TAMPERE UNIVERSITY OF TECHNOLOGY

AKI KÖSSILÄ

PRODUCTION SCHEDULING IN COPPER ELECTROREFINING AND ELECTROWINNING PLANTS WITH MANUFACTURING EXECUTION SYSTEMS

Master of Science Thesis

Examiner: Professor Hannu Koivisto
Examiner and topic approved in the Council meeting of Depart of Automation Science and Engineering on 8.2.2012

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Automation Engineering
KÖSSILÄ, AKI: Production Scheduling in Copper Electrorefining and Electrowinning Plants with Manufacturing Execution Systems
Master of Science Thesis, 89 pages, 26 Appendix pages
August 2013
Major: Automation and Information Networks
Examiner: Professor Hannu Koivisto
Keywords: Manufacturing Execution System, MES, scheduling, dispatching, ISA-95, workflow, Service Oriented Architecture, SOA, electrorefining

Copper electrolysis process is used to produce pure, over 99.99% copper with the help of electricity. Production dispatching and scheduling in copper electrolysis processes have been done mostly manually and without direct link to automation systems. Disruptions in the production processes usually require new decisions from production planners and changes to production schedules. Manual transfer of information is prone to errors and the transfer process usually takes longer time. In copper electrolysis production scheduling is put into practice by controlling the cranes, which are handling all the material transfers in the plant.

One of the objectives of this thesis was to research how to implement new production dispatching features to the existing manufacturing execution system. Theoretical part of this thesis consists of introduction of suitable technologies and standards that could be considered as basis for the empirical research.

The foundation for a fully automatic production scheduling and dispatching is a reliable communication between the manufacturing execution system and the automation system. This thesis work contains a review of multiple standards and ready-made commercial products that support these dispatching operations.

The empirical part of this thesis consists of design, implementation and testing of a daily production schedule software for cranes. The implementation includes communication and data collection infrastructure but all user interface components are out the scope of this thesis. Also the high level fully automatic scheduling workflow was designed in a high level but was not implemented during this project.

This implementation leveraged new products in the field of software technology. Suitability of these products for this and similar projects were evaluated. Most of the problems faced during the implementation were related to a commercial software platform selected for the implementation. Therefore it was suggested that the further use of the commercial product will be put on hold until these problems get fixed by possible software updates. Further development is related to the fully automatic scheduling and also for the development of the user interface.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Automaatiotekniikan koulutusohjelma
KÖSSILÄ, AKI: Kuparielektrolyysiprosessin aikataulutus tuotannonohjausjärjestelmillä
Diplomityö, 89 sivua, 26 liitesivua
Elokuu 2013
Pääaine: Automaatio- ja informaatioverkot
Tarkastaja: professori Hannu Koivisto
Avainsanat: tuotannonohjaus, aikataulutus, ISA-95, palvelukeskeinen arkkitehtuuri

Kuparielektrolyysiprosessissa tuotetaan sähkövirran avulla puhdasta, yli 99.99% kuparia. Kuparielektrolyysiprosessien tuotannon ohjaaminen ja tuotannon aikataulutus on tapahtunut tähän asti hyvin pitkälti manuaalisesti ja ilman suoraa yhteyttä automaatiojärjestelmään. Poikkeukset tuotantoprosessissa vaativat usein tuotannonsuunnittelijoiden päätöksiä ja muutoksia aikatauluihin. Manuaalisessa tiedonsiirrossa tapahtuu helposti virheitä ja tiedon välittäminen operaattoreille on hidasta. Tuotannon käytännön aikataulutus tapahtuu ohjaamalla nosturia tai nostureita, jotka ovat keskeisessä asemassa hoitaen kaikki materiaalien siirrot.

Tämän työn tavoitteena on tutkia miten olemassa olevaan tuotannonohjausjärjestelmään lisättäisiin aikataulutustoiminnallisuus ja kaksisuuntainen kommunikaatio automaatiojärjestelmään. Teoriaosiossa käydään läpi toteutukseen soveltuvia teknologioita ja standardeja, jotka ovat perustana käytännön toteutukselle.

Täysin automaattisen aikataulutuksen perustana on kommunikaation muodostaminen ja tiedonsiirto automaatiojärjestelmän ja tuotannonohjausjärjestelmän välillä. Kommunikaatioon liittyen on olemassa useita standardeja ja valmiita ratkaisuja joita työssä on tutkittu ja käytetty.

