Jouko Virta

APPLICATION INTEGRATION FOR PRODUCTION OPERATIONS MANAGEMENT USING OPC UNIFIED ARCHITECTURE

Faculty of Electronics, Communications and Automation

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 5.5.2010

Thesis supervisor:

Prof. Kari Koskinen

Thesis instructor:

Dr.Sc. (Tech.) Ilkka Seilonen

Aalto University
School of Science
and Technology

AALTO UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author:	Jouko Virta		
Title:	Application integration for production operations management using OPC Unified Architecture		
Date:	May 5th, 2010	**Number of pages:**	10 + 59
Faculty:	Faculty of Electronics, Communications and Automation		
Department:	Department of Automation and Systems Technology		
Professorship:	AS-116 Information and Computer Systems in Automation		
Supervisor:	Prof. Kari Koskinen		
Instructor:	Ilkka Seilonen, Dr.Sc.(Tech.)		

A modern manufacturing company employs many information systems which require both vertical and horizontal integration in order to enable effective manufacturing operations management. The integration confronts many challenges, like different interfaces, a large amount of information to exchange and the lack of standardized information models.

OPC Unified Architecture (OPC UA) specification defines a platform-independent data exchange interface which enables exchanging complex information models. It provides an interface not only for read and write operations and method calls but also for transmitting asynchronous events. The ISA-95 standard can be used as the basis for designing the system integration in an abstract level. It defines the manufacturing execution system (MES) activities, their relations and the data exchanged between them. The ISA-88 standard defines equipment hierarchies, procedures and recipe structures characteristic to batch process systems.

This study is focused on utilizing OPC UA in conjunction with a service-oriented integration middleware and the ISA-88/95 standards for integration between an MES and a process control system in the context of batch process management. First, the requirements of integration are identified and then an integration design combining the mentioned technologies is proposed. The design is evaluated with an experimental implementation and test scenarios. The result of the work is a testbed environment which is usable for research and educational purposes in the future. Based on this work, one can state that the utilization of the ISA-88/95 standards simplifies the integration design process considerably and OPC UA combined with a service-oriented integration middleware is quite an adequate tool to complement the design.

Keywords: OPC UA, manufacturing execution system, MES, application integration, ISA-95, ISA-88, batch process, automation

AALTO-YLIOPISTON TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Jouko Virta		
Työn nimi:	Tuotannonohjaustoimintojen sovellusintegraatio OPC Unified Architecture -tekniikkaa hyödyntäen		
Päivämäärä:	5.5.2010	**Sivuja:**	10 + 59
Tiedekunta:	Elektroniikan, tietoliikenteen ja automaation tiedekunta		
Laitos:	Automaatio- ja systeemitekniikan laitos		
Professuuri:	AS-116 Automaation tietotekniikka		
Työn valvoja:	Prof. Kari Koskinen		
Työn ohjaaja:	TkT Ilkka Seilonen		

Nykyaikaisessa tuotantolaitoksessa on monia tietojärjestelmiä, joiden vertikaalinen ja horisontaalinen integraatio on edellytys tehokkaalle tuotannonohjaukselle. Järjestelmien integroinnin ongelmia ovat monet erilaiset rajapinnat, välitettävän tiedon määrä ja yhtenäisten tietomallien puute.

Uusi OPC UA -spesifikaatio määrittelee alustariippumattoman tiedonsiirtorajapinnan, joka mahdollistaa semantiikaltaan monimutkaisten tietomallien käyttämisen tiedonsiirrossa. Se tarjoaa rajapinnan paitsi luku- ja kirjoitusoperaatioille sekä metodien kutsumiselle, myös asynkronisten tapahtumien välittämiselle. ISA-95-standardin avulla valmistuksenohjaus- ja automaatiojärjestelmän integraation arkkitehtuuri voidaan määritellä abstraktilla tasolla. Standardi jakaa integraatioon vaadittavat valmistuksenohjaustason toiminnallisuudet ryhmiin ja määrittelee myös niiden välisen tiedonsiirron sisällön. ISA-88-standardi määrittelee panosprosesseille tyypilliset laitehierarkiat, proseduurit sekä reseptitietorakenteet.

Tässä työssä tarkastellaan kuinka OPC UA soveltuu prosessinohjaus- ja valmistuksenohjausjärjestelmien integraatioon. Työssä tarkastellaan myös kuinka ISA-88- ja ISA-95-standardien määrittelemiä tietorakenteita sekä hierarkia- ja järjestelmämalleja hyödynnetään valmistuksen- ja prosessinohjausjärjestelmien integraatiossa. Toteutettu opetus- ja tutkimuskäyttöön tarkoitettu järjestelmäympäristö osoittaa ISA-88- ja ISA-95-standardien sujuvan käytön OPC UA -teknologian kanssa. Se osoittaa myös, että OPC UA -palvelujen sisällyttäminen niin valmistuksenohjaus- kuin prosessinohjausjärjestelmiinkin mahdollistaa tulevaisuudessa tehokkaan ja yksinkertaisen järjestelmäintegraation.

Avainsanat: OPC UA, valmistuksenohjausjärjestelmä, MES, järjestelmäintegraatio, ISA-95, ISA-88, panosprosessi, automaatio

Preface

This thesis has been written at the laboratory of Information and Computer Systems in Automation at the School of Science and Technology at the Aalto University and is a part of the POJo research project.

I would like to thank my instructor Ilkka Seilonen for all the instruction and guidance during this process. I am also grateful to my supervisor, professor Kari Koskinen for giving me the opportunity to work in this research group during my studies. It has been pleasant to work with such great people in this stress free atmosphere.

I sincerely thank my dear mother for sending me to school every morning and for all the encouragement and support during my later studies in academia. Words cannot express the gratitude I feel for the possibilities you have provided me for my life.

Finally, I wish to thank you Noora for the love, support and patience.

Otaniemi, 5.5.2010

Jouko Virta

Contents

Abbreviations

ANSI	American National Standards Institute
API	Application Programming Interface
B2B	Business-to-Business
B2MML	Business-to-Manufacturing Markup Language
BatchML	Batch Markup Language
BPEL	Business Process Execution Language
BPIOAI	Business Process Integration-Oriented Application Integration
CAEX	Computer Aided Engineering eXchange
CM	Condition Monitoring
COM	Component Object Model
DCOM	Distributed Component Object Model
DCS	Distributed Control System
DLL	Dynamic-Link Library
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
FIM	Financial Information Management
HMI	Human-Machine Interface
HRM	Human Resources Management
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IOAI	Information-Oriented Application Integration
IOHN	Integrated Operations in the High North
ISA	Instrumentation, Systems, and Automation Society
KPI	Key Performance Indicator
LOB	Line-Of-Business
MES	Manufacturing Execution System
OEE	Overall Equipment Effectiveness
OLE	Object Linking and Embedding

OOP	Object-Oriented Programming
OPC	Open connectivity via open standards
PCS	Process Control System
PIM	Product Information Management
PLC	Programmable Logic Controller
POAI	Portal-Oriented Application Integration
POM	Production Operations Management
SCADA	Supervisory Control And Data Acquisition
SCM	Supply Chain Management
SDK	Software Development Kit
SOA	Service-Oriented Architecture
SOAI	Service-Oriented Application Integration
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SRM	Supplier Relationship Management
TCP	Transmission Control Protocol
UA	Unified Architecture
UML	Unified Modeling Language
W3C	World Wide Web Consortium
WBF	World Batch Forum
WCF	Windows Communication Foundation
WF	Workflow Foundation
WPF	Windows Presentation Foundation
WS	Web Service
XML	eXtensible Markup Language
XSD	XML Schema Document

List of Figures

1 Introduction

1.1 Background

A modern manufacturing company contains many information systems, for example a resource planning system for business-oriented company administration, a manufacturing execution system for monitoring and controlling the manufacturing processes and automated process control systems for controlling the process equipment. It may also employ many subcontractors which again have their own information systems. This means that the supply chains are long yet the manufacturing company needs to manage the chain information in real-time. These mentioned systems are often implemented by different system providers, use different terminology, interfaces, protocols and data structures, have different considerations of importance of information and may situate all over the world.

In the modern markets, the manufacturing companies need to be efficient; they must be able to adapt to the market changes, provide a wide range of products and their production processes must be cost-effective. This requires that the information flow within the manufacturing company's own information systems as well as between the systems of other companies in the supply chain needs to be fluent; the information systems must be integrated to each another and there shall not be any communication breaks between the systems.

In order to achieve the required efficiency, the integration of manufacturing operations management must be implemented between the information systems. This may mean both vertical and horizontal integration in the company system hierarchy. The integration is however a complex task and some techniques have been developed to simplify the many problems.

OPC Unified Architecture (OPC UA) is a new standard for communication in industrial applications developed by the OPC Foundation. An important motivation for OPC UA has been the drawbacks of the previous OPC specifications. Because of this, OPC UA contains several important enhancements, for example the service-oriented architecture (SOA), data security and configurable information models. The configurability of the information models makes it possible to utilize standardized data models, e.g. ISA-95 and ISA-88, in conjunction with OPC UA. However, there are not yet many experiences how the mentioned standards should be used with OPC UA and what would be the actual benefits of this.

The manufacturing execution systems (MES) have increasingly been adopted as information systems for production operations management in manufacturing companies. The ISA-95 standard is a partially finished standard aimed for enhancing the development of MES applications and integrating them to other information systems of manufacturing companies, particularly enterprise resource planning (ERP) systems. The ISA-95 defines the production operations management activity and data models which may be used together with the respective models of the ISA-88 for batch process automation. In the case OPC UA will be used as a communication mechanism between an MES and a process control system (PCS), it would be possible to use the mentioned data models during the communication. In this way the application developers could use the same concepts both in the development of MES applications and in the integration of MES and PCS systems.

1.2 Objectives

Since the integration of manufacturing operations management is such a big problem, the scope of this study is limited; it lays in the integration of production operations management activities between a manufacturing execution system and a process control system in the context of batch process management. The purpose of this study is to present an approach for combing the OPC UA technology with an SOA-based middleware and the ISA-88/95 standards for the integration of production operations of a manufacturing execution system with a process control system.

The approach contains the identification of the requirements of the production operations management integration that can be enhanced with the utilization of the mentioned techniques and a software design that enable the combination of them. The design is evaluated with an experimental implementation. The properties of the design are assessed in order to make conclusions about the approach and to propose topics for further research and development.

This thesis is made as a part of the POJo [1] research project. As a result of the project, a "testbed" environment is implemented. It is a system environment which aims to demonstrate a real manufacturing company with its many information systems, including a manufacturing execution system, an enterprise asset management system, a distributed control system and a condition monitoring system. The testbed is intended to provide a platform for future research as well as for educational student projects.

1.2.1 Research questions

The main research question in the scope of this thesis was the applicability of the OPC UA technology to the integration of information systems in manufacturing. The goal was to evaluate the features of the OPC UA technology in the service-oriented application integration as extensively as possible within the boundaries of this thesis.

In the context of the integration of production operations management, the goal was to find a reasonable design for an integration solution using the service-oriented architecture, the OPC UA technology and the ISA-88/95 standards. The design proposed in this thesis covers the data exchange interfaces of the information systems, the representation of the data as well as the architecture of an SOA-based integration solution.

[1] POJo is a research project in the second work package of the Intelligent and resource-efficient production technologies research program, EffTech, of Forestcluster Ltd. The project is funded by Tekes.

Also the application of the data structures of the ISA-88/95 standards and their suitability to the application integration using them in configurable address spaces of OPC UA servers was studied.

1.2.2 Research methods

The requirements for the integration of the production operations management were identified based on a review of relevant literature and other existing research on the subject. The service-oriented application integration itself is quite extensively studied already. However, quite few publications concerning the application of OPC UA in the application integration are released and they are rather preliminary. Also the topic of application of the ISA-88/95 standards in the manufacturing-to-shop floor integration seems to be still more or less theoretical; few publications of implementations exist.

Based on the identified requirements, an integration design is proposed. The design is then evaluated with the help of an experimental prototype implementation and test scenarios.

1.3 Outline of the thesis

This thesis is outlined as follows.

Chapter 2 discusses the production operations management as a part of the manufacturing operations management based on the definitions in the ISA-95 standard. Also the information systems involved in the production operations management are studied.

Chapter 3 discusses the topic of the application integration from a general viewpoint but also in the context of the integration of production operations management. Also the related research on the topic of this thesis is presented. The OPC UA technology and the ISA-88/95 standards are studied later in this chapter.

Chapter 4 presents the integration research platform; the system environment to be implemented in the POJo research project for future research and educational activities.

Chapter 5 discusses the requirements for the integration of production operations management. The requirements for the integration solution are presented as use cases.

Chapter 6 presents the proposed design. At first, an overview of the integration solution is presented, then the components of the solution are discussed in more detail.

Chapter 7 presents the experimental prototype implementation of the proposed integration design and demonstrates its functionality with the performed tests.

Chapter 8 contains the conclusions and discussion about the future work.

2 Production Operations Management

In order to be economically viable in the global competition of the modern markets, manufacturing companies need to be adaptive to the market situations. Nowadays, it is a buyer's market in the manufacturing business which means that manufacturing companies must be able to provide products customized to buyers' needs. Therefore the manufacturing companies have moved on to provide manufacturing services, and they will not necessarily any longer stand out from the markets by the quality of the product but rather the quality of their manufacturing processes. The high quality of the manufacturing process results to production flexibility, reduction of the time-to-market and delivery times and higher capability to provide a wider range of product variants, which are all significant advantages in the modern, demand-driven market conditions. [1, 2]

To achieve the previously described level of adaptivity, an extensive integration of manufacturing information and operations must be implemented. Since the integration of information systems involved in the manufacturing is such a wide subject, the focus in the scope of this work is on the integration of production operations management between manufacturing execution systems and process control systems in the context of batch process management. To understand the requirements of the integration scenarios and to be able to design and create a well-performing integration design, the hierarchy and the functionalities of the systems must be first studied.

The definition of the production operations management is one part of the concept of the manufacturing operations management which is based on the functional hierarchy model and the functional enterprise-control model defined in the part 1 of the ISA-95 standard. These models are discussed in the following section. [3, 4]

2.1 Manufacturing operations management

This section studies the relevant model definitions and the manufacturing operations management activities after the ISA-95 standard. The study is mainly based on the part 1 [3] and part 3 [4] of the ISA-95 standard.

2.1.1 Functional hierarchy model

The part 1 of the ISA-95 standard defines a functional hierarchy model, which represents the hierarchy of the information systems of a manufacturing company divided into levels. The functional hierarchy model is shown in the figure 1 and explained below.

The level 4, "Business planning and logistics" includes the enterprise-level functionalities which are often represented by an enterprise resource planning (ERP) system. An ERP system provides tools for managing the business-related activities of a manufacturing company. The most important function group of an ERP system

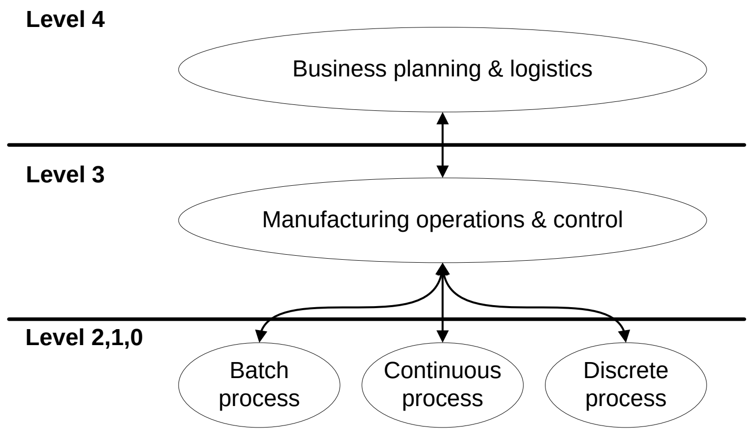

Figure 1: The functional hierarchy model [3]

is the finance management (FIM) which is used for accounting, sales and financial functions. Other important functions are the supply chain management (SCM), the human resources management (HRM) and the supplier relationship management (SRM). The SCM functions are used for planning the medium- and long-term material and energy requirements and the logistics involved in the manufacturing process. The HRM functions are used for managing the personnel of the company and the SRM for managing the supplier relationships, i.e. customer information which is important for the sales, marketing and purchasing functions of the company. The time horizon in the enterprise domain processes is days, weeks, months or even years. [5]

The level 3 is the production management layer, which contains the manufacturing execution functionalities. The level 3 functionalities will be examined in detail in the next section. The time frame of the level 3 processes range from days to seconds. Levels 2 and 1 contain the process control functions. The level 1 contains sensitive and manipulative functions and typically has a time horizon of seconds or less whereas the level 2 contains functions for monitoring and controlling the process and typically operates on time frames of hours, minutes, seconds or even less. The level 0 is the process itself. The combination of the MES- and PCS-layers is referred as the control domain.

2.1.2 Production operations management activity model

The part 1 of the ISA-95 standard includes the functional enterprise-control model which defines the 12 main functions of a manufacturing company. The names and importances of the functions may vary between the different companies but the functions still exist on some extent [6]. The functional enterprise-control model also presents the information exchange interactions between the functions. What is of great significance, the model represents whether the functions belong to the enterprise domain, control domain or both. The dotted line represents the interface between the enterprise and control domains and divides the functions accordingly.

The functional enterprise-control model is studied in more detail in the part 3 of the ISA-95 standard where the manufacturing operations management model is defined. The model is presented in the figure 2 and explained below.

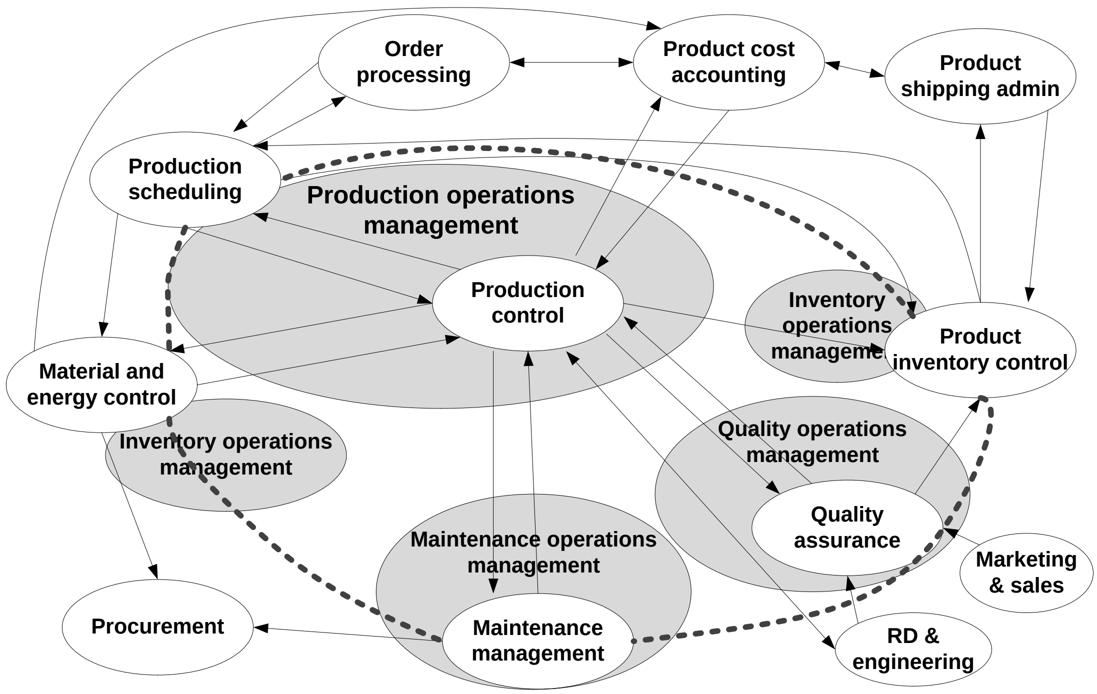

Figure 2: The manufacturing operations management model [4]

The manufacturing operations management consists of the activities within the level 3 that are used to manage the resources, for example personnel, equipment, material and energy required for production. This means managing information about schedules, capability, availability, capacity and the current statuses of the resources involved in the manufacturing processes. The previously introduced manufacturing operations model is based on the functional enterprise-control model but it sets the focus to the control domain by selecting four function groups within this domain: the production, inventory, maintenance and quality activities as illustrated in the figure 2. The part 3 of the ISA-95 standard defines one generic activity model plus specific activity models for each of the four activity groups. The activity model of production operations management is presented in the figure 3.

The production operations management is defined as the activities that coordinate, direct, manage and track the functions that use information of material, equipment, personnel and other resources to produce products considering the requirements of costs, qualities, quantities, safety and timeliness. The production operations management contains all the production control functions (the production control group in the figure 2) and the subset of production scheduling functions (the production scheduling group in the figure 2) that are part of the level 3 domain.

The information exchanged between the levels 3 and 4 are divided into four groups; the product definition, production capability, production schedule, and production

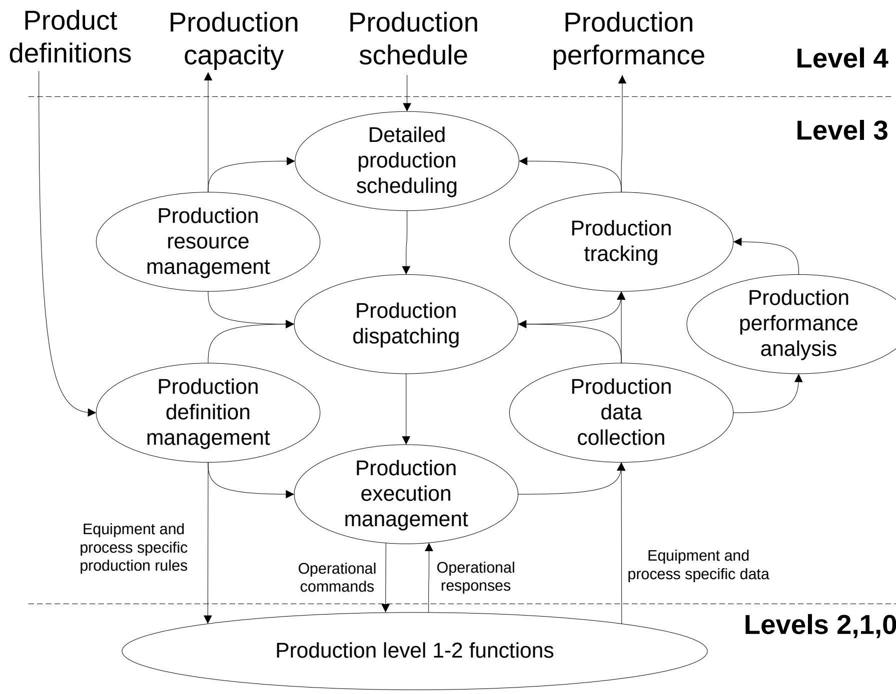

Figure 3: The production operations management activity model [4]

performance information whereas the information exchanged between the levels 3 and lower consists of production rules, operational commands and responses and equipment and process specific data. The production rules mean for example machine programs or recipes. The operational commands are for example requests to start the process or to initialize the equipment and the operational responses are typically received responses to the requests. The process specific data consists of information about the process execution or the resources involved received from the level 2.

Production operations management activities

A coarse production schedule is created in the enterprise level planning system but the detailed scheduling of work orders happens in the control domain. The detailed schedule contains information about local resources required for manufacturing the product whereas the coarse schedule is more business-oriented. The detailed production scheduling activity optimizes the production schedule based on the production definitions, real-time capacity and resource availability information. The optimization may mean for example minimizing the equipment usage or merging or splitting the work orders. Merging and splitting the production orders may be necessary to minimize the setup and cleaning times or to optimize the batch sizes. [6]

The production resource management activity handles the information about the equipment, personnel, material and energy resources required for production. With the resource and current process information, the activity can calculate the production capability information which is passed to the upper-level planning systems. The production capability information may also be used to optimize the production schedules. The production resource management activity handles the resource information based on the current availability information, production schedule and future plans (e.g. future production plan, maintenance schedules and vacations). It also manages the resource reservations according to the schedule.

The product definition management activity handles the product definitions, for example ANSI/ISA-88 recipes, which contain information about production rules and required resources (bill of material, bill of resources). This activity manages the product definition information exchange to the product information management system (PIM) on the level 4 at the necessary level of detail. It also manages the changes to the product definitions and provides the information to other activities within the level 3. It also may need to translate the received product definition information to comply with the site-specific resource information.

The production dispatching activity manages the production by assigning production tasks to the production personnel and systems. This means sending production work orders to the production systems and work centers according to the schedule. It also maintains the current statuses of the work orders, allocates resources, receives information about the resource availability and quality issues and delivers this information to the detailed process scheduling activity. The production orders to be dispatched are maintained in a dispatch list, for example in a ISA-88 batch list in a batch process system.

The production execution management activity coordinates and controls the work order execution by launching process control functionalities on the level 2 system based on the dispatch list maintained by the production dispatching activity. To achieve a real-time and reliable execution management, this activity must be closely connected to the lower-level process control systems. The production execution management activity also makes sure that the production is carried out using the right resources and according to the required quality standards.

The production data collection activity collects and archives information from the level 2 system. The collected information is specific to production processes or work orders and includes for example events, alarms, process data, equipment usage and sensor readings. The data is collected and maintained to enable the production reporting, analysis, tracking, tracing and monitoring functionalities. [4]

The production tracking activity records the resource movements, changes and usage, the process information and events and prepares the information reports for the upper-level systems. This activity summarizes all the production process-related information, for example the resources involved and materials consumed, to production performance information which is important for the analysis functions of the business-oriented systems in the level 4.

Some analysis operations are done to the process data in the production performance analysis activity group. The analysis means for example measuring the cycle times or calculating resources and equipment utilization by evaluating the process data against the resource availability information. This analysis provides important information about the production process efficiency as well as product quality and can be used for production optimization in the level 4. The performance report may contain for example comparison of production units or production runs and calculations of production key performance indicators (KPI).

2.2 Information systems of the production operations management

The development of the information systems on the different hierarchy levels has not been concurrent. The earliest enterprise-level information systems were designed for automating the accounting and inventory management in the 1970s [7, 5]. Since then, many other enterprise management tools, like the resource management and the production planning, have been integrated to the enterprise-level information systems making the enterprise-level systems more dependent of the current production data. Modern systems can be used to manage resources, i.e. human, material, and equipment resources, as well as sales and marketing information. However, the ERP systems themselves are business-oriented and meant to be financial management tools for the company administration, not for controlling and monitoring the manufacturing processes. This has formed a gap between the ERP and automated process control systems [5]. The gap resulted a few problems: without a close connection from the process control systems to the ERP, the currency and the reliability of the received data was unsure, the amounts of data were huge and the data was difficult to analyze because it was based on the other department's conception of importance [8].

With the enterprise-level systems becoming more common and the global competition raising its head, the "order-to-shop-floor" cycle times became the measure of efficiency in production. Modern manufacturing companies must be adaptive and flexible; react to the market changes fast, providing short time-to-market times and a wide range of products, even customized according to the customer's needs. Traditional enterprise-level systems had severe limitations to efficiency: they were not closely integrated to the production systems. The time resolution in production planning is long, typically days or even weeks. Without a close connection to the production process, the production planning is only suggestive: it is not based on the actual processing times, the capacity utilization or the future load horizon. It also cannot react to exceptions in production, for example faults in the production equipment. [1]

In the past decades, the manufacturing companies have invested heavily to ERP systems but the problem due to the gap between the enterprise and shop-floor systems has become a critical factor reducing the possibilities of efficient production management [6]. Several organizations have been working on this issue and published recommendations for a solution thus the concept of a manufacturing execution system (MES) has formed. A manufacturing execution system is a "hub" directing, storing, processing and providing information between the enterprise and process control level systems [9]. The Manufacturing Enterprise Solutions Association (MESA) was the first organization to define the requirements of an MES by publishing a paper which defined 11 functionalities of an MES [7]. These functionalities are explained briefly below.

1. **Resource allocation and status:** The management of resources like equipment, labor and materials include checking the availability, making reservations and setting them up for production.
2. **Detailed scheduling:** With fine scheduling, the production can be optimized based on the resource availability, to minimize the setup needs and to balance the equipment loading.
3. **Dispatching:** The system manages the execution of the production orders based on the production schedule. The system must be able to react to unexpected production events, for example by changing the production plan in case of an equipment fault.
4. **Document control:** The management of production-related information, for example work instructions, recipes and equipment information. Necessary information is provided to the operators and machines.
5. **Data acquisition:** The production data must be collected from the factory floor in order to implement the production analysis, tracing and tracking functionalities based on the actual production information.
6. **Labor management:** The labor management includes for example work time logging, task allocation and activity tracking functionalities.
7. **Quality management:** The analysis based on the real-time production data allows detecting product quality problems and reacting to them rapidly.
8. **Process management:** Enables monitoring the actual production process and tuning the process if needed.
9. **Maintenance management:** The collected maintenance information (for example service hours, diagnostics, alarms) is used to manage the equipment maintenance tasks to ensure their availability.
10. **Product tracking:** The previously collected production data can be used to trace individual end products or batches. The tracking covers the entire production chain, including even the material quality information from the subcontractors.
11. **Performance analysis:** Provides the performance analysis information which is based on the collected production data. The performance data can include for example calculated values of overall equipment efficiency (OEE), key performance information (KPI) and resource utilization and availability. This

information is used to form production performance reports for the system operators.

The 11 functionalities of an MES defined by the MESA resemble closely the production operations management activity model defined in the part 3 of the ISA-95 standard and explained in the previous section. This is because the MESA model was the starting point for the standard committee when developing the ISA-95 activity models. The MESA model, however, was based on the existing MES systems whereas the ISA-95 models aim to be general, vendor-independent models. In addition, the MESA model does not specify the relations of the functionalities, differentiate the hierarchy levels they include in nor the flow of information between the functionalities. The ISA-95 definitions of the system hierarchies, the level 3 activities, their relations and information exchange make it a suitable base definition of a modern manufacturing execution system. [6]

3 Application integration in production operations management

A modern manufacturing company contains several information systems designed to handle different tasks and typically implemented by different system vendors with different technologies. Simultaneously, manufacturing companies face requirements to manage global supply chains and to be adaptive to the demand-driven market situations and customers' needs. This requires efficient integration of the manufacturing operations management functionalities within the system environment of a manufacturing company [10]. Application integration has many challenges (different technologies, non-uniform data structures etc.) which can be solved by using suitable integration methods. This section studies the concept of application integration after Linthicum [11] and explains modern integration technologies, including the integration architectures, data exchange interfaces and standards essential to this thesis.

3.1 Application integration in general

The concept of application integration can be divided into two forms, an external application integration, business-to-business (B2B) integration and an internal application integration, enterprise application integration (EAI). In the scope of this work the integration of systems within a manufacturing company is studied, so the focus is on the EAI, the internal form of application integration. In the EAI the integration aims to create a seamless information flow between the enterprise's internal systems so the information systems of a company would form one collaborative system environment.

As the information systems of manufacturing companies are becoming more complex and versatile, they are moving towards a service-oriented architecture (SOA) where they can be seen as service providers. The information systems of a manufacturing company provide services like analysis of data, process execution and product definition management, consumed by other systems. The service-oriented application integration makes it possible to compose different systems and their data to one collaborative application environment.

The application integration approaches can generally be divided into four types:

- Information-oriented
- Business process integration-oriented
- Service-oriented
- Portal-oriented

The information-oriented application integration (IOAI) is basically data exchange between databases. The data can be replicated between the databases (data replication), several databases can be combined to act as one virtual database (data federation) or the data sources can be connected to other systems using well-defined interfaces (interface processing). The information-oriented integration is an efficient approach to integration of simple data. However, it is not suitable for integration which needs functionalities integrated as well. It is not applicable to integration scenario which involves business processes or remote services.

The business process integration-oriented approach is based on the idea that the integration is implemented with well-defined business processes controlling the data movements between the systems and the execution of services in the specified order. The business process integration-oriented application integration (BPIOAI) often involves a middleware application, for example an integration server. The business processes can perform complex tasks by composing remote services and moving data between systems as specified in the business process rules. With the BPIOAI, the integration logic can be separated from the system-specific details resulting to a simple but really flexible integration approach.

The service-oriented application integration (SOAI) means composing applications or remote services to common methods which are then shared using for example the web services technology. It is suitable for the integration of functionalities, where the business logic wrapped into remote services needs to be shared and the reuse of it is valuable.

The portal-oriented application integration (POAI) means providing a single-user interface for the systems. The applications are not directly integrated to each other but instead they are integrated through the user interface, for example a web browser. Unlike the other integration approaches, the idea is to compose and provide current information in the user interface, not to exchange data between the systems.

The focus in this work lays on the business process integration-oriented application integration as it is considered to be the future technology for application integration within and between companies. It has the characteristics of both IOAI and SOAI; it defines the logic of executing information movements and consuming remote services in the business process models. It is suitable for both the EAI and B2B integration projects.

An integration solution following the BPIOAI approach typically consists of five essential parts [11]:

1. A graphic tool to create and define the execution logic of a business process model.

2. A business process engine which controls the execution of the business processes.

3. A user interface for monitoring and controlling the execution of the business processes.

4. An interface through which the business processes communicate with the business process engine.

5. An integration technology which the business process engine uses to communicate with the systems involved in the integration.

To implement the previously mentioned BPIOAI parts, an integration server is used as a middleware. The integration server is a separate application which acts as an information broker between other applications, systems and databases. The integration server manages the integration by following the integration rules which define the information transforms, schema conversions and controlling and routing the information flows. The integration rules may be defined for example as business process diagrams.

The communications between the business process engine and the remote systems can be handled through adapters. An adapter provides a convenient way to hide the complexity of the underlying communications which may be required in order to exchange data with an application or a system. The adapter is a software module, which contains all the necessary functionalities to exchange information with another system. The adapter provides the information through an interface which is understandable by the integration server. As the integration server may be connected to several different systems, it may also employ several adapters.

A simplest kind of an adapter only converts the underlying interface to another interface, supported by the integration server. This kind of an adapter is called a thin adapter. A thick adapter, on the contrary, is an adapter that implements more complex functionalities, like error handling, information transformations or other data processing tasks.

3.2 Application integration in the production operations management

Application integration in the production operations management aims for a seamlessly integrated operations and information flow from the enterprise business-oriented system through the plant manufacturing executions system to the actual process control system. With fast information flow through this chain, the manufacturing company can be adaptive; react quickly to the market changes with shorter order-production-delivery processes and carry out more accurate resource planning. Since the globalization has lengthened the supply chains of the manufacturing companies, more and more transparency is required both between and within the system hierarchy levels to enable the fluent information flow. With an effective integration of the information systems, the companies can increase their flexibility, responsiveness and cost-efficiency. [1, 5]

The ISA-95 functional hierarchy model levels are often implemented by software systems from different vendors which often means a wide scale of different interfaces, protocols and proprietary software involved in the integration. This makes integrating these systems a complex task. The production operations management activity model in the part 3 of the ISA-95 standard can be used as a guideline for integrating MES activities to lower-level systems like a PCS.

3.2.1 Related research

The integration of the information systems of manufacturing with automation systems has been studied in several research projects. The utilized technologies have been similar to the ones in this study. However, a solution with exactly the same combination has not been reported. In a paper by Parapar [12], a design and implementation of a batch scheduling and material reporting interface between an MES and a distributed control system (DCS) is proposed. Through the interface three types of information is exchanged: recipe information, production and material consumption reports and batch execution information. The data exchange between the ERP, MES and PCS systems are implemented with structured query language (SQL) database tables. The control application follows the ISA-88 part 1 definitions.

The Integrated Operations in the High North (IOHN) [13], a joint industry project of the POSC Caesar Association is studying the integration of the operations and maintenance activities using open standards in the petrochemical industry. The study presents interoperability scenarios, where also an integration between the MES and the DCS is presented. They suggest that the data exchange between the MES and the DCS would consist of ISA-95-compliant detailed production schedule and performance information. The information format would be XML following the B2MML schemas. However, the technology behind the MES-DCS data exchange has not been defined.

Several research projects have concentrated on applying the SOA model to the integration of information systems and automation. For example, Karnouskos et al. [14] propose integration of the enterprise business processes with the shop-floor activities with web services provided by SOA-ready embedded devices. The design is motivated by the dependency of the ERP systems on real-time device information and a media break between the ERP and MES systems. The idea behind the proposed design is to have a distributed business-to-shop floor integration rather than establishing a central manufacturing execution system. The paper does not define the communication technology, but mentions the OPC UA technology and the B2MML data structures as suitable techniques. De Souza et al. [15] have presented SOCRADES, an integration architecture that uses the Device Protocol for Web Services (DPWS) for connecting to the services provided by shop-floor devices. The work also suggests that OPC UA could be used in conjunction with DPWS. However, they do not specify how this combination would be designed.

In a few research projects the role of OPC UA as an integration mechanism for manufacturing operations management has been addressed. However, the results of these projects have been rather preliminary. Stopper and Katalinic [16] discuss the features and consequential benefits of the OPC UA technology. The OPC UA is presented as an important new technology bringing a great deal of new possibilities to the communication of industrial applications. Also using the OPC UA technology in

MES, human-machine interface (HMI) and supervisory control and data acquisition (SCADA) system integration is speculated. A study by Schleipen [17] presents OPC UA as a mechanism for the automation of the configuration of control systems and an applicable communication mechanism for the automated engineering. The study also demonstrates an implementation where standardized computer-aided engineering exchange (CAEX) information is exchanged using OPC UA. The study proposes that the engineering data should be integrated in the address space of the OPC UA servers of the control systems as standardized information models.

3.3 OPC Unified Architecture

The concept of OPC, coming from ”open connectivity via open standards” generally means a data access interface which was specified by the OPC Foundation and released in 1996. It enables the client-server data exchange through a specified interface. The OPC data access (OPC DA) technology is strongly dependent on Microsoft's component object model (COM) and distributed component object model (DCOM) technologies as Microsoft was one of the most important members of the OPC Foundation. Since its release, the OPC data access interface has become popular and nowadays it is a de facto standard in the automation engineering. Many extensions have been released to extend the original OPC specification, for example OPC for alarms and events (OPC AE), OPC for historical data access (OPC HDA) and OPC Batch for batch process data models. The classic OPC has a few disadvantages. Being dependent on COM/DCOM, it is also dependent on the Microsoft operating systems. Also, remote connections across networks or through firewalls using DCOM are difficult to configure. Having these drawbacks, the classic OPC is not applicable for modern application integration projects. OPC XML-DA is an extension which replaces the COM/DCOM techniques with the web service technology and the simple object access protocol (SOAP) communication. This extension is no longer Microsoft-dependent, but it still has not become very common. [18]

The newly released, the next generation OPC standard, the OPC Unified Architecture (OPC UA) specification is intended to group all the previous OPC specifications to one universal specification. The specification consists of 13 parts of which the parts 12 and 13 are to be released in 2010. The OPC UA technology is based on the service-oriented architecture (SOA) and platform-independent technologies so implementing cross-platform communications is simple. By using common web technologies OPC UA is independent of the network topology and firewalls. OPC UA servers and clients can be implemented for any platform, regardless of the operating systems. Even embedded systems can employ the OPC UA technology. The OPC UA specification also takes the security issues seriously, and with modern WS-* specifications about authentication and encryption methods it is secure to be used even across public networks. [19]

The transport layer of OPC UA is based on common web technologies. An OPC UA connection can employ the binary transmission control protocol (TCP) or the hypertext transfer protocol (HTTP) with web services using human-readable XML-format SOAP messages. The binary protocol provides better performance compared to HTTP/SOAP and it requires the minimum processing resources thus it can be used with performance-critical, embedded systems. Being text-based, the HTTP/SOAP protocol requires more processing, and it can be used in systems, where the data should be managed in a human-readable format. The OPC UA applications communicate using the SOAP protocol through HTTP and HTTPS ports, which causes no problems with firewalls. [19]

The OPC UA technology introduces a configurable address space in which data can be represented with complex relations using node metadata to form a full mesh network. This means that the information exchanged using OPC UA can easily follow standard information models, for example the ISA-88/95 data structures. The OPC UA address space consists of nodes, which represent objects, variables, methods and types. An object node structure corresponds to an object in the object-oriented programming (OOP) paradigm. It can contain attributes, other objects, variables with readable data values and methods which can be remotely called to execute functionalities. It can also raise events to transmit data asynchronously. [19]

An overview of an object node structure is visualized in the figure 4. It shows an example of an OPC UA object and its references in the address space. The object is named ControlValve and is of type ControlValveType which again is a subtype of the ValveType type. The object contains variables Setpoint and Value. It also contains another object, Configuration, which contains variables OpenLimit and CloseLimit. The ControlValve object also contains a method, SetTarget. The object can send notifications of data modifications and events.

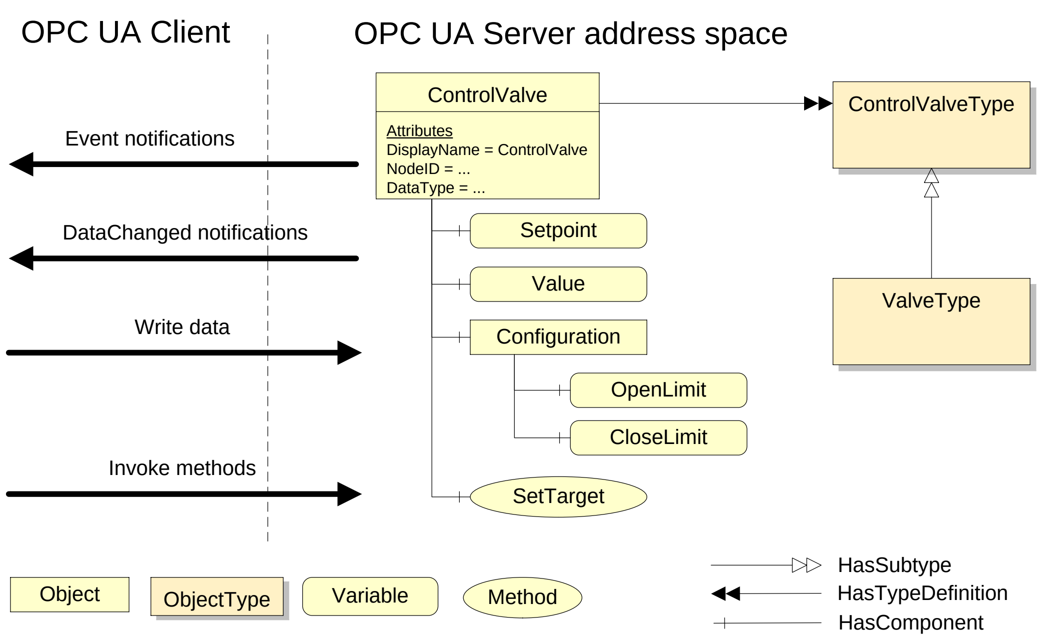

Figure 4: An example of an OPC UA object node [19]

The service-oriented architecture in OPC UA means that the functionalities (for example Read, Call, Subscribe etc.) of an OPC UA server are defined as services. These services are grouped to 10 service sets depending on their purpose. The service sets are the following: [20]

- **Discovery service set** contains the services that allow OPC UA clients to discover endpoints provided by an OPC UA server.
- **SecureChannel service set** defines the services that allow clients to establish a secure communication channel with a server. In the secure communication, the confidentiality and integrity of the exchanged messages is ensured.
- **Session service set** contains the services that clients use to authenticate users and manage sessions.
- **NodeManagement service set** consists of the services that allow clients to add, edit and delete nodes in the address space of an OPC UA server.
- **View service set** defines the services that clients use to browse the address space of an OPC UA server.
- **Query service set** consists of the services that allow clients to read a subset of data from the address space of an OPC UA server.
- **Attribute service set** contains the services that clients use to read and write node attributes and variable values.
- **Method service set** defines the services that allow clients to invoke methods in the address space of an OPC UA server.
- **Subscription service set** consists of the services that allow clients to create, modify and delete subscriptions. Subscriptions are needed in order to receive data changed and event notifications from the server.
- **MonitoredItem service set** defines the services that clients use to create, modify and delete monitored items in the address space of a server. The monitored items are used to monitor attributes for data changes and objects for events.

The application developers can implement OPC UA software using the OPC UA communication stacks through their application programming interfaces (APIs). Software development kits containing the communication stack and an API already exist for .NET, Java and ANSI C/C++ developers. [18]

3.4 Standards

In the scope of this work, two standards are essential for the production operations management integration: the ISA-88 batch control standard and the ISA-95 enterprise-control system integration standard. These standards and their implementations are discussed in this section.

The domains of these standards are presented in the figure 5. The ISA-95 standard focuses on the communications between the enterprise and manufacturing control levels but also to defining the functionalities in the level 3. The ISA-88 is focused on the process control level, but can also be used in communications between the levels 2 and 3.

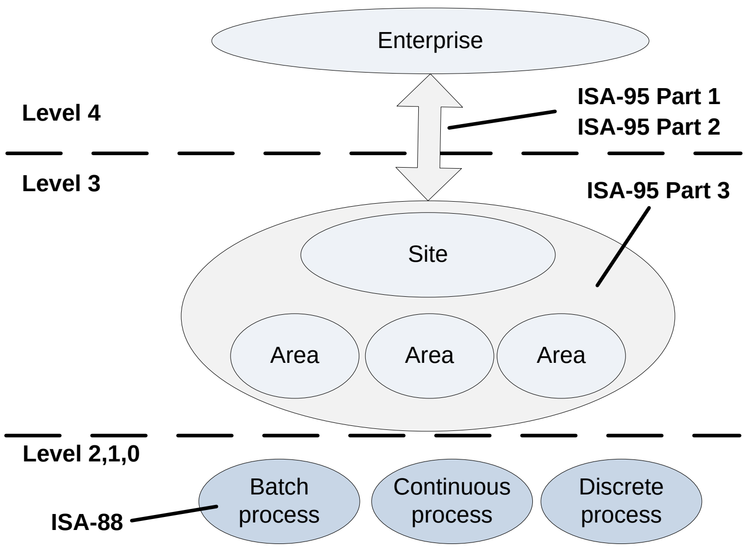

Figure 5: The domains of the essential parts of the standards

3.4.1 ISA-88 Batch control standard

Prior to the ISA-88 standard, a few basic problems made designing and implementing batch process systems difficult. No universal models about the batch process equipment nor the control existed so the integration of systems from different system providers was difficult. Also communicating about the batch process requirements and configuring the batch process solutions was difficult as no common terminology or models existed.

The ISA-88 standard set is intended to provide solutions to the previously mentioned problems. It is designed to be used with batch process systems, but it is also widely used with discrete, continuous, hybrid and storage process systems [21]. The batch process systems designed and implemented following the ISA-88 standard provide many benefits, like reduced batch cycle times, reduced recipe modification times, increased batch run rates, reduced raw material loss and better data availability for analysis functions [22].

The ISA-88 standard is also known as the international standard IEC 61512. The standard consists of five parts of which the part 5 is still under development. The parts are explained briefly below. [23]

- **ISA-88.01-1995, Batch Control Part 1: Models and Terminology**

  The first part defines models and terminology which are intended to represent a good practice for the design and operation of batch process plants and to be used to define the control requirements of a batch process. The models and the terminology are defined in an abstract level so they can be applied regardless of the level of automation. [24]

- **ANSI/ISA-88.00.02-2001, Batch Control Part 2: Data Structures and Guidelines for Languages**

  The part 2 of the standard defines data structures for describing the batch process control (recipes) and for sharing the batch information across the batch control system implementations. It also defines a symbolic language guideline for recipe depiction. The definitions are supposed to help in designing, implementing and operating batch control systems. [25]

- **ANSI/ISA-88.00.03-2003, Batch Control Part 3: General and Site Recipe Models and Representation**

  The part 3 of the standard provides models for the general and site recipes. It defines their usage within a company and across companies. Also the data structures and the representation of these recipes are specified. [23]

- **ANSI/ISA-88.00.04-2006, Batch Control Part 4: Batch Production Records**

  The part 4 defines the batch production record data structures which are used for storing the batch information. These definitions are important for batch information retrieval, analysis and reporting. [23]

- **Part 5: Implementation Models & Terminology for Modular Equipment Control**

  The fifth part of the standard is intended to expand the part 1 definitions and introduce models and terminology for a modular and distributed design of equipment control. It defines the models to be used in defining and implementing control strategies that execute process task strategies in the equipment and control modules. Meaning that the goal is to create specifications which automation component vendors can use to develop modular automation components consistent with the ISA-88 part 1 models. This part of the standard still remains unreleased. [26]

The ISA-88 standard introduces the definition of a batch process accordingly: "a batch process leads to the production of finite quantities of material (batches) by subjecting quantities of input materials to a defined order of processing actions using one or more pieces of equipment." The batch process is a special kind of a process; it is neither continuous nor discrete, but is has characteristics of both.

In the scope of this work, the parts 1, 2 and 4 play the most significant role as they define the equipment and recipe hierarchies, the recipe and production record data structures and the control activity model. These definitions are explained below.

The part 1 of the standard defines the physical model which can be used as a basis when designing a batch process system. It is presented in the figure 6. The enterprise level correlates to a manufacturing company, the site level to a manufacturing plant of the company and the area level to a section of the manufacturing site. These three up-most levels are considered highly business-oriented and are not defined more specifically in the scope of the standard. [24]

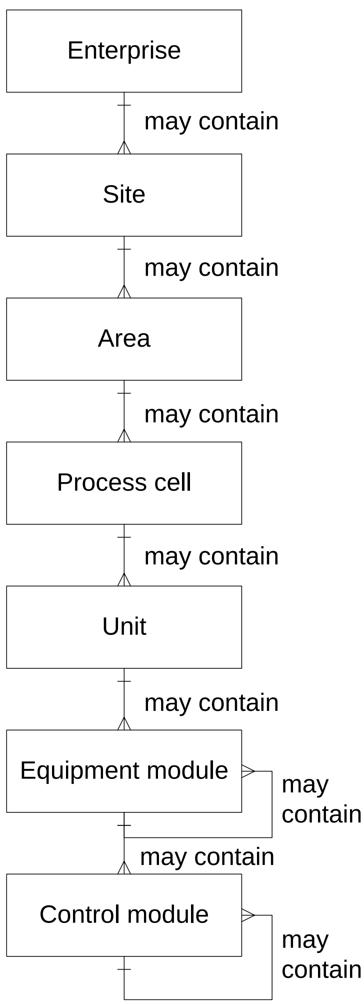

Figure 6: The ISA-88 physical model [24]

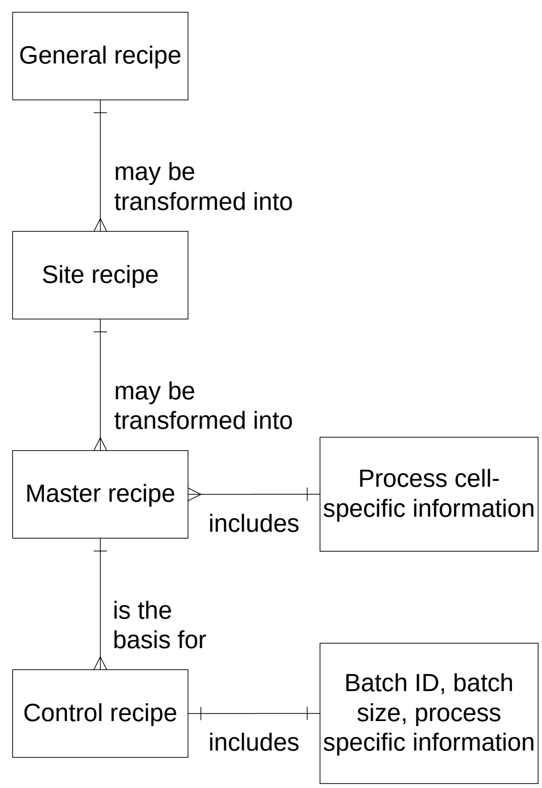

Figure 7: The ISA-88 recipe hierarchy [24]

The four lower levels of the physical model are used to describe the hierarchy of the actual batch process equipment. A process cell contains all the lower level modules (units, equipment and control modules) necessary to produce a batch. A unit is a module that is able to perform a major processing activity to a batch. An equipment module is group of control modules which perform minor activities in the batch process. A control module consists of automation components like sensors and actuators and the equipment needed to control the module. [24]

The part 1 [24] defines a recipe as ”an entity that contains the minimum set of information that uniquely defines the manufacturing requirements for a specific product”. The part defines the recipe types which are used within the manufacturing company. The recipe types and their hierarchy are shown in the figure 7. A general recipe is the most abstract description of a product. It is intended to be used in the business planning in the enterprise level and as a basis for more specific recipes. It defines the input materials, their relative quantities and the processing activities but without any site-specific information. A site recipe is derived from a general recipe and contains site-specific information, like information of local materials, equipment and other resources. A master recipe is specific to a process cell and it contains information of the equipment involved in the production of a batch. Master recipes are required in an ISA-88-compliant batch process system as they are the templates from which control recipes are derived. A control recipe is derived from a master recipe and supplemented with the scheduling and operational information which makes it specific to a single batch. A control recipe contains all the information required to control the batch process equipment to produce a batch. Using ISA-88 definitions to design the structure of the batch process control procedures and recipes, the production rules described in the recipes can be fully separated from the batch process equipment. This makes the product definition management flexible to recipe changes [22].

The control activity model defined by the part 1 of the standard is shown in the figure 8. It shows the functions that are required to implement a batch production. It contains functions to initialize a batch process, execute processes and to report batch information. The parts 1 and 2 of the ISA-88 standard are focused on the functions of process management, unit supervision and process control. From the later parts, the part 3 is focused on the recipe management and the part 4 on the production information management.

The recipe management activity manages the general, site and master recipes and outputs a master recipe, which is converted to a control recipe in the process management activity. The production information management collects, stores, processes and reports the batch production information using the batch production record data structures defined in the part 4. The process management activity is a collection of functions managing the resources and the execution of batches within a process cell. It creates the batch-specific control recipes, controls the execution of the batch process and collects the batch process information. The unit supervision activity is responsible for managing the unit resources and executing the procedures. The process control activity contains the basic process control functions possibly distributed to equipment and control modules. The lowest level of the process control is represented by the personnel and environmental protection activity which contains

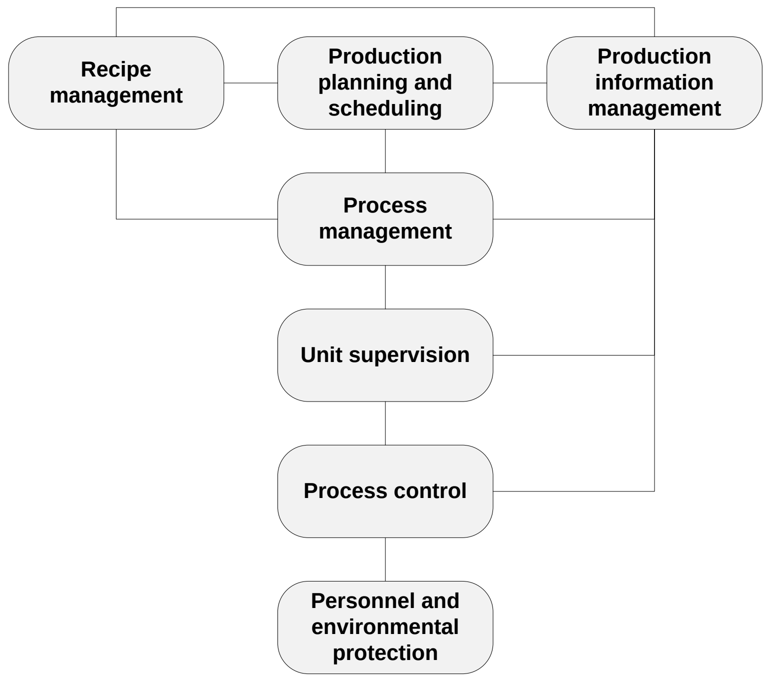

Figure 8: The ISA-88 control activity model [24]

functions to ensure that the process will be secure regardless of the higher level control. [24, 21]

3.4.2 ISA-95 Enterprise-control system integration standard

In the past decades, the manufacturing companies have been investing massively to develop enterprise level information systems, ERPs, to improve their business functions [6]. These functions are although largely based on the information of the resources involved in the manufacturing. This information originates from the process control level thus to achieve efficient and reliable ERP functionalities, the enterprise level must be closely connected to the process control level. But since the enterprise level systems are business-oriented and the process control systems are engineering-oriented, there is a wide cultural gap between these two levels. Also the large amount of information, non-uniform data structures and protocols cause problems in the integration.

The ANSI/ISA-95 Enterprise-Control System Integration standard was developed to close the gap [27]. It provides solutions for simplifying the integration difficulties between the enterprise and process control levels not only by dividing the functionalities of the level 3 to groups with clear boundaries and responsibilities, but also by defining the terminology and the contents of the exchanged information allowing the different systems to communicate using a standard language. The ISA-95 is also known as the international standard IEC 62264. The standard aims to increase the interoperability and integration possibilities while reducing the risk, costs and errors related to the business-to-manufacturing integration [4].

The ISA-95 standard consists of six parts of which the parts 4 and 6 still remain unreleased [28]. The contents of the parts are explained briefly below.

- **ANSI/ISA-95.00.01-2000, Enterprise-Control System Integration, Part 1: Models and Terminology**

  The first part defines the hierarchy and the relevant functions of systems in the enterprise and process control domains. It also defines the terminology and the information objects typically exchanged between the systems in an abstract level. It is supposed to be a basis for designing and implementing manufacturing facilities, system interfaces and system integration solutions. [3]

- **ANSI/ISA-95.00.02-2001, Enterprise-Control System Integration, Part 2: Object Model Attributes**

  This part focuses on explaining the data exchange interface contents between the enterprise and manufacturing level systems in detail. The purpose of the part 2 is to minimize the effort, costs and error possibilities when implementing interfaces. Its guidelines about system integration and interoperability should also be considered when designing new systems. [29]

- **ANSI/ISA-95.00.03-2005, Enterprise-Control System Integration, Part 3: Activity Models of Manufacturing**

  The part 3 takes the level 3 functions into its scope and defines the terminology, functions and models characteristic to the manufacturing operations management in the level 3. It divides the manufacturing operations management into four groups: production, maintenance, quality and inventory operations management. Activity models which define the functions and the involved data are defined for each of these parts of manufacturing operations management. The part 3 is essential for designing and implementing successful business-to-manufacturing integration. [4]

- **Part 4: Object models and attributes of manufacturing operations management activities**

  The part 4 aims to defining a common object model and attributes which could be used for expanding the information exchange between the enterprise and process control levels to cover not only the production information but also other types of data. It defines the information shared between the manufacturing activities which then helps defining the contents of the interfaces between the level 3 and 4 systems. This part of the standard is still unreleased. [30]

- **ANSI/ISA-95.00.05-2007, Enterprise-Control System Integration, Part 5: Business-to-Manufacturing Transactions**

  The fifth part of the standard defines the data transactions between the level 3 and 4 systems. It defines the transaction execution and the structures of the exchanged information. The verb and noun parts of exchanged messages are defined in compliance to the object models defined in the parts 1 and 2 of this standard. This part is intended to be a guideline for designing and implementing system integration between the level 3 and 4 systems. [31]

- **Part 6: Manufacturing operations transactions**

  The part 6 intends to define the transactions of information within the activities of manufacturing execution systems [32]. This part of the standard is still unreleased.

The main focus of the ISA-95 standard is to define the manufacturing operations and their integration between the enterprise and process control domains, meaning the enterprise resource planning and manufacturing execution systems. The parts of the standard are each focused to different parts of the integration as explained above. In the scope of this work, the parts 1 and 3 play the most significant roles. The part 1 is important because of the definitions of the system hierarchy and the manufacturing level functions. The part 3 is significant as it defines the activities and the related data within the level 3.

3.4.3 Applying the standards

Applying the standards in practice means typically considering their viewpoints when designing and implementing the systems. The hierarchy and activity models defined in ISA-88/95 can be used as a basis for system design. In addition to the abstract models, the World Batch Forum (WBF) organization has developed schemas which represent ISA-88/95-compliant data structure implementations in XML format [33]. The schemas are written in XML schema language (XSD) defined by the World Wide Web Consortium (W3C).

B2MML The schema that implements the ISA-95 models is called the business to manufacturing markup language, B2MML. It is a complete implementation of the ISA-95 standard. It defines the data structures that are meant to be a common format in the data exchange between the enterprise resource planning, supply chain management and the manufacturing execution systems.

BatchML The batch markup language (BatchML) is an implementation of the ISA-88 standard written in XML schema language. It contains data structures for representing batch, recipe and equipment information. However, it still lacks the support for the batch production record data structures which are defined in the part 4 of the ISA-88 standard.

The BatchML schema set defines the BatchInformation data structure which contains the information managed by a batch process system. Among other information, it contains control recipes, recipe building blocks and equipment elements which are used to represent the equipment- and batch-specific production rules. It also contains batch list data structures which are used to represent the work orders and their schedule. The recipe data structure follows the recipe contents defined in the part 1 of the ISA-88 standard (header - formula - equipment requirements - recipe procedure - other information). The more detailed structures of the recipes are defined in the part 2. The batch list structure is based on the ISA-88 standard, but it is not a full implementation of the batch schedule data structure [34]. The batch information and control recipe data structures according to the BatchML schemas are visualized in the figure 9.

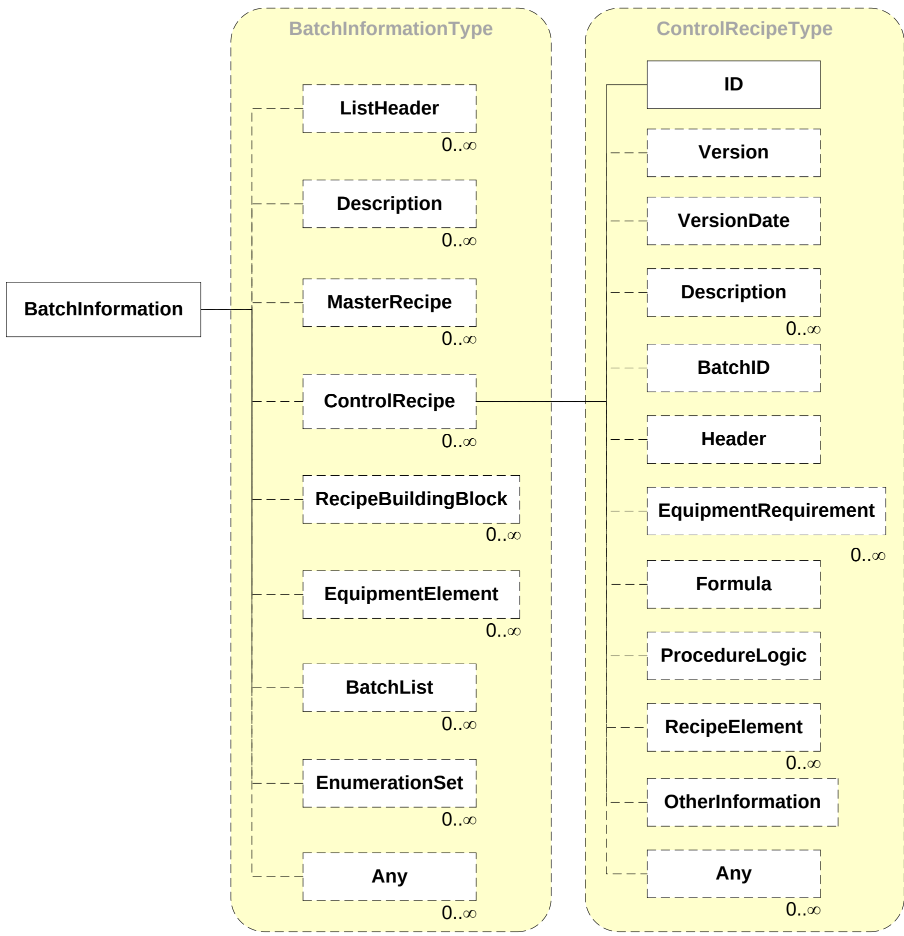

Figure 9: The BatchInformation data structure [34]

In the latest release (V0401) of the previously mentioned schema sets, the BatchML schemas has been included to the B2MML namespace and they both use the same schema defining the common components. Fusing these schema sets together is intended to simplify the standard implementations in the integration processes. [33]

4 Integration research platform

4.1 Overview of the research platform

The integration research platform is a manufacturing company-like environment which consists of several information systems. This so called ”testbed” is going to be used as a platform for integration and manufacturing information system-related research. It is also supposed to be used as an experimental environment for educational purposes and for hands-on project platform for students.

The testbed consists of several different information systems, as illustrated in the figure 10. It contains an enterprise resource planning system (ERP), a manufacturing execution system (MES), an automated process control system (PCS) and a connection to the batch process equipment. It also contains a maintenance system implementation based on a condition monitoring (CM) system and an enterprise asset management system (EAM).

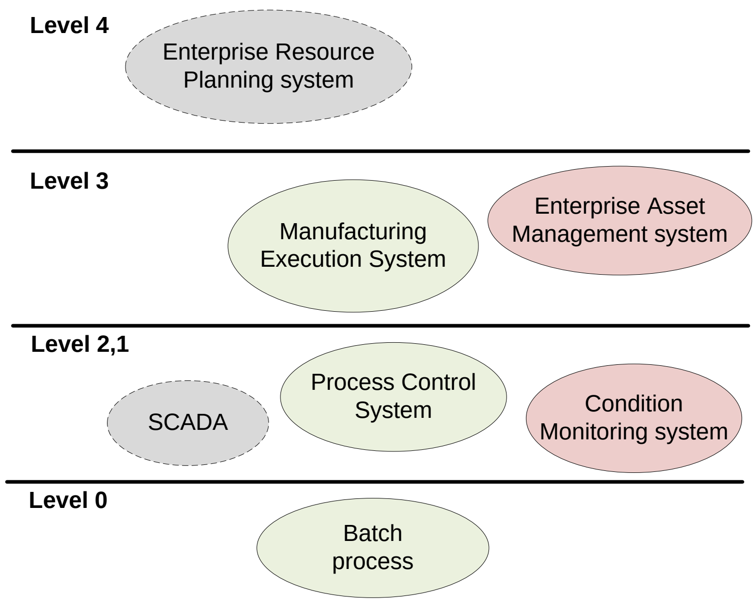

Figure 10: The system hierarchy of the research environment

Most of the presented system environment will be implemented as a result of two master's theses. In the scope of this thesis the parts of the system environment which are involved in the production operations management are implemented. The other thesis is focused on the integration of maintenance information thus the parts relevant to the maintenance operations management are implemented. However, some parts of the system environment are to be implemented in the future work. The components of the system environment are explained briefly below.

Process: The actual process on the level 0 is a mini pulp batch process system which is located in the laboratory facilities. It is previously implemented as a result of a master's thesis to simulate a pulp batch production process [35]. It contains all the necessary equipment to run through the phases of a pulp process.

PCS: The pulp process is controlled by a process control system (PCS). It is designed for batch process control and is ISA-88-compatible. It receives production orders, maintains the short-term execution schedule and controls the process using batch-specific process instructions, control recipes.

SCADA: The supervisory control and data acquisition (SCADA) system is used for monitoring the process execution in the process facilities. It collects information from the process control system and visualizes the real-time statuses of the process equipment and the stage of the process itself in a user interface. With the SCADA system the process operators can perform supervisory actions and react to unexpected events, for example alarms. The SCADA system will not be implemented in this stage of the research project.

CM: The condition monitoring (CM) system is used to read and analyze the diagnostic information of smart actuators in the process equipment. This software manages alarms and other maintenance-related information which originate from the actuators. The information analysis in the CM is crucial for implementing a preventive maintenance system.

MES: The manufacturing execution system (MES) provides tools for the process administrators to manage the manufacturing processes. With the MES, the process execution is monitored and the production and quality reports which are based on the information collected from the process control system, are managed. The production is optimized and scheduled in high detail based on the coarse schedule information created in the enterprise level and the resource availability information collected from the process control system and enterprise asset management system. The MES also manages the product definition information (recipes) which describe the production rules for the products available for production.

EAM: Using the enterprise asset management (EAM) system, the maintenance engineers can create preventive maintenance work orders based on the analysis of the equipment diagnostic information collected from the condition monitoring system. This functionality may also be included as a part of an ERP system, but in this system environment it is separated into an EAM information system.

ERP: Among many other more business-oriented tasks in the enterprise level, the ERP system is used to manage the product definition information and to schedule a coarse production plan for a manufacturing site. The system manages the information about material, personnel and equipment resources required for planning the production. In the scope of this research project, the ERP system is not yet implemented in the system environment.

4.2 Production operations integration in the research platform

In the scope of the integration of production operations management, the enterprise resource planning system, the manufacturing execution system and the process control system are the essential parts of the system environment. As discussed in the section 2, the activities involved in the production operations management are defined by the part 3 of the ISA-95 standard. The activities of the production operations management are often implemented in a manufacturing execution system in the level 3. As seen in the production operations management activity model, the production operations are dependent on the information which originates from both the lower process control level and the upper enterprise level systems.

In order to implement a platform for the research of integration of production operations management, the production operations of the manufacturing execution system needs to be integrated to the other systems. However, as this thesis is focused on the integration of a manufacturing execution system with a process control level system, the integration of production operations with the enterprise-level systems is not implemented. The implementation of an ERP system and the integration of production operations of the manufacturing execution system with the ERP are left as an option for the future work.

4.3 Maintenance operations integration in the research platform

The ISA-95 part 3 also defines the activity model for the maintenance operations management. This model in addition to the service-oriented architecture paradigm are the basis for the design and implementation of the integration of the maintenance operations management in the research platform.

The components involved in the maintenance operation management are an enterprise asset management system, CalemEAM, and a condition monitoring systems, Metso FieldCare. The CM system is connected to Neles ND9000P smart valve controllers. Since the laboratory equipment does not contain any smart valves, the connection and the functionality of a smart valve actuator is only simulated using a Profibus PA communications simulator.

CalemEAM is a web-based EAM system which is based on a MySQL relational database. The database contains all the asset information including the identification information, purchasing information and service history. The maintenance operations are based on preventive or corrective maintenance work orders which are scheduled to the maintenance staff. These work orders can be created based on the information collected from the condition monitoring system.

The condition monitoring system Metso FieldCare is used to access the information of Neles ND9000PA valve controllers which have self-diagnostic functionalities for monitoring the state and the health of the actuator. An OPC UA server was implemented for accessing the condition monitoring information which is represented by applying the OPC UA device information model which defines data structures for device configuration and diagnostics. [36, 37]

The implemented integration system communicates with the EAM system using a web service interface and with the CM system using the OPC UA technology. The maintenance operations integration is evaluated with a selected set of maintenance operations management use cases. [36]

5 Requirements of application integration

In order to reduce the production costs by shortening the product time-to-market, minimize the production risks and for the whole manufacturing company to work more efficiently, the information flow between the different information systems involved in the production chain must be fluent. The data needs to be transparent through the system level hierarchy and be available for all the manufacturing process participants which are dependent on the information.

The production operations management activity model of the ISA-95 presented in the figure 3 defines the production operations activity groups, their relations and the exchanged data in an abstract level. In the scope of this work, the focus is on the activity groups that are responsible for the data exchange between the levels 2 and 3. These three groups are the product definition management, production execution management and production information collection. The activities and the meaning of their efficient integration to the business of a manufacturing company are explained below.

By implementing extensive product definition management functionalities able to handle the growing number of product variants, it is possible to reduce the product time-to-market delay. This makes the manufacturing company adaptive to the changes in the market and customer habits. This requires that the product definition functionalities can share the product definition data seamlessly with the ERP and the process control system.

With the detailed scheduling activity group functions the manufacturing company can optimize the work order execution according to the resource availability and many other production-related variables. However, in addition to the efficient product definition management, this also requires real-time data exchange communications with the process control system. The production execution management activities are responsible for keeping the dispatch list in touch and delivering operational commands to the process control system.

One of the most important values for a modern manufacturing company is the quality of the products. Controlling the quality factors during the process execution requires an extensive production data collection functionalities in the MES level. These activities collect all production-related data and archive it in the data storage. This data is also important for the product tracing and tracking functionalities. By monitoring and analyzing the quality indicators, the production process can be adjusted and optimized to produce higher quality products. The high quality is important value not only to the end products but also to the manufacturing process itself. The process can be optimized by using calculated quality indicators, for example the resource utilization, product quality and production times.

5.1 Use cases and their requirements

The ISA-95 production operations management activity model presented in the figure 3 shows that the data exchange between the levels 2 and 3 consists of four types of information: product definition, operational commands and responses and equipment and process specific information. In order for the testbed platform to successfully fulfill the requirements set by the MES-level activities, all these data exchange scenarios must be implemented in the application integration. Therefore, it is logical to define three use cases in compliance with the three MES-level activities that require direct data exchange with the level 2 systems. The use cases are:

1. Integration of product definition management
2. Integration of production execution management
3. Integration of production data collection

The use cases relevant to the integration of production operations management are presented in the figure 11 and explained below.

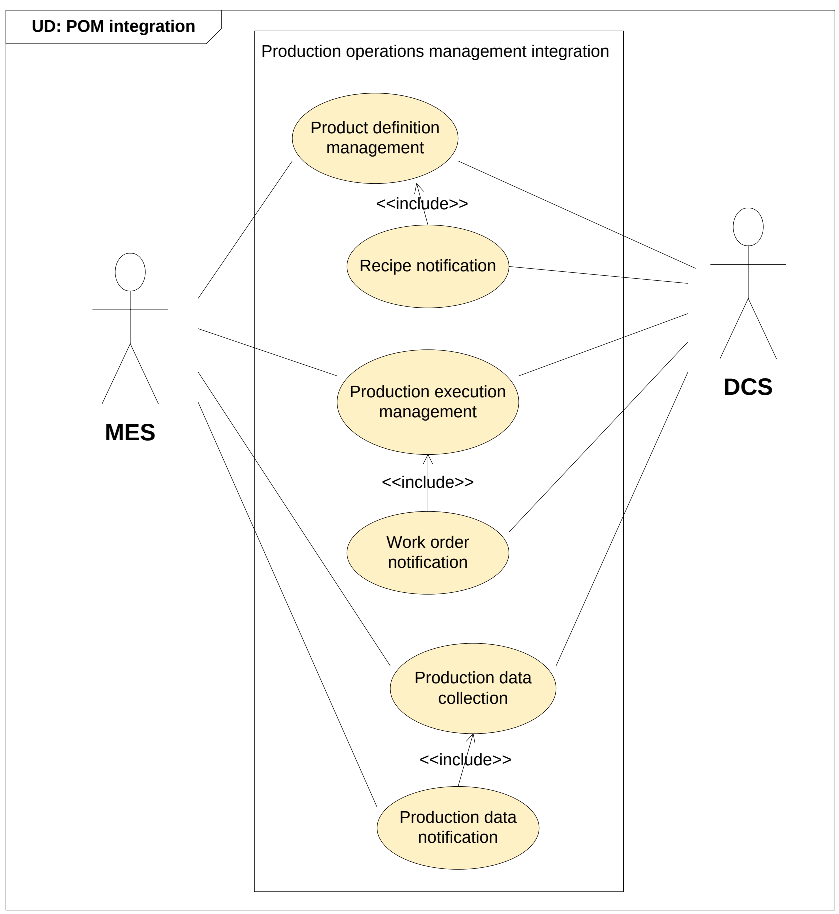

Figure 11: The use case diagram of the production operations integration

These use cases represent the requirements for the integration of the production operations management between a manufacturing execution system and a process control system. They contain the data exchange functionalities which are required by the MES- and PCS-level activity models. As also the other activities within the production operations management are dependent on the information integrated in these use cases, they all three are important for efficient production operations management of a manufacturing company.

5.1.1 Integration of product definition management

Product definitions mean the production rules that are specific to the production process and on the lowest level even to the process equipment and the batch itself. With a batch process system, product definitions are represented as recipes. The product definition management activity in the MES level handles master recipes which are specific to the process facilities. Master recipes are based on the site recipes, which are managed in the enterprise-level systems.

Once a batch is scheduled in the detailed production scheduling activity and a work order needs to be dispatched to production, a batch-specific control recipe is created and a notification is sent to the level 2 process control system which then retrieves the recipe information. The control recipe contains the equipment-specific production rules and information about all the required resources that will be used to produce a single batch. When creating the control recipe, the resource availability will be taken into account.

The process control system must be able to manage the control recipes and use them when executing the production process. The control recipes can contain even the simplest details about the process, but repeating all these minor details in every control recipe is not appropriate to the efficient information managing. This is why the process control system has an equipment-specific recipe element library where the common process actions, for example heating or controlling material movements in the pipes, are already defined. These actions are managed by the site engineers. The procedure logic defined in the control recipe then only needs to contain references to the predefined recipe elements in the library.

5.1.2 Integration of production execution management

The production dispatching activity manages the batch process execution and provides work orders for the process control systems according to the detailed production schedule information. It also monitors the statuses of the work orders. The work orders are defined in a dispatch list which is made available to the production execution management activity. As new work orders are added, the PCS is notified with an asynchronous event and the work order information is synchronized between the systems.

The production execution management activity shares the dispatch list information with the level 2 process control system. It can also send operational commands to the process control system, for example to directly control the production by initiating level 2 activities as defined in the dispatch list elements. The operating command interface should be implemented with method-like functionalities.

The production execution management also receives information from the process control system. This information is typically carried by asynchronous production events and contains data concerning the production execution, for example production phase changes, production start-up or completion notifications.

5.1.3 Integration of production data collection

The production data collection activity gathers and manages the batch-specific production data. This information can be delivered with asynchronous events or synchronous readings using method-like functionalities to read data from the process control system. The information content itself may be sensor readings, equipment states, quality values, production process alarms, events or other information which is of any importance to tracking or tracing the batches. The information is stored to a data structure, which contains all the production information concerning a single batch. This information is archived and provided for other activities or operators as production reports.

5.2 Requirements for the information contents

The information exchanged between the PCS and MES levels is not only shared between the activity groups of the production operations management, but also exchanged further with the enterprise level systems. As the activity groups and systems may be separate applications from different system providers, the information contents need to be in an understandable format. This means that the information models need to follow known industry standards.

The information models in the process control system comply with the ISA-88 batch process standard. The information in the manufacturing execution system and higher follow the ISA-95 standard and its data structures. The ISA-88 is specific to batch process systems and the ISA-95 is a generalized to be independent of the process type. As the ISA-88 is used in the process control system and ISA-95 is applied in the MES and ERP levels and they are partly overlapping, these two standards need to be used consistently in the application integration scenario.

5.3 Requirements for the integration technology

The data exchange interfaces between the manufacturing execution system and the process control system must be able to provide an extensive support for application integration. It must be able to provide simple read and write functions for exchanging information objects. The information is represented using defined information models, for example ISA-88/95-based data structures. In addition to the simple data integration, the interface must also support invoking functionalities remotely by calling methods provided by an information system. Also transmitting information with asynchronous events must be possible.

The integration technique must be suitable to handle the simple data transactions but also integrating the method calls and asynchronous events. Since the MES and PCS level operations must be closely integrated, the integration should not cause unnecessary delays; the integration operations should be executed as close to real-time as possible. Since also other systems in different hierarchy levels may need to be integrated to the MES or PCS, the integration technique should make both vertical and horizontal integration possible.

The integration technique needs to contain all the required integration rules to manage the information flows between the systems. As the different systems typically use different kinds of protocols and data structures, the integration technique should be able to perform data transforms for the integrated information where necessary.

6 Design of the application integration

The system environment in the scope of this thesis contains a manufacturing information system which needs to be integrated with a process control system in order to increase the efficiency of the manufacturing processes. The application integration needs to comply with the requirements described in the previous section. The design proposed in this section makes the system integration flexible to changes, feasible for both vertical and horizontal integration and compliant with the present and future technology standards.

6.1 Overview of the design

The application integration is implemented with a middleware application, an integration server. It follows the business process integration-oriented application integration approach which means that the actual integration logic is included in business process descriptions. The communication interfaces of the middleware application are implemented using an adapter technology for the integration platform to be independent of the varying communication protocols.

The use of a business process integration-oriented application integration with an integration server allows both vertical and horizontal integration scenarios. It also enables complex data transactions, invoking remote service and transmitting asynchronous events thus it is a suitable technique for integration of the production operations management between an MES and a PCS.

An overview of the designed application integration environment is visualized in the figure 12. It presents the information systems and other software modules involved in the integration scenario and also the system hierarchy according to the ISA-95 standard.

The presented application integration environment contains a manufacturing execution system and a batch process control system. The manufacturing execution system activities are defined by the ISA-95 standard and the batch process controller activities by the ISA-88 standard. As seen in the figure 12, these activity definitions are very much alike and even partly overlapping. These standards provide good guidelines for the integration solution presented in this work. The systems are integrated using an integration server as a middleware and the OPC UA technology for communications. The details of the application integration design are explained in the following subsections.

6.1.1 Data exchange interfaces

To enable the exchange of the information, events and commands throughout the whole system chain, the data exchange interfaces are implemented with the OPC UA technology. Using OPC UA, the systems can provide or access process data through a secure interface using configurable information models. An OPC UA interface can

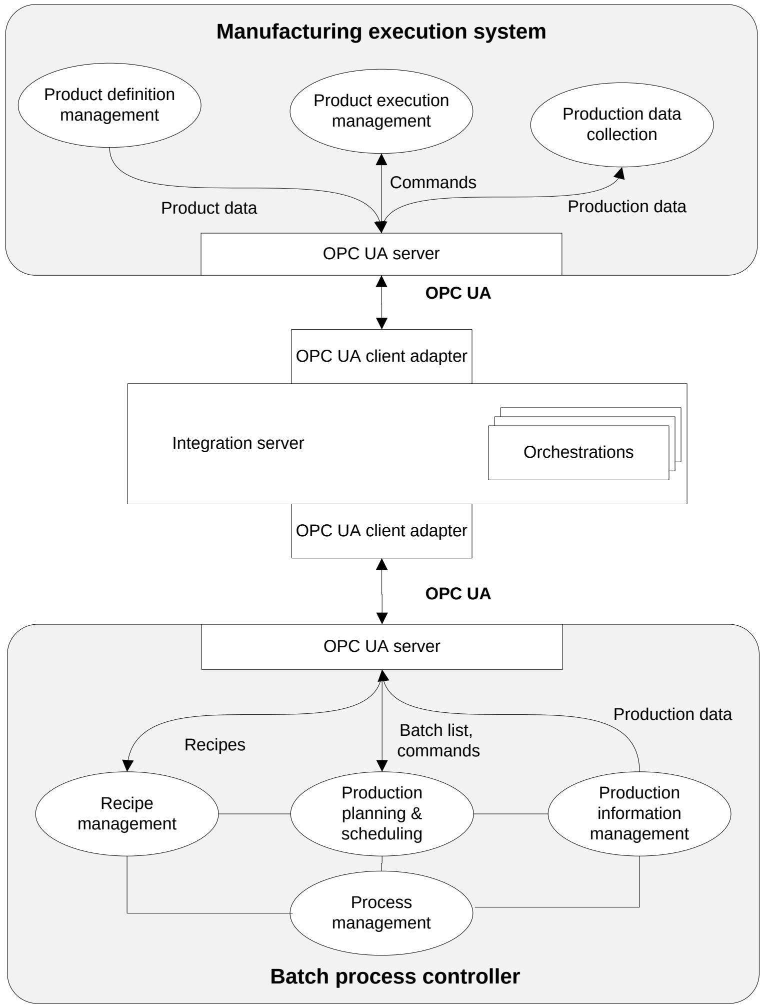

Figure 12: An overview of the designed system environment

also deliver asynchronous events using the publish/subscribe messaging paradigm where the clients subscribe to receive events which the server publishes through an object in the OPC UA address space. Commands from the clients to the server can be implemented as methods.

With both synchronous and asynchronous data exchange possibilities and configurable information models as the most important aspects for this integration scenario, the OPC UA technology provides all the necessary functionalities for an efficient application integration in a manufacturing company.

In this application integration environment design, the information systems are suggested to have OPC UA interfaces. The interfaces can be native parts of the systems or wrapper interfaces built on top of other, possibly proprietary interfaces. As the both MES and PCS systems in this integration case act as service providers with big amounts of information, these systems host OPC UA servers for other systems to connect to. The middleware on the other hand contains the business process rules and acts as a service and information broker, so it hosts OPC UA clients which consume the services provided by the MES and PCS.

The systems hosting OPC UA servers provide address spaces that present the system structures as folder tree hierarchies which contain objects representing meaningful information, for example recipes and batch schedule entries. The connected OPC UA clients can add and edit the information objects dynamically.

6.2 Application integration participants

6.2.1 Batch process control system

The actual production process system is a pulp batch process system controlled by a batch process control system (PCS). The automation system is ISA-88-compatible; the architecture of the batch process control system follows the ISA-88 control activity model thus the three functionalities which communicate with the upper-level system can be identified, as seen in the figure 12. Also the control procedure hierarchy is ISA-88-compliant and the process control system can run the batch production process automatically by following the production rules defined as ISA-88 recipes. The schedule is represented by a production schedule data structure as defined in the part 2 of the ISA-88 standard and the production information is collected and reported using batch production record data structures which are defined in the ISA-88 part 4.

The PCS has a data exchange interface for higher-level systems like an MES or a SCADA. Through this interface, the MES product definition functionalities can add or edit the recipe information managed by the recipe management activity of the batch process control system. The interface is implemented using the OPC UA technology. The PCS has an integrated OPC UA server which provides the relevant process data to the clients using configurable address space information models. In this integration scenario, the information models comply with the data structures defined in the parts 2 and 4 of the ISA-88 standard. The data structures provided by the PCS interface are represented in the figure 13.

The BatchProcess structure is created as the base object in the address space of the OPC UA server of the PCS. The BatchProcess object contains ControlRecipe data structures which define the batch-specific production rules that are used to control the process equipment to produce a batch. The BatchProcess object also contains the batch scheduling information as BatchScheduleEntry data structures which represent the work orders dispatched from the MES level. The work orders contain among other things a unique batch id, requested execution times and have references to the corresponding batch-specific control recipes. The work orders are executed automatically in the process control system on the requested times. The batch process information is collected into BatchProductionRecordEntry data structures which have references to the specific batches. The BatchProductionRecordEntry data structure follows the batch production record structure defined in the part 4 of the ISA-88 standard.

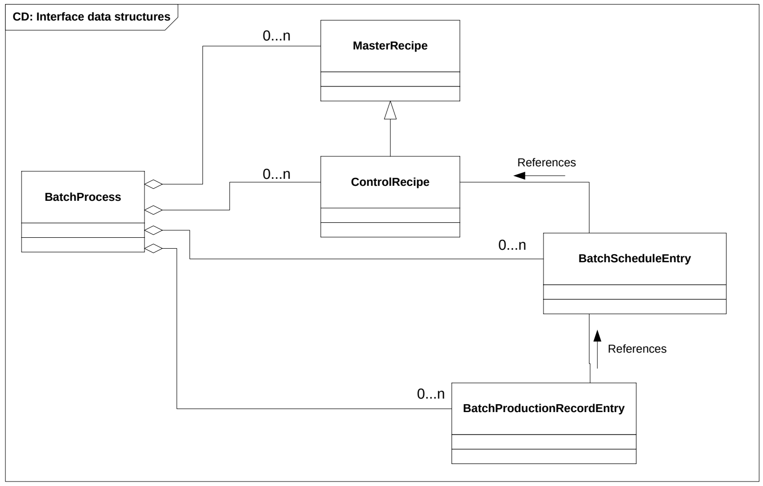

Figure 13: The ISA-88 data structures of the batch process control system represented in a UML class diagram

The data managed in the PCS and exchanged with the MES consists not only of recipe, schedule entry and production record entry data structures but also of production events, alarms and remote method calls. The asynchronous events are for example production started, paused, resumed, completed events but also alarms which inform about unexpected events during the production process. The methods are services provided by the PCS for example for updating the recipes or the schedule information. The initialization, executing and reporting the batch process information operations are presented in the figure 14 as a UML sequence diagram.

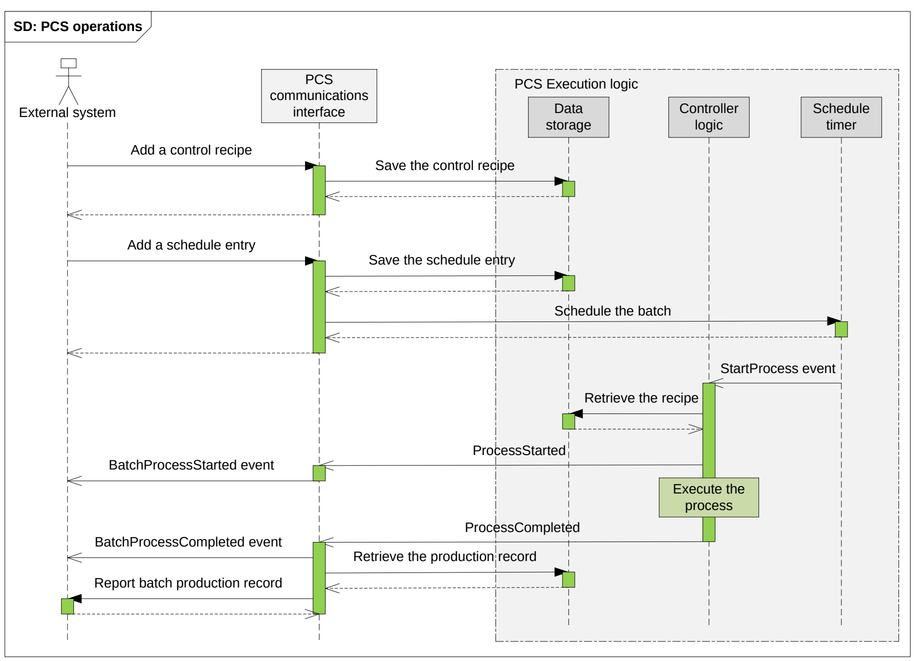

Figure 14: The sequence diagram of the PCS activities

At first, in the initialization stage, the PCS must be provided with a control recipe which contains the production rules for the batch to be produced. Then a work order which references the recipe is created in the MES level and sent to the PCS. In the execution stage, the batch will be produced on the defined time. The PCS raises reporting events when the batch process starts, stops or moves from a phase to another. When the process completes, a corresponding event is raised and the batch production record data structure is sent to the MES.

6.2.2 Manufacturing execution system

The manufacturing execution system complies with the ISA-95 standard and contains the functionalities defined in the production operations management activity model. In this design, however, the functionalities in the product definition management, production data collection and production execution management handle ISA-88-compliant data structures like recipes, schedule entries and batch production records because the process which they connect to, is dependent on the ISA-88 terminology. The corresponding ISA-95 data structures (product definition, production schedule and production performance information) are independent of the process type and therefore more suitable for communications within the level 3 and between the levels 3 and 4. [21]

The data exchange interface of the MES is implemented using the OPC UA technology. The MES has an integrated OPC UA server which provides the relevant data to the clients using configurable information models. The address space contains a data structure similar to the PCS model presented in the figure 13. The data structure represented in the OPC UA address space consists of four objects which correspond to the three activities which communicate with the level 2 systems as defined the ISA-95 production operations activity model.

The product definition management functions handle the product definitions as ISA-88 recipes. The recipes which represent the different products the manufacturing equipment is able to produce, are defined as master recipes. The master recipes are based on site recipes which define the products that a manufacturing site is able to produce. Batch-specific definitions on the other hand, are defined as control recipes which contain the detailed rules and the information of the required resources to produce a single batch. When a batch is scheduled in the detailed scheduling activity group and a work order is dispatched to production, the MES creates an equipment-specific control recipe according to the resource availability and saves it to the data storage which is also accessible through the OPC UA interface.

Controlling the batch process execution is handled by the production execution management functionalities. When a work order is created in the detailed scheduling activity, it is dispatched to the production execution management activity where a unique, batch-specific control recipe and a batch schedule entry data structures are created. By creating the control recipe in the MES level, the production can be optimized according to many variables, for example the availability of equipment and other resources. After that, the control recipe and the scheduled work order are moved to the PCS, where the batch process is executed on the defined time according to the rules in the control recipe.

The production data collection activity manages all the batch production data. It receives production events and alarms and collects the process information, for example sensor readings, actuator states or any other information which is significant for the later production tracking and tracing functionalities. The information is stored into ISA-88 batch production record data structures.

In addition to the data structures mentioned above, the data exchange interface of the MES implements asynchronous events and remotely invokable methods as well. The events transmitted from the MES consist for example of control recipe added and batch scheduled events. The methods represent functionalities for updating the production process status and reporting the production information. A typical operations flow of the MES is presented in the figure 15 as a UML sequence diagram.

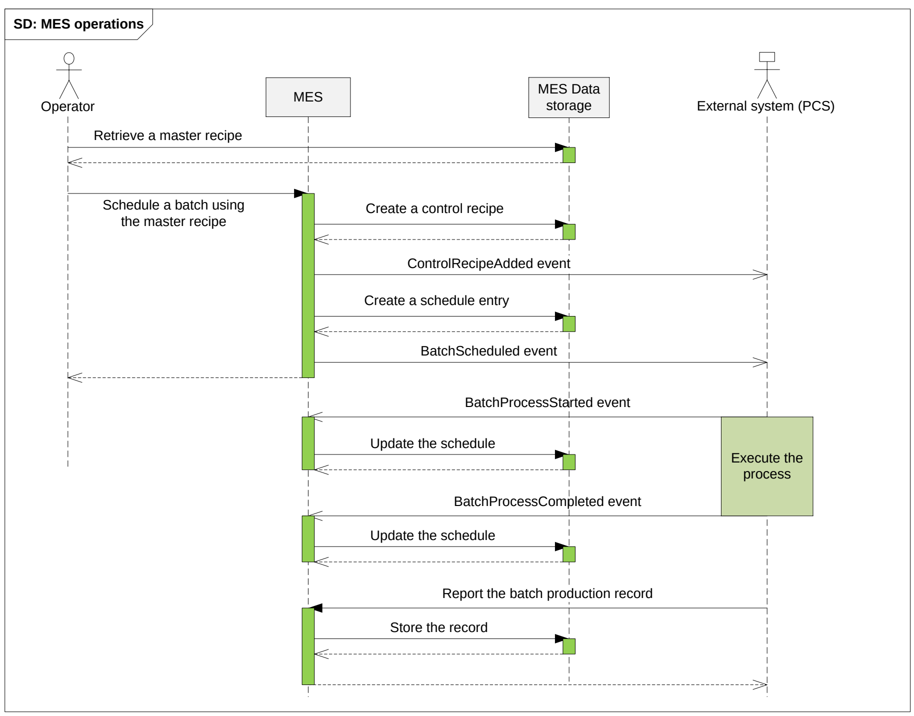

Figure 15: The sequence diagram of the MES activities

In the beginning of the presented operations sequence, an operator selects a product (master recipe) in the user interface of the manufacturing execution system and schedules a work order of the selected product. A control recipe is created and the work order is stored in the dispatch list. Corresponding events are sent to the PCS. The PCS updates the control recipe and schedule information from the MES and executes the process accordingly. When events of process execution are received by the MES, it updates the status of the manufacturing process in its memory. After the batch process has completed, a reporting method is called and the production information sent to the MES as a batch production record data structure.

6.2.3 Integration middleware

The middleware application follows the business process integration-oriented application integration approach. It is an integration server which uses software adapters for the data exchange communications with other information systems. The adapters enable the integration server to connect to different types of systems using different types of buses, protocols and data structures. The data content handled within the middleware is be represented in XML format, so the adapters present the system information in XML using appropriate XML schemas.

In this integration scenario, the data structures handled by the middleware are ISA-88-compliant XML messages. The XML messages follow the BatchML/B2MML schemas defined by the World Batch Forum (WBF). The schemas represent the ISA-88/95-consistent data structures for XML messages.

An abstract depiction of the integration server architecture is presented in the figure 16 as a UML component diagram. The business processes are represented as workflows which are separate executable modules in the integration server. The workflows are executed by the business process engine, which is the core of the integration server. The workflows can communicate with the business process engine using the defined interfaces. The communications to the MES and DCS are managed with the corresponding OPC UA adapters. The integration server contains also an user interface, where the execution of the workflows can be monitored and controlled.

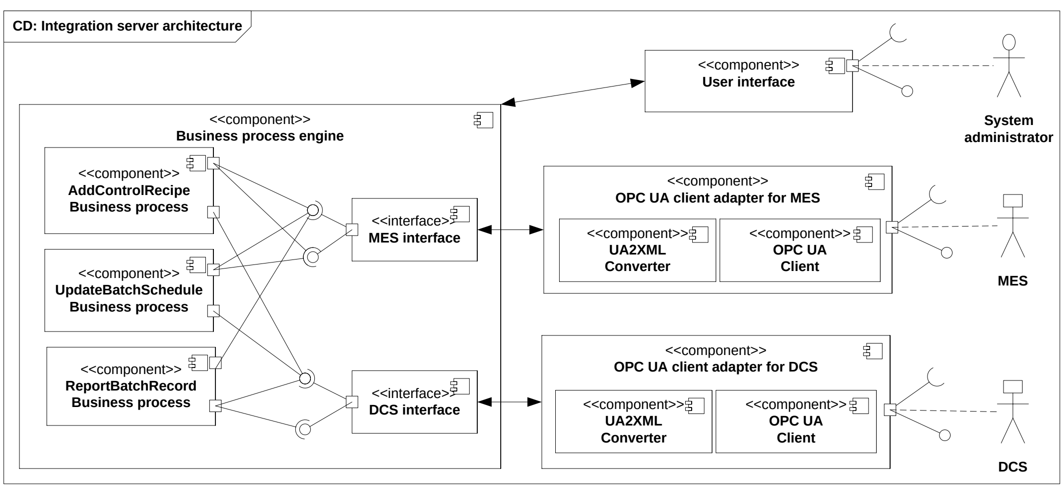

Figure 16: The integration server architecture as a UML component diagram

The business process engine of the integration server executes the business process workflows which are described as graphic orchestration diagrams. An orchestration diagram defines the business process execution path and all the data transactions and transforms involved in the process. The business processes related to the production operations management application integration divide again into three groups consistent with the three integration use cases: product definition management, production execution management and production data collection.

The business processes of the product definition management enable exchanging and editing the recipes or other production instructions between the manufacturing execution system and the process control system. As an example, the business process responsible for adding a control recipe to the PCS is presented in the figure 17. When a batch is dispatched to production, a control recipe is created in the MES and a corresponding event is raised. The business process then reads the control recipe from the MES and writes it to the memory of the PCS.

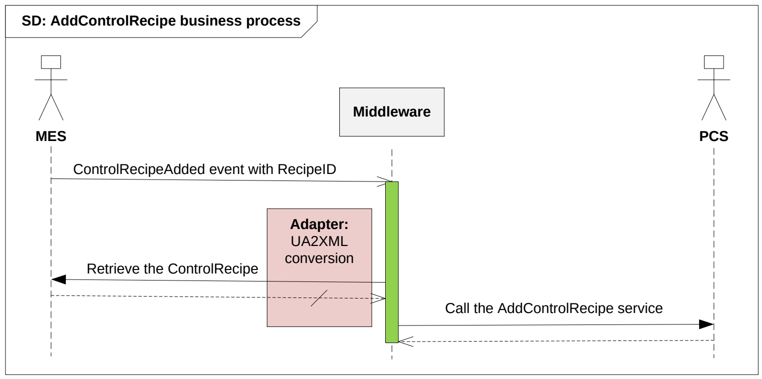

Figure 17: The sequence diagram of the AddControlRecipe business process

The business processes involved in the production execution management enable the MES to monitor and control the process execution. Among other things, the business processes enable transmitting and editing work orders, sending production commands and transmitting events between the MES and PCS. The business process responsible for updating the work order status between the MES and PCS is presented in the figure 18. When a batch is previously scheduled, a work order has been stored in the production schedule dispatch list and moved to the PCS by another business process. When a BatchProcessStarted event is raised in the PCS, the business process reads the work order from the PCS and updates it to the MES. When a BatchCompleted event is raised, the work order is read again from the scheduled batch list of the PCS and updated to the MES.

The business processes managing the production data collection receive production events and alarms and transmit all the data concerning a batch process to the MES. Collecting the batch production process-related information is important for product tracking and tracing. The business processes can be event- or time-driven. As an example, the business process responsible for reading the batch production record which, among other informations, contains the actual size of the completed batch and the exact execution times of the batch process, is visualized in the figure 19. When a BatchProcessCompleted event is raised, the business process reads the batch production record data structure from the PCS and stores it to the MES by calling the AddProductionRecord method.

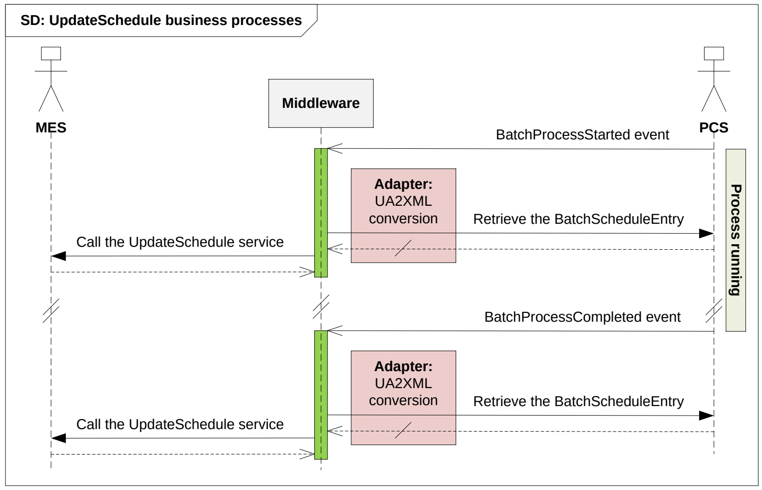

Figure 18: The sequence diagram of the UpdateScheduledBatch business process

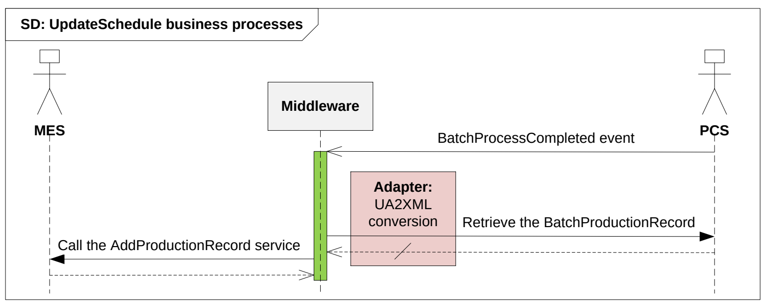

Figure 19: The sequence diagram of the ReportBatchRecord business process

6.2.4 OPC UA adapters

The integration server uses the adapter technique to connect to the other information systems using the OPC UA technology. An OPC UA adapter contains an OPC UA client which is used to communicate with the OPC UA servers hosted in the MES and DCS. The OPC UA client module contains implementations for all the OPC UA services necessary for this integration scenario. The adapter wraps the OPC UA client functionalities behind an interface which is understandable by the integration server. In addition, an information transform from the OPC UA object model to XML elements is needed. The OPC UA adapter is thus a thick adapter.

The data structures in the OPC UA servers are ISA-88/95-compliant OPC UA object models and the messages handled in the business processes use the XML format and follow the BatchML and B2MML schemas. Therefore the adapter must be able to transform the information from OPC UA objects to XML and vice versa. The basic idea of this serialization/deserialization functionality is presented in the figure 20.

The conversion method is called the UA2XML conversion. It is based on a rule file, which describes how the OPC UA objects are mapped to XML elements according to their OPC UA type information. The file simply contains the rules about what

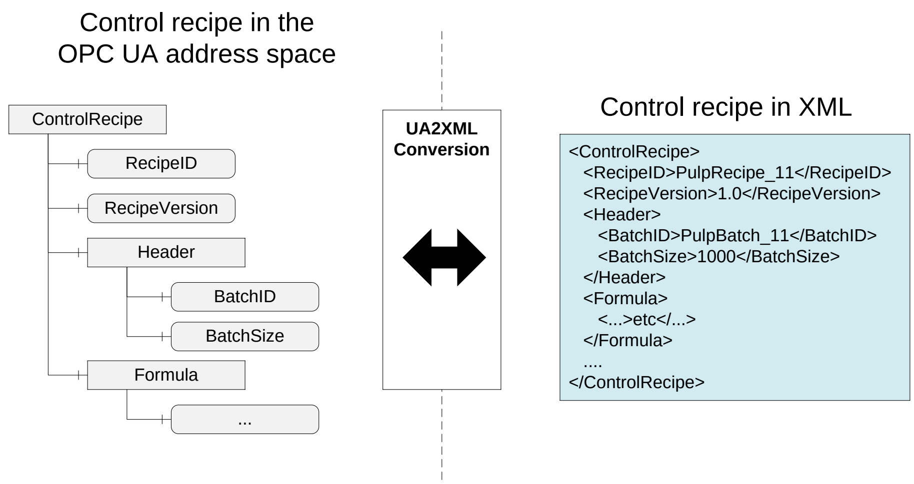

Figure 20: The UA2XML conversion

kind of an XML element structures OPC UA object types are serialized to . The UA2XML conversion is done every time when objects are exchanged through the OPC UA adapter.

7 Implementation and testing

7.1 Experimental prototype implementation

The implemented solution for the integration scenario follows fully the requirements and the design discussed in the previous sections. An overview of the implemented production operations management integration environment is presented in the figure 21. It contains all the significant participants involved in a real-life integration project in the industry.

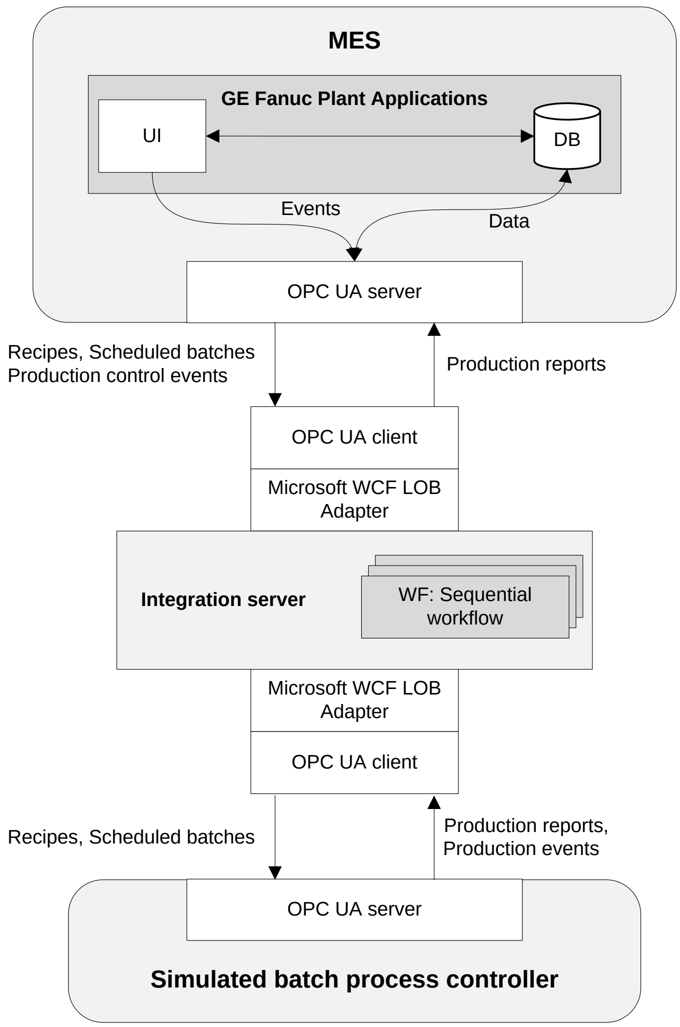

Figure 21: The implemented integration environment

The implementation of the integration environment is explained in detail below.

7.1.1 Data exchange interfaces

The OPC UA interfaces are implemented using the OPC Foundation's OPC UA software development kit for Microsoft .NET. Creating OPC UA server and client applications is possible using Microsoft .NET programming languages like C# with the OPC UA .NET SDK.

7.1.2 Manufacturing execution system

The actual manufacturing execution system implementation is based on the software by GE Fanuc. It employs Proficy Plant Applications software for operations management and Proficy Historian for the manufacturing and process information database management.

The MES operator has a simple user interface where the batch production rules (recipes) can be defined, work orders created and production scheduled. The created control recipes are derived from the product-specific master recipes. The database contains all the manufacturing and process data. GE Fanuc provides an API for Proficy Historian, thus the OPC UA server application is implemented to have a direct access to the database information. It is also possible to direct events from the Plant Applications through the OPC UA objects using the API.

GE Fanuc MES system provides a hybrid application interface: it is basically information-oriented, but provides also SOA-based tools to develop business logic workflows within the system using the Proficy SOA platform and Proficy Workflow software. These workflows can manage data transactions and consume remote services of other information systems. [11, 38]

The OPC UA address space is presented in the figure 22. It shows a ISA-88-compliant and BatchML-based batch information data structure which contains batch lists and recipes. Also a MES object is included. It contains the functionalities that are used by the clients, for example the ReportBatchCompleted method. The MES object is also the source of the ControlRecipeAdded and BatchScheduled events.

7.1.3 Process control system

The physical connection to the process equipment was not implemented in this work. Instead the batch process control system and the physical batch process are simulated with a DCS simulator software. In this implementation, the DCS simulator application acts as an ISA-88-compliant process controller. The simulator software can be extended to control the actual process equipment in the future work.

The DCS simulator contains a simple user interface where the control recipes and work orders are visible. Using the DCS simulator, the execution of the work orders can be simulated with the start and stop buttons.

The OPC UA server is implemented as a part of the DCS simulator software.

It has a direct access to the process control data and can therefore be used for business-to-manufacturing integration as well as for the data exchange interface for SCADA systems. The OPC UA address space provided by the DCS simulator is shown in the figure 23. Similar to the MES interface, it contains a BatchML-based batch information structure containing control recipes and batch lists. It also contains a DCS object which includes the DCS functionalities provided for client applications, for example the AddControlRecipe and AddScheduledBatch methods and the BatchRunning and BatchCompleted events.

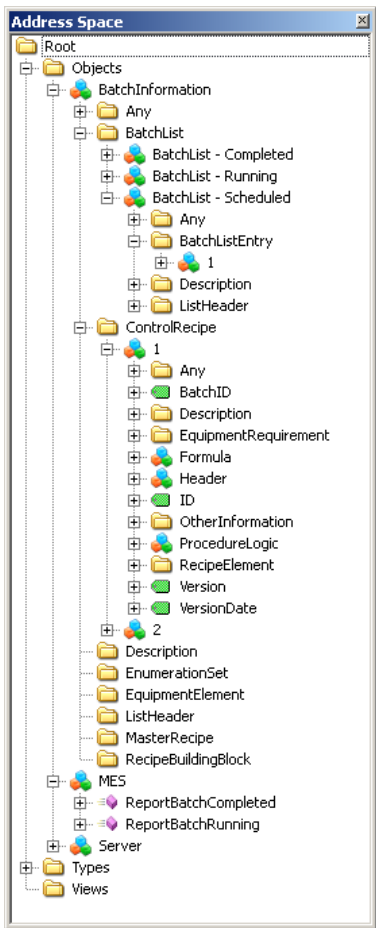

Figure 22: The address space of the MES OPC UA server

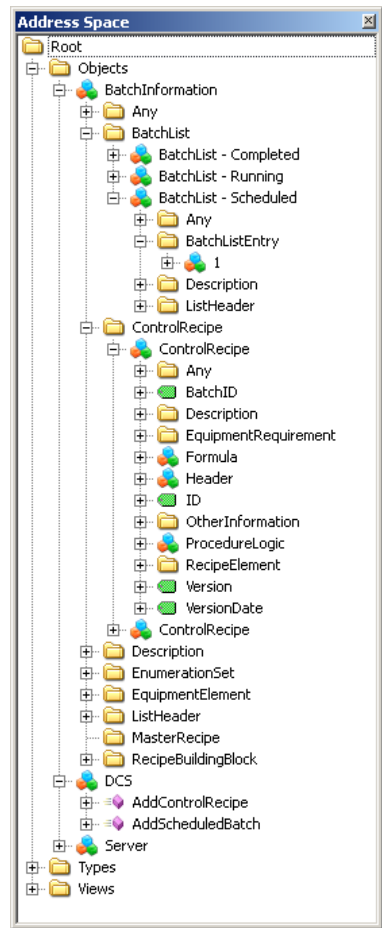

Figure 23: The address space of the DCS OPC UA server

7.1.4 OPC UA client adapters

The OPC UA adapter includes an OPC UA client implemented with the OPC UA .NET SDK. It can consume all the OPC UA services necessary for this integration scenario.

The adapter itself is programmed using Microsoft's Windows Communication Foundation (WCF) Line-of-business (LOB) adapter technology. This technology is based on the Windows Communication Foundation, a framework for creating applications with service-oriented architecture for the .NET platforms. An adapter created with the WCF LOB Adapter SDK is an implementation of a reusable and metadata-rich adapter consumable by business process workflows in a BizTalk server but also by any custom .NET application. [39]

The adapter implements an interface similar to the OPC UA client interface which contains methods for using the OPC UA services. In addition to the necessary services defined in the OPC UA specification, the adapter also implements the ReadToXml service as an extension to the Read-service. The ReadToXml service executes an UA2XML conversion, i.e. a functionality that serializes an object from the OPC UA address space into XML format with defined schema. Using this service, the adapter consumer applications can read for example control recipes straight into a well-formatted XML message.

7.1.5 The middleware application

The middleware application is implemented as a simple business process integration-oriented integration server. The actual integration server program is a stand-alone .NET application which is hosting and executing workflows which represent the business processes responsible for managing data transactions, consuming remote services and transmitting events in the application integration. The business process workflows executed in the integration server use the OPC UA client adapters to connect to the information systems hosting OPC UA servers.

The business processes are implemented using the Windows Workflow Foundation (WF) technology. With WF, it is possible to create business processes as simple, graphic sequential or state machine workflow diagrams and host them in any .NET application. An example of an implemented WF sequential workflow is presented in the figure 24. It presents the business process responsible for transmitting a scheduled work order (a BatchListEntry object) from the MES to the PCS. When it is loaded to the workflow runtime engine, it subscribes to receiving events from the MES. When a BatchScheduled event is received, the workflow reads the scheduled batch from the MES to an XML file using the ReadToXml method of the OPC UA client adapter. Finally, the AddScheduledBatch method of the PCS object is called in the OPC UA server of the PCS and the BatchListEntry XML file is sent as a parameter. The AddScheduledBatch method dynamically adds a new BatchListEntry object to the BatchInformation data structure in the address space of the OPC UA server of the DCS simulator.

The integration server is written in C# and its architecture is illustrated in the figure 25. The components are divided into four namespaces. The GUI namespace contains the main class GUI which contains an instance of the workflow runtime engine. The GUI class also manages the workflow diagrams which belong to the Workflows namespace and are located in separate DLL files. The GUI contains two OPC UA client adapters, the MESAdapter and DCSAdapter which realize the

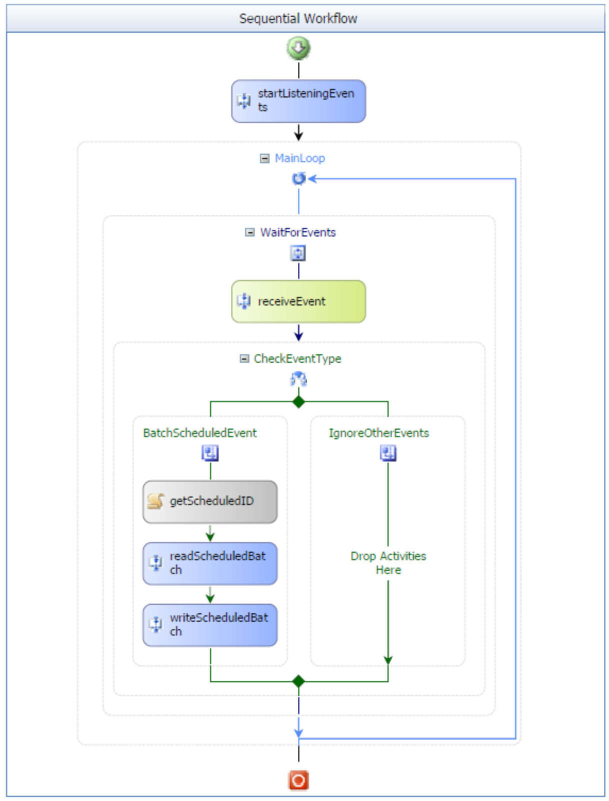

Figure 24: An example of the implemented workflow diagrams

interfaces MESInterface and DCSInterface respectively. With predefined interfaces for the OPC UA client adapters, the business process workflows can be developed independently of the integration server. The business process workflows are built into dynamic-link library (DLL) files and can be loaded dynamically to execution into the WF runtime engine in the integration server.

7.2 Performed tests and demonstrations

With the previously described system environment, all the three production operations management integration use cases were successfully implemented. They can be demonstrated with the following examples. The first represents the product definition management functionality, second, third and fourth represent the production execution management functionality and the fifth represents the production data collection functionality.

1. **Adding a control recipe**

   When a master recipe is selected and the button "Create batch" is pressed in the MES user interface, a control recipe based on the corresponding master recipe is created and stored to the OPC UA address space. After that, a ControlRecipeAdded event is raised. The SendControlRecipe workflow which

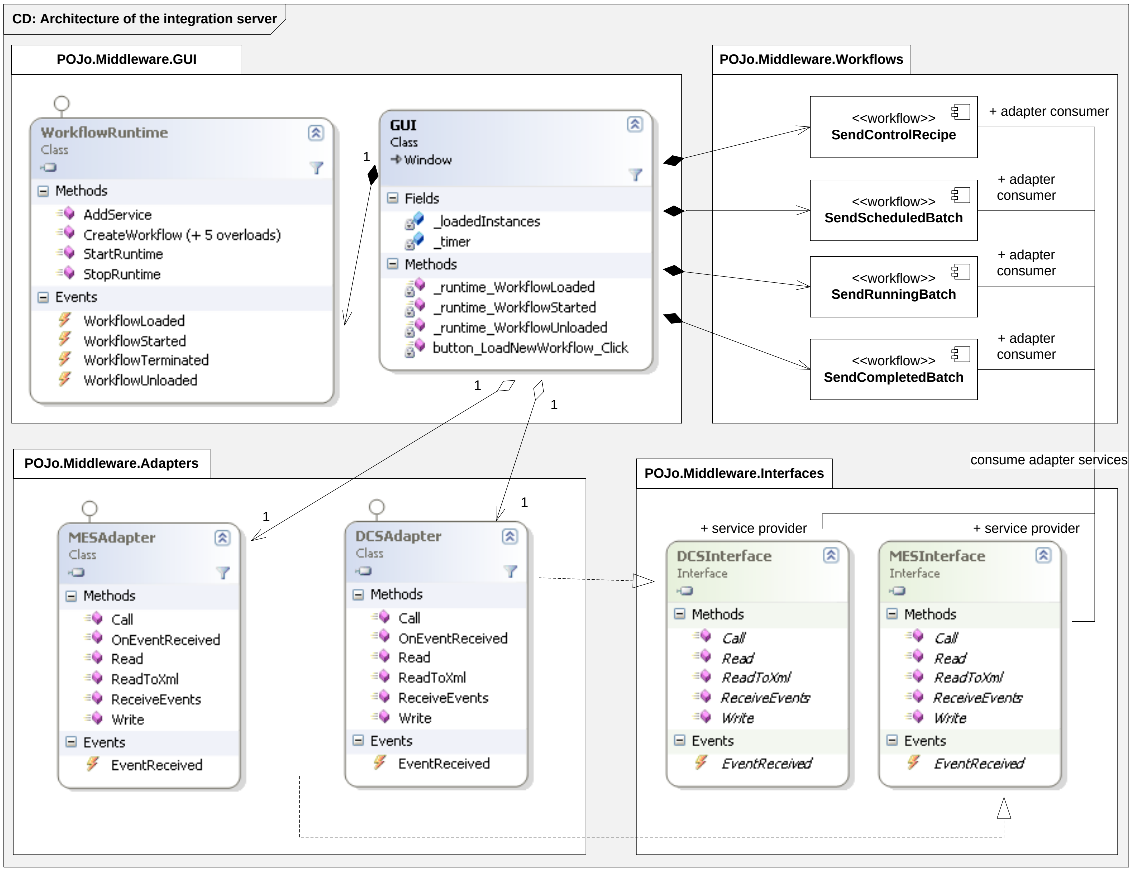

Figure 25: The class diagram of the implemented integration server

was waiting for the event, starts executing its business process. It reads the control recipe from the OPC UA server of the MES to an XML message and calls the AddControlRecipe method of the DCS object in the address space of the DCS OPC UA server. The control recipe is then stored to the DCS memory and becomes visible in the BatchInformation data structure.

2. **Adding a scheduled batch**

   When a control recipe is selected, requested start and stop times entered and the button "Schedule batch" pressed in the MES user interface, a work order is created as a BatchListEntry object and stored to the "BatchList - Scheduled" batch list in the OPC UA address space. Also an event BatchScheduled is raised. Upon detecting the event, the SendScheduledBatch workflow starts executing the business process. It reads the work order from the OPC UA server of the MES to an XML message and calls the AddScheduledBatch method of the DCS object in the address space of the DCS OPC UA server. The work order is then stored to the DCS memory and becomes visible in the BatchInformation data structure.

3. **Starting the process**

When a work order is selected and the ”Start process” button pressed in the user interface of the DCS simulator, a BatchRunning event is raised in the DCS object in the address space of the DCS OPC UA server. The unique ID of the work order is passed with the event as a parameter. Upon detecting the event, the SendRunningBatch workflow starts executing the business process. It reads the work order from the DCS to an XML message and calls the ReportBatchRunning method of the MES object in the MES OPC UA address space. The work order is removed from the ”BatchList - Scheduled” batch list and added to the ”BatchList - Running” batch list in the address space of the MES OPC UA server. The change of the batch status is also visible in the MES user interface.

4. **Stopping the process**

   When a work order is selected and the ”Stop process” button is pressed in the user interface of the DCS simulator, an event BatchCompleted is raised in the DCS object in the OPC UA address space. The unique ID of the work order is passed with the event as a parameter. Upon detecting the event, the SendCompletedBatch workflow starts executing the business process. It reads the work order from the DCS to an XML message and calls the ReportBatchCompleted method of the MES object in the OPC UA address space of the MES. The work order is removed from the ”BatchList - Running” batch list and added to the ”BatchList - Completed” batch list. The change is also visible in the MES user interface.

5. **Reporting the production information**

   The SendBatchRecord workflow also detects the BatchCompleted event and starts executing the business process. It reads the batch production record data structure from the DCS to an XML message and calls the ReportBatch method of the MES object in the MES OPC UA address space. As the batch production record data structure is not yet implemented in the BatchML schema set, it is simulated with a batch list entry data structure. The reported batch information is stored to the MES database.

8 Conclusions

8.1 Summary

In this work, the goal was to study the integration of production operations management between a manufacturing execution system and a process control system and to evaluate the usability of the OPC UA technology in this integration scenario. The study also included designing the integration environment based on the ISA-88/95 hierarchy and activity models and using the standard data structures in the integration. As a result, a part of an integrated system environment, a testbed, was implemented.

In the section 5, the requirements for the application integration were set and the integration scenario was divided into three use cases according to the ISA-88 control activity model and the ISA-95 production operations activity model: the integration of the product definitions management, production execution management and production data collection functionalities. In the section 6, a design of an integration solution was introduced. The solution fulfilled the previously set requirements and was based on the technologies explained in the sections 2 and 3. In the section 7 an experimental implementation of the design was presented.

Each integration project where a manufacturing execution system and a process control system are to be integrated, has different, case-specific requirements. Despite this fact, the abstract integration design proposed in this work can be used in different integration scenarios to give guidelines for a well-performing and adaptive integration solution. The design presented in this work provides a flexible and system-independent solution, which offers many possibilities for effective integration solutions.

Although the business processes implemented in this work were functionally simple, they demonstrate well the functionality of the integration design. By using the application integration design presented in this work, an effective and flexible solution for both vertical and horizontal integration can be achieved.

8.2 OPC UA in application integration

In the section 3, the OPC UA specification was studied from the parts essential to the application integration in the production operations management. With the services defined in the specification, the technology provides diverse possibilities for data transactions. With synchronous read and write operations, commands and asynchronous events it sets no limits to the data exchange. Information models with complex semantics can be used to deliver different data structures used in the integration scenarios. In addition to these mentioned aspects, the OPC UA specification also provides modern security features by defining the authorization, encryption and user control functionalities.

An extensive system integration requires flexible interfaces which are independent of the data structures and system platforms. In comparison to old OPC interfaces, the new OPC UA technology brings many advantages. With the functionalities mentioned in the previous paragraph, the OPC UA technology is suitable for complicated system integration scenarios. Taking the modern security and networking issues into account, the technology is suitable for business-to-manufacturing and even business-to-business integration even through public networks, like the Internet. Based on the implemented use cases, it is obvious that a broad use of the functionalities provided by the OPC UA (configurable data models, methods, events) simplify the integration cases significantly.

8.3 Business process integration-oriented design

In sections 2 and 3, the production operations management and the application integration techniques were discussed. The production operations management are divided into activity groups in the production operation management activity model defined in the part 3 of the ISA-95 standard. This model also defines the relations of the activities and the data transactions between them in an abstract level. In addition, partly overlapping functionalities are defined in the control activity model in the ISA-88 standard. In the scope of this work, these two models for production operations management within and between the MES and PCS levels form a logical basis for the proposed solution for the integration scenario presented in the section 6.

In the section 6, the design of the production operations management integration solution was presented and explained in detail. The center of the solution is the integration middleware, an integration server application. The integration server employs adapters which provide interoperability with other systems using different buses, protocols and data structures. Using this kind of middleware and adapter architecture, the integration design is independent of the other systems involved in the integration scenario. The middleware thus stays independent of the data structures and system platforms.

The integration server only represents the framework used for the integration; it does not contain the actual integration logic involved. The integration server is based on the business process integration-oriented technology, where the actual integration logic is implemented as business processes which handle the data transactions and transformations and communicate with other systems through adapters. With the business process-oriented architecture and the adapter architecture combined, the integration logic can be separated from the involved systems. The business processes only handle information through adapters, which hide the actual protocols and other technical details beneath them.

The represented design was chosen because the integration solution between the manufacturing execution system and the process control system must be adaptive to different systems and data structures. With the mentioned features it provides a flexible integration platform setting no limits to the integration logic which is a separate and independent part of the integration.

The ISA-88 and ISA-95 standards proved to be suitable guidelines for designing and implementing the application integration. Also the data structures defined in the ISA-88/95 standards are a remarkable improvement for the collaboration of systems. By designing the system interfaces to provide information based on standards like ISA-88 and ISA-95, the integration process will be significantly simplified.

8.4 Further discussion

8.4.1 Usability of the OPC UA technology

In this work, the OPC UA technology proved its capabilities for application integration although it was only tested with a small data load. As the integration of MES requires massive amounts of data to be transferred, the performance of the OPC UA data exchange functionalities should be studied more extensively.

Since there are only few commercial systems providing OPC UA interfaces yet, it remains to be seen how well the configurable address spaces will be supported in MES and PCS systems.

8.4.2 Design

In the section 7, the implemented application integration testbed was presented. The work shows that the proposed design is suitable for integrating information systems across the system hierarchy levels. However, as the implemented use cases were simple, the actual performance of the integration design with more complicated business processes and larger amounts of data still stays untested.

The proposed integration solution adds more levels and interfaces for the communications path which may effect to the integration efficiency as increased communication delays and reduced reliability.

8.5 Future work

The goal of this work was not to build the system environment only for demonstrational purposes. However, in order for the environment to function as an extensive platform for future research and educational purposes, a lot of future work for the testbed is still required.

In the process control level, a real process control system should be implemented to replace the DCS simulator software. A commercial automation system should be used here for the environment to represent a real-life manufacturing company. By implementing an OPC UA interface for a commercial process controller system, the possibilities of using the OPC UA functionalities and data structures in the data exchange with SCADA systems could also be studied.

Another possibility is to forget the batch processes and to implement an MES for a discrete process. It could be integrated to the control system of the production line equipment located in the laboratory facilities. Also using the ISA-95 information models or other standards in the integration of an MES and a discrete process control system could be studied.

All the functionalities required by the integration of the production operations management activities were successfully implemented in this work. In the future, more activities should be implemented to the manufacturing execution system in order to study more complex integration scenarios which require larger amounts of data and involve more complex business processes in the integration server. This work could be done as a student project.

Also implementing a simple ERP system in the system environment should be done. By integrating the MES to the ERP system, a full production operations management integration could be implemented. The communications between the ERP and MES using the OPC UA technology and the ISA-95 information models would be an interesting subject for research.

The integration server could be extended to support BPEL-compliant business process definitions. This way the business process definitions could be even more independent of the integration server and other systems. On the other hand, to use commercial software, the middleware could be replaced by the Microsoft BizTalk Server which is also able to consume the OPC UA client adapters implemented in this work.

The use of the OPC UA technology in the data exchange between the activities of a manufacturing execution system could be an interesting research topic. This idea, proposed by D. Brandl [40], includes implementing the data exchange interfaces for the activities of the ISA-95 activity models. The interfaces and the data contents will be the main topic of the part 4 of the ISA-95 standard which is still unreleased.

References

[1] J. Kletti, *Manufacturing Execution System - MES*. Springer, 2007.

[2] S. N. Sahu, "*Close Gap For Adaptive Manufacturing*," *AutomationWorld.com*, 2007. Available on-line: http://www.automationworld.com/news-3162, Referenced 2.5.2010.

[3] International Society of Automation, *ANSI/ISA-95.00.01-2000, Enterprise-Control System Integration Part 1: Models and Terminology*, 2000.

[4] International Society of Automation, *ANSI/ISA-95.00.03-2005, Enterprise-Control System Integration, Part 3: Models of Manufacturing Operations Management*, 2005.

[5] H. Meyer, F. Fuchs, K. Thiel, *Manufacturing Execution Systems*. The McGraw-Hill Companies, 2009.

[6] B. Scholten, *The road to integration - A guide to applying the ISA-95 standard in manufacturing*. ISA - Instrumentation, Systems and Automation Society, 2007.

[7] MESA International, *MES Functionalities & MRP to MES Data Flow Possibilities*, 1997. Available on-line: http://www.mesa.org/, Referenced 2.5.2010.

[8] M. McClellan, *Applying Manufacturing Execution Systems*. St. Lucie Press, 1997.

[9] M. McClellan, "*Introduction to manufacturing execution systems*," in *MESA 2001 MES Conference & Exposition*, 2001. Available on-line: http://www.cosyninc.com/papers/3.pdf, Referenced 2.5.2010.

[10] C. Giffor et al., *The hitchhiker's guide to manufacturing operations management - ISA-95 best practices book 1.0*. International Society of Automation, 2007.

[11] D. Linthicum, *Next generation application integration*. Addison-Wesley, 2004.

[12] R. E. Parapar, "*MES-DCS Integration Case Study*," in *World Batch Forum - 2001 North American Conference*, 2001. Available on-line: http://www.controlglobal.com/Media/News/080826_WBF_MES-DCS.pdf, Referenced 2.5.2010.

[13] MIMOSA, *Oil&Gas / PetroChem Industry OpenO&M Interoperability Use Cases & Scenarios*, 2008. Available on-line: https://www.posccaesar.org/raw-attachment/wiki/IOHN/InformationDissemination/Oil%26Gas-PetroChem%20Industry%20OpenO%26M%20Interoperability%20Use%20Cases%20--%20Complete%20(updated%202008-08-22).ppt, Referenced 2.5.2010.

[14] S. Karnouskos et al., "*Integration of SOA-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure*," in *12th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, (Patras, Greece), 2007.

[15] L.M.S. de Souza et al., "*SOCRADES: A Web Service Based Shop Floor Integration Infrastructure*," in *Proceedings of the Internet of Things (IOT)*, Springer, 2008.

[16] M. Stopper, B. Katalinic, "*Service-oriented Architecture Design Aspects of OPC UA for Industrial Applications*," in *International Multi Conference of Engineers and Computer Scientists (IMECS), Vol. II*, (Hong Kong), 2009.

[17] M. Schleipen, "*OPC UA Supporting the Automated Engineering of Production Monitoring and Control Systems*," in *13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, (Hamburg, Germany), 2008.

[18] OPC Foundation. The web page of OPC Foundation. Available on-line: http://www.opcfoundation.org/, Referenced 2.5.2010.

[19] W. Mahnke, S.-H. Leitner, M. Damm, *OPC Unified Architecture*. Springer, 2009.

[20] OPC Foundation, *OPC Unified Architecture Specification, Part 4: Services*, 2009. Release 1.01.

[21] B. Scholten, *Integrating ISA-88 and ISA-95*, 2007. Available on-line: http://www.isa.org/filestore/Division_TechPapers/GlassCeramics/Integrating_TP07EXPO140.pdf, Referenced 2.5.2010.

[22] J. Parshall, L. Lamb, *Applying S88 - Batch control from a user's perspective*. ISA - The Instrumentation, Systems and Automation Society, 2000.

[23] International Society of Automation. The ISA-88 web page of International Society of Automation. Available on-line: http://www.isa.org/ISA88/, Referenced 2.5.2010.

[24] International Society of Automation, *ANSI/ISA88.011995, Batch Control Part 1: Models and Terminology*, 1995.

[25] International Society of Automation, *ISA Draft 88.00.02, Batch Control Part 2: Data Structures and Guidelines for Languages*, 2000. Draft 18.

[26] International Society of Automation, *ISA Draft 88.00.05, Batch Control Part 5: Implementation Models & Terminology for Modular Equipment Control*, 2008. Working draft 4.

[27] D. Brandl, *Business to manufacturing (B2M) collaboration between business and manufacturing using ISA-95*, 2007. Available on-line: http://www.cpdee.ufmg.br/~seixas/Especializacao/Download/DownloadFiles/Brandl%20-%20Business%20to%20Manufacturing%20Collaboration.pdf, Referenced 2.5.2010.

[28] International Society of Automation. The ISA-95 web page of International Society of Automation. Available on-line: http://www.isa.org/ISA95/, Referenced 2.5.2010.

[29] International Society of Automation, *ANSI/ISA-95.00.02-2001, Enterprise-Control System Integration, Part 2: Object Model Attributes*, 2001.

[30] International Society of Automation, *ISA Draft Standard 95.01.04, Enterprise-Control System Integration, Part 4 Section 1: Common Object Model and Attributes*, 2006. Draft 2.

[31] International Society of Automation, *ANSI/ISA-95.00.05-2007, Enterprise-Control System Integration, Part 5: Business-to-Manufacturing Transactions*, 2007.

[32] D. Brandl, *What is ISA-95? - Industrial best practices of manufacturing information technologies with ISA-95 models*, 2008. Available on-line: http://www.apsom.org/docs/T061_isa95-04.pdf, Referenced 2.5.2010.

[33] World Batch Forum. The web page of the World Batch Forum. Available on-line: http://www.wbf.org/, Referenced 2.5.2010.

[34] World Batch Forum, *Batch Markup Language BatchML - BatchInformation*, 2008. Version 0401.

[35] P. Arrenius, "*Evaluation environment for new technologies in automation design*," Master's thesis, Helsinki University of Technology, Department of automation and systems technology, Espoo, Finland, 2006.

[36] A. Tuomi, "*Application integration for condition based maintenance*," Master's thesis, Helsinki University of Technology, Department of automation and systems technology, Espoo, Finland, 2010. Draft 1.

[37] OPC Foundation, *OPC UA For Devices 1.00 Companion Specification*, 2009. Release 1.00.

[38] GE Intelligent Platforms. The web page of GE Intelligent Platforms. Available on-line: http://www.ge-ip.com/, Referenced 2.5.2010.

[39] Microsoft, "*Microsoft Windows Communication Foundation Line of Business Adapter SDK*." Available on-line: http://msdn.microsoft.com/en-us/library/bb798080(v=BTS.10).aspx, Referenced 2.5.2010.

[40] D. Brandl, "*OPC and MES*." Presentation and slides at OPC day 2009, Espoo.