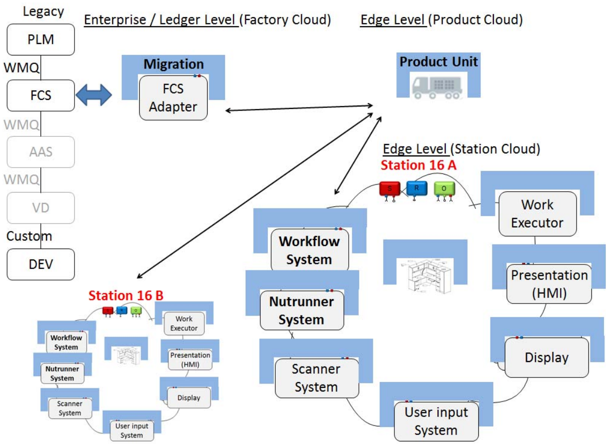

Fig. 7. Block diagram of RAMI 4.0 enabled work station utilizing Edge Automation Services from the Far-Edge platform and the Arrowhead Framework.

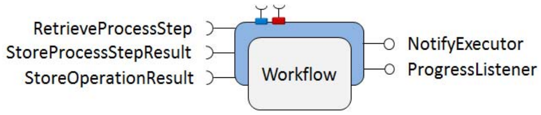

Fig. 8. I4.0 enabled Workflow Manager System with administrative shell and services. Up stream services are on the left side and down stream services are to the right.

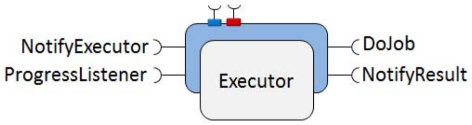

Fig. 9. I4.0 enabled Workflow execution system with administrative shell and services. Up stream service connecting to the Workflow manager are to the left and down stream services connecting to the smart equipment and objects are to the right.

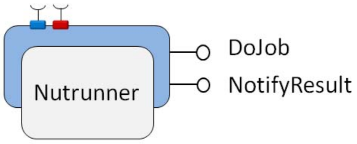

Fig. 10. I4.0 enabled Nutrunner with administrative shell and services. DoJob and NotifyResult are standard services for generic equipment requests.

VII. Future work

Still to be added is the configuration integration and the plant description updates. Being able to generate the Orchestration rules through the plant description and then utilizing a workflow through the Workflow Manager and Executor systems. Fully utilizing the Next Generation Access Control mechanism in the Arrowhead Authorization system.

VIII. Conclusion

Demonstrated in this paper is an approach to decentralizing the ISA 95 architecture utilizing the RAMI 4.0 approach on edge computing and SOA-based system design. A Workflow Manage is proposed as the integration point of MES data between smart products and smart equipment. Reducing the demands on centralized infrastructure whilst maintaining centralized resource planning of the overall production. The proposed solution is demonstrated on an industrial use case.

Summarizing the Arrowhead concepts can be described as:

1) Plant Description is used for planning and scheduling.
2) Work flow Manager is used for coordination and sequencing.
3) Orchestration is used for realizing the Plant Description.

In this regard, the paper has also contributing to creating an seperation of concerns between the Arrowhead Plant Description, Orchestration and Work flow management.

Acknowledgment

The authors would like to thank the Far-Edge and Productive 4.0 projects for funding and sharing thoughts through discussion and debate.

References

[1] "The Fourth Industrial Revolution - What it means and how to respond. World Economic Forum," Jan 2018. [Online]. Available: https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/

[2] "The Reference Architecture Model RAMI 4.0 and the Industrie 4.0 component - zvei.org," August 2017. [Online]. Available: https://www.zvei.org/en/subjects/industry-4-0/the-reference-architectural-model-rami-40-and-the-industrie-40-component/

[3] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, May 2008, pp. 363–369.

[4] W. M. van der Aalst and K. M. van Hee, *Workflow Management: Models, Methods, and Systems*. The MIT Press, 2000.

[5] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business process management: A survey," in *Proceedings of the 1st International Conference on Business Process Management, volume 2678 of LNCS*. Springer-Verlag, 2003, pp. 1–12.

[6] (2018) Workflow patterns. [Online]. Available: http://workflowpatterns.com/

[7] E. Sardis, V. Anagnostopoulos, A. Voulodimos, and T. Varvarigou, "On surveillance and workflow control system design," in *2010 IEEE International Conference on Progress in Informatics and Computing*, vol. 1, Dec 2010, pp. 267–271.

[8] F. G. Quintanilla, O. Cardin, and P. Castagna, *Product Specification for Flexible Workflow Orchestrations in Service Oriented Holonic Manufacturing Systems*. Cham: Springer International Publishing, 2014, pp. 177–193. [Online]. Available: https://doi.org/10.1007/978-3-319-04735-5_12

[9] B. Balis, "Hypermedia workflow: A new approach to data-driven scientific workflows," in *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, Nov 2012, pp. 100–107.

[10] J. Delsing, *IoT Automation Arrowhead Framework*. CRC Press, 2017.

[11] O. Carlsson, C. Hegeds, J. Delsing, and P. Varga, "Organizing iot systems-of-systems from standardized engineering data," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 5277–5282.

[12] "Arrowhead Framework Wiki," May 2016. [Online]. Available: https://forge.soa4d.org/plugins/mediawiki/wiki/arrowhead-f/index.php/Main_Page

[13] "Far-Edge Project," Jan 2018. [Online]. Available: http://www.faredge.eu/