Käytännön toteutukseen kuului päivittäisen nosto-ohjelman mahdollistavan ohjelmiston suunnittelu, toteutus ja testaus. Ohjelmisto sisälsi kommunikaatio- ja datankeruuinfrastruktuurin. Aikataulutustyökalun käyttöliittymä ei kuulunut työn laajuuteen. Myöskään ylätason täysautomaattista aikataulutusta ei toteutettu osana tätä työtä.

Toteutuksessa käytettiin uusia ohjelmistoteknisiä tuotteita, joiden soveltuvuutta arvioitiin tämän tyyppisiin projekteihin. Suurimmat ongelmat projektin käytännön toteutuksessa liittyivät valittuun valmiiseen ohjelmistotuotteeseen, joka ei ollut käytännössä tarpeeksi vakaa tuotantokäyttöön. Työn pohjalta ehdotuksena on olla käyttämättä kyseistä tuotetta jatkossa kunnes kohdattuihin ongelmiin saadaan korjaukset mahdollisten ohjelmistopäivitysten myötä. Jatkokehityskohteet liittyvät erityisesti täysautomaattiseen aikataulutukseen, sekä tähän läheisesti liittyvään uuteen käyttöliittymään.

FOREWORDS

This thesis project was mainly focused on the actual implementation of the system. The commissioning of the first system using this software and models developed as part of this thesis is now commissioned and handed over to the end customer. This thesis work took about two years longer than expected, because after this first demo implementation I was drawn into commissioning projects that took me to travel around the world, mainly in China and Russia. Even though this took longer than expected I still think it was better this way, because I have had great experiences along the way and I have learnt a lot during this journey of two and a half years.

This Master's thesis was done for Oy Delta-Enterprise Ltd and the customer who ordered this work was Outotec (Finland) Oy. This thesis was part of a large scale manufacturing execution system development project named Integrity.

My special thanks go to everyone in the Integrity development team for support and guidance. My gratitude also lies with Antti Varis who first of all contacted and hired me to work for Delta-Enterprise and he was also commenting and giving advices about this thesis work. I would also like to thank Outotec personnel Martti Larinkari and Ahti Rossi for good advices and support.

CONTENTS

ABBREVIATIONS AND NOTATION

.NET	.NET Framework is a software framework developed by Microsoft
ANSI	American National Standards Institute.
API	Application Programming Interface.
APICS	Advancing Productivity, Innovation, and Competitive Success the Association for Operation Management.
APM	Anode Preparation Machine.
B2MML	Business to Manufacturing Markup Language
BCD	Binary Coded Decimal.
BPM	Business Process Modeling.
C#	Programming language developed by Microsoft.
CIMS	Computer Integrated Manufacturing System.
CLI	Common Language Infrastructure forms the core of the .NET Framework.
CLR	Common Language Runtime is the common execution engine in the .NET Framework.
COM	Component Object Model
CSM	Cathode Stripping Machine.
DB	Data Block, used in PLC.
DCOM	Distributed Component Object Model.
DCS	Distributed Control System.
ER	Electrorefining, a form of electrodeposition process.
ERP	Enterprise resource planning is a term used with systems that provide enterprise level management and reporting functionalities.
EW	Electrowinning, a form of electrodeposition process.
FAT	Factory Acceptance Test.
FDSM	Full Deposit Stripping Machine.
FMF	Finnish MES Forum
GE	General Electric is American multinational conglomerate corporation.
GUI	Graphical User Interface
HMI	Human Machine Interface.
IGS	Industrial Gateway Server
ISA	International Society of Automation.
ISA-88	The international standard ANSI/ISA-88 for flexibility in production.
ISA-95	International standard ANSI/ISA–95 for the integration of enterprise and control systems.
ISO	International Standardization Organization
KPI	Key Performance Indicator.
MES	Manufacturing Execution System is an information system that manages and reports manufacturing operations. The scope of the system is between ERP and automation layers.

MESA	Manufacturing Enterprise Solutions Agency is a nonprofit global community of solution providers, manufacturers and other industry professionals.
MOM	Manufacturing Operations Management
MRP	Material Requirements Planning.
MRP II	Manufacturing Resource Planning.
OEE	Overall Equipment Effectiveness.
OPC	Open Connectivity via Open Standards.
OPC-UA	OPC Unified Architecture.
OPC-Xi	OPC Express Interface, currently known as OPC .NET 3.0.
PCS	Process Control System.
PLC	Programmable Logic Controller.
SCADA	Supervisory control and data acquisition.
SOA	Service Oriented Architecture is an architecture level software design principle, which is based on independent and interoperable services.
SOAm	Service Oriented Architecture in Manufacturing.
SOAP	Simple Object Access Protocol.
SOP	Standard Operating Procedures
SQL	Structured Query Language.
TIMS	Tankhouse Information Management System, manufacturing execution system developed by Outotec.
T-SQL	Transact-SQL, is a proprietary extension to standard Structured Query Language.
UI	User Interface
VB.NET	Visual Basic .NET, is an object oriented programming language.
VSS	Volume Shadow Copy Service.
WBF	The organization for Production Technology, which were formerly known as World Batch Forum.
WCF	Windows Communication Foundation
WF	Workflow is a group of operations that can describe different mechanisms.
WIP	Work In Progress.
WPF	Windows Presentation Framework.
XML	Extensible Markup Language
XSD	XML Schema definition, see XML.

1 INTRODUCTION

1.1 Background

The purpose of this thesis is to research the current situation in the Manufacturing Execution Systems domain. Specific focus is in production dispatching with needs of customized communication to the devices in the field. The empirical research aims in implementation of production dispatching and scheduling tool to be used copper electrorefining plant.

The state in the production information system domain is disorganized today. Deployment of multiple custom-made software products has led to this situation and most of those separate products are completely unaware of the existence of the others. There are many already-made commercial products in the market providing total solutions, but in many cases those products don’t offer enough customization options. A vast amount of information is located in different databases and optimization could be done more efficiently if the clustered information were integrated together. Generic software design patterns could also be used with manufacturing execution systems to solve these integration and communication problems.

Production scheduling in copper electrorefining and electrowinning processes needs information from multiple data sources and a generic implementation of scheduling and production dispatching algorithm can be problematic, because of the differences in systems and interfaces used in different production plants. The implementation should take into account the differences in copper electrorefining and copper electrowinning processes, which both are copper electroplating techniques. Customer and end user conventions to operate systems the way they are used to is also a challenge when implementing new production dispatching systems using new technologies.

Manufacturing execution systems are at least partially considered production critical systems, making the demands of the reliability and robustness high. State of the art software technologies are not used in environments like these and one part of this thesis is to research how the latest innovations and design models are suited for manufacturing execution system environments.

1.2 Contribution

This work is done as a part of large scale development project. The scope of this thesis is the development of back-end systems for production dispatching operations. These back-end systems include design and implementation of database structures, actual dispatching engine, communication and integration with existing systems. For example all of the user interface components are out of the scope of this thesis.

The system to be implemented is the first of its kind when discussing about online production dispatching in copper electrorefining plant but some other offline scheduling tools are already made. The documentation and material about those old tools are properties of their original owners and thus cannot be used as a part of this thesis.

1.3 Research Objectives

Objective of this thesis is to increase knowledge about SOA and new workflow based programming tools and the different possibilities to use these technologies especially in integration projects. Objectives also include comparing different technologies and already made SOA and workflow products and evaluate those as a part of manufacturing execution systems. In addition implement automatic detailed production scheduling system for the case project and research possibilities to implement higher level plant wide scheduling system. The system to be implemented should be generic enough to be used in different production sites with varying number of controllable devices.

1.4 Thesis Contents

This first chapter is an introduction to the thesis containing information about research objectives and the definition of the scope. The second chapter is an introduction to manufacturing execution systems and other software related tools that could be used in the empirical research part. The third chapter is about the actual research problem and introduction to the environment in question. The fourth chapter is a description of the design and implementation of the actual system that is implemented as part of this thesis work. The fifth chapter contains the results and discussion of the empirical research. The sixth and final chapter concludes the thesis work.

2 THEORETICAL BACKGROUND

Theoretical background of this thesis mainly focuses on manufacturing execution systems (MES) in general with most important standards and technologies in the MES domain. There is also an introduction of commercial MES platform that can be considered as alternative for implementation in empirical part of this thesis.

2.1 Manufacturing Execution Systems

Manufacturing execution systems are in the key role when orchestrating information flows between high level ERP (Enterprise Resource Planning) systems and low level automation and control systems. Production dispatching is one of the production control functions provided by manufacturing execution systems. Production dispatching is considered to be the start of actual production process and execution of different activities along the whole production workflow. Modern MES suites provide tools to implement these production control functions in natural workflow format (Kashyap & Emerson 2010; LeBlanc 2002). These workflow tools can be combined with other recent software technologies like service oriented architecture (SOA) to build modern manufacturing information systems that can flexibly interact with other systems (Komoda 2006).

General approach to manufacturing operations management (MOM) is to divide manufacturing operations into five different levels defined by the ISA-95 standard. Then the actual tasks and responsibilities of different systems can be assigned to these levels. With common language and allocation of tasks to different levels the communication between different actors becomes more straightforward. Actors in manufacturing domain are commonly the manufacturer or producer, number of software and hardware solution providers and possible system integrators. (Gifford 2011; LeBlanc 2002.)

The role of system integration and ability to communicate with other systems is important in the modern manufacturing systems. Concepts of vertical and horizontal integration are very common when discussing about MES systems. Horizontal integration stands for integration within the same level of business or manufacturing operations hierarchy whereas vertical integration is communication and operations among the systems in different hierarchy levels. In MES, vertical integration goes up to ERP and other business oriented information systems and down to automation layer containing manufacturing equipment.

2.2 History of MES

Production line and machine automation have been a part of industrial production since beginning of 20th century and thus that layer is highly evolved. The higher level ERP-systems have also been under high development and use for many decades. This development has led to neglecting the middle layer, which is MES. Operations to move information between systems have been done mostly manually by operators and production planners. In the recent years there has been significant change and interest to develop more modern MES-systems. (Gifford 2011.)

Even today the production planning and other MES-layer activities are handled in many companies with so called office automation products like Microsoft Excel and Word (Scholten 2009, p. 1). This has been sufficient way of operating but new demands and business drivers have created need for sophisticated manufacturing IT solutions. The trends in manufacturing have also been drivers for MES-systems. Lean manufacturing and Six sigma methodology are widely adopted trends and as their part they have been drivers to develop MES-systems. Many of the manufacturing methodologies need constant tracking of production performance statistics and also ways to present these to different decision makers. Decision makers at different levels of operational and strategic hierarchies are interested in different kinds of reports and data. Subcontractors and suppliers can be a part of the global partner network and they also have the need to interact with production IT-systems. SOA in manufacturing (SOAm) is one solution for the need for flexibility (Buhulaiga 2007). New MES-products from different vendors are coming with SOA based solutions (Gifford 2011).

2.3 Manufacturing Execution System Standards

Different standards are in important role when implementing scalable and interoperable systems that are understood unambiguously by professionals. Standards based solutions also extend the life cycle of the system by allowing future enhancements to be added to the system.

Standards that support interoperability and integration usually include common characteristics. These characteristics could be the following: standard notion and other properties of the domain, information model, clear structure of the information, interfaces between different levels or systems, communication media, representation or markup language and also semantic information with definitions of different processes. Commonly one standard contains only a subset of those characteristics and other standards can be used to supplement the base standard (ISA 2011; Gifford 2011.)

There are large organizations behind different industry standards, who promote the preferred solutions in the MES domain. Examples of these organizations are the International Society of Automation (ISA) and Manufacturing Enterprise Solutions Association (MESA) International.

ISA is a nonprofit organization that develops standards for the automation domain, where MES is also interfacing. ISA also provides education materials and hosts conferences about new technologies for industry professionals. ISA has published over 150 standards and the organization has over 30,000 members. (ISA 2011.)

MESA is also a nonprofit and global community of solution providers, manufacturers and other industry professionals. MESA members are committed on improving manufacturing operations management practices with the latest technologies and industry proven best practices. MESA members can be individual persons or organizations and the membership types are divided into manufacturer and solution provider master categories. MESA publishes articles about the best practices and operates education programs but does not publish official standards like ISA does. (MESA 2011.)

2.3.1 The ISA-95 Enterprise Control Systems Standard

The ISA-95 is one of the most important standards in MES integration and it is highly used and adopted among different manufacturing companies (Gifford 2007). The ISA-95 provides the common terminology and the language used in the MES domain. The ISA-95 also defines the object models used to model the systems. By using the terminology and models the ISA-95 defines the boundaries of the MES domain and the standard interfaces between MES and higher level systems like ERP. (ANSI/ISA–95.00.01–2000; Cholten 2010).

The ISA-95 standard is divided into five parts, each focusing on a certain important aspect of system integration. The first part defines the common terminology and the object models that can be used when defining what information is needed in certain manufacturing operations and what information is transferred between systems and subsystems. The part two defines the attributes of the object models defined in the first part. The attribute definitions define in detail what information can be exchanged between systems or subsystems. The part three defines the operations and activities at MES-layer also known as the third layer, which will be defined below in more detail. The part three can be used as a basis for comparison of different implementations of the MES systems in a standardized way. The fourth part is still in progress but there are working drafts published. The purpose of the fourth part is to specify internal information and data contents in MES-layer more accurately than the part three does. The fifth part specifies the transactions between the third and the fourth layers. The fifth part is also known as the business to manufacturing transactions and the communication defined in the standards is two-way. The fifth part is also under development and is based on models and attributes defined in parts one and two.

The ISA-95 classifies systems to five different levels. The Figure 2-1 describes those levels in general. Level four is on the top of the domain hierarchy and in standard it is called the Enterprise domain. The Enterprise domain contains business planning and logistics operations. The boundaries between different levels are also defined in the standard. In many cases the actual boundaries lies inside distinct operations like scheduling. Scheduling for example can be classified to level four and also in level three operations, the difference being in granularity and timescale.

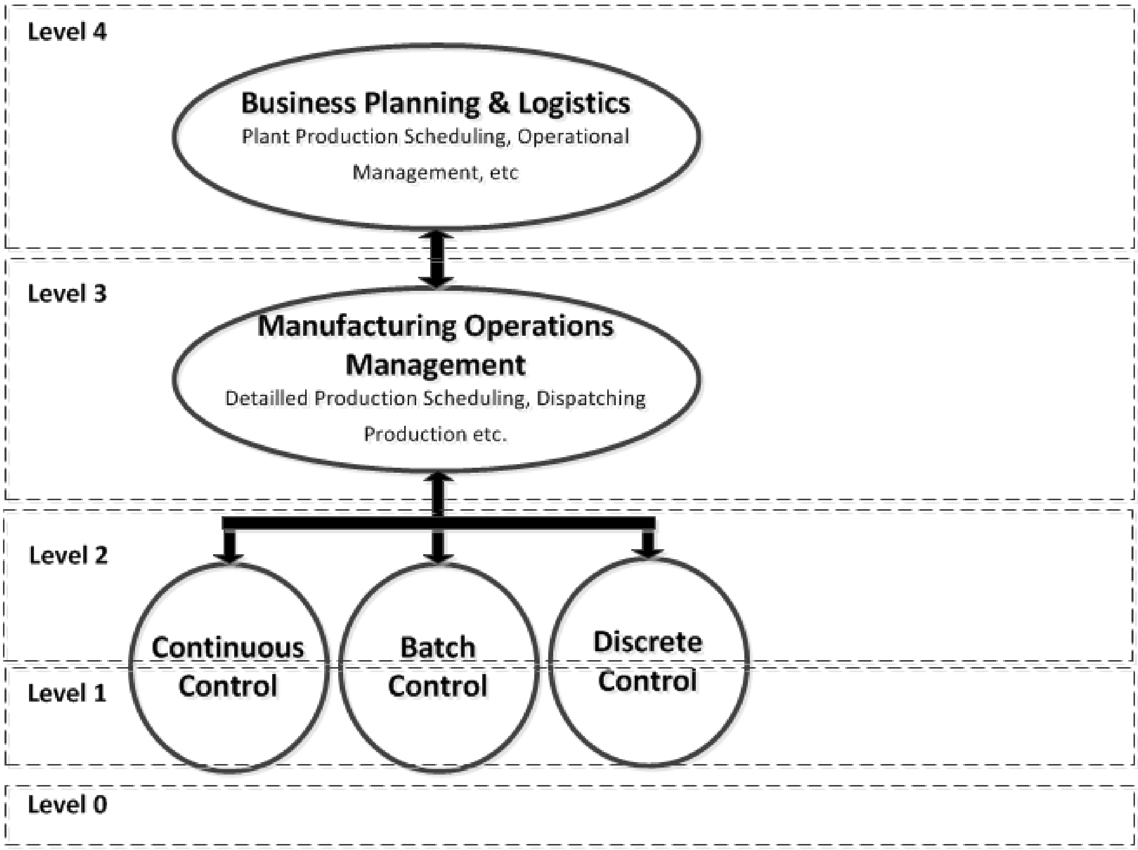

Figure 2-1 ISA-95 Domain Hierarchy (ANSI/ISA–95.00.01–2000, 18-21)

Main tasks in level four are higher level production scheduling, order processing, product cost accounting, delivery and shipping, inventory management, material and energy management and procurement. The timescale in level four is usually months or weeks. Levels three and below are part of control domain. The level three is also known as the MES level contains the manufacturing operations management (MOM) functions like detailed scheduling and production control in general. The MOM includes production, maintenance, quality and inventory operations. The timescale in level three is days, hours or in some cases minutes. Level two and below handles with the actual production and production equipment with the timescale ranging from minutes to milliseconds. (ANSI/ISA–95.00.01–2000, 18-21; Scholten 2009 16-18)

2.3.2 The ISA-88 Standard

The ISA-88 Standard defines operations that are below the scope of the ISA-95 standard. Those operations include structuring of a production process and defining the control of equipment. The main focus of the ISA-88 standard is in the batch processing but it can be applied also to discrete and continuous manufacturing. Like ISA-95 the ISA-88 consists of models and terminology and it can be applied to processes containing different levels automated operations. (ISA-88 2010.)

The ISA-88 standard is divided into five different parts, which are: "Batch Control Part 1: Models and Terminology", "Data Structures and Guidelines for Languages", "General and Site Recipe Models and Representation", "Batch Production Records" and "Implementation Models & Terminology for Modular Equipment Control". The fifth part is still in development. (ISA-88 2010.)

2.3.3 Other Manufacturing Execution System Standards

In addition to the standards introduced in previous chapters there are other more generic standards that could be used within the MES domain. These other standards include for example: regulatory standards, quality standards and other more generic software integration standards. These other standards can be used to supplement industry specific base standards in cases where the base standards do not provide comprehensive solutions. Communication standards are common supplementary standards, because for example ISA-95 does define what protocols are used when sending messages between systems (ANSI/ISA-95.00.01-2000). In certain industries regulatory compliance requires to keep track of the genealogy of the products (Gifford 2011). International Organization for Standardization (ISO) has published for example quality standards like the ISO 9000 series and environmental management standards, which could be taken into consideration when designing MES.

The ISA-95 standard does not define actual implementations for communication between third and fourth levels (ANSI/ISA–95.00.05-2007). Common approach to implementation of ISA-95 is to use Business to Manufacturing Markup Language (B2MML), which is an Extensible Markup Language (XML) based standard published by WBF. WBF was formerly known as the World Batch Forum, but currently they use only slogan “The Organization for Production Technology” (WBF 2011). Basically B2MML consists of XML schemas that define the data models structures and formats according to ISA-95 standard. The XML schemas are written in XML schema language (XSD), which defines sets of rules for validity of documents based on certain schema. (Brandl & Emerson 2008.) Common approach for communication using B2MML is the transfer of the XML documents using Web services with Simple Object Access Protocol (SOAP).(Gifford 2011.)

2.3.4 Production Scheduling in MES Standards

Scheduling is generally divided into multiple layers defined by the scope and the time horizon used. The different levels of scheduling mentioned in the ISA-95 standard are presented in the Table 2-1. Detailed production schedule and production dispatch list are activities of the MES domain and all others are in the scope of layer four systems like ERP. The ISA-95 standard acknowledges also long time horizon scheduling activities but does not give exact definitions for those. Business plan and demand plan are elements defined in the APICS (the Association for Operations Management) dictionary. (ANSI/ISA-95.00.03-2005.)

Table 2-1 Hierarchy of schedules and scheduling activities in ISA-95 and APICS standards (ANSI/ISA-95.00.03-2005)

Scheduling Level	Scope	Time Horizon	Activity definition
Business plan	Product line	Very long	Not in the scope
Demand plan	Product	Long	Not in the scope
Production plan	Product	Long	Production forecasting activity (ISA-95.01)
Production Schedule	Site/Area, Product	Medium	Production scheduling activity (ISA-95.01)
Detailed production schedule	Work center, Product/ Intermediate	Short	Detailed production scheduling activity (ISA-95.03)
Production dispatch list	Work center (element), produced item	Very Short	Production dispatching activity (ISA-95.03)

Highest level of scheduling defined by the ISA-95 standard is the production plan activity. Output of production plan activity is a master production plan (MPS). Target of a MPS is a single site or facility and a MPS is based on site capabilities including manufacturing capacity, distribution capacity and capital capacity. MPS can be used to create production schedules with a material resource planning (MRP) or other enterprise resource planning (ERP) related activities. According to ISA-95 standard a production schedule is created in layer four systems based on information gathered in MES layer (see Figure 2-2). Operations requests based on the actual production schedule and product definition are transferred from layer four systems to MES and based on production requests a detailed production scheduling activity is executed. Detailed production schedule defines what resources are used during the production. (ANSI/ISA-95.00.03-2005; (ANSI/ISA-95.00.01-2000.)

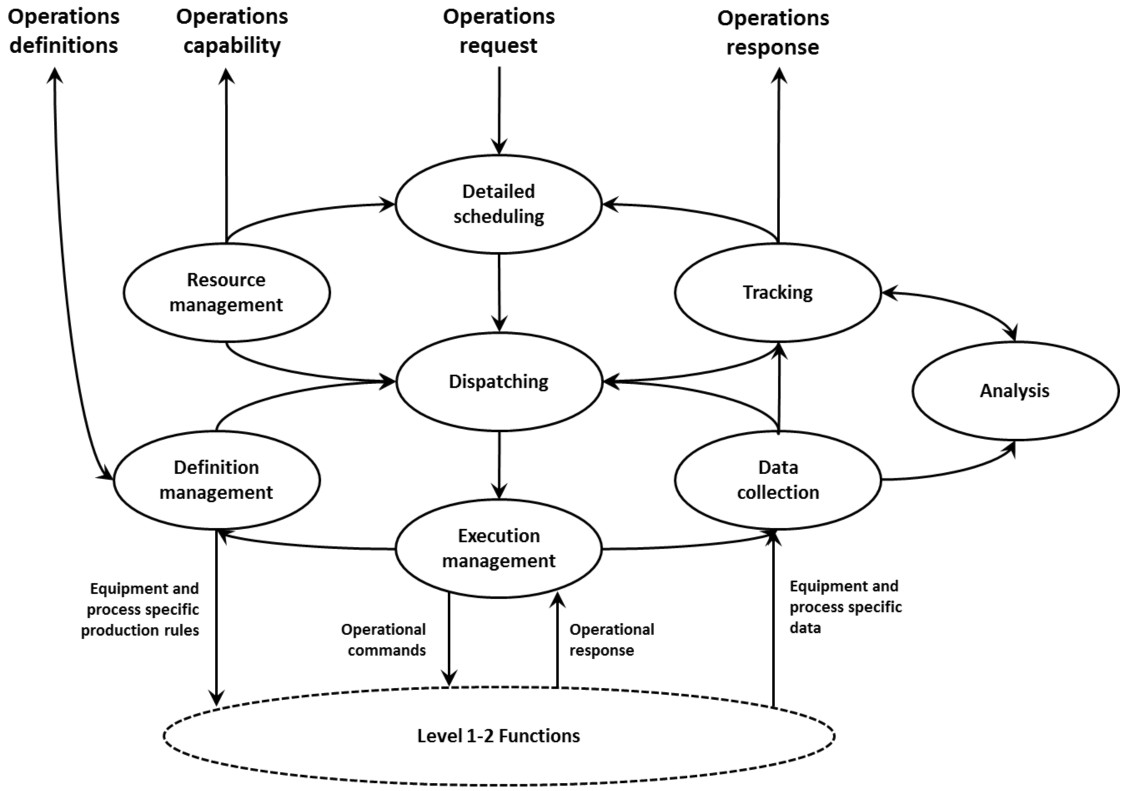

Figure 2-2 ISA-95 Generic activity model of MOM (ANSI/ISA-95.00.03-2005)

Production Dispatch Activity

Production dispatching activity is defined in the ISA-95 standard by using a collection of activities that manage the flow of production. The production dispatching activity has interactions with the equipment and personnel models; more precisely it focuses on work center or work unit level. Production dispatching activity can be a part of any type of production, which are batch, continuous, repetitive or discrete and equipment or storage movement. Examples of production dispatching activities presented by the ISA-95 standard (ANSI/ISA-95.00.03-2005, p. 40) are:

a) Scheduling batches to start in a batch control system.
b) Scheduling production runs to start in production lines.
c) Specifying standard operating condition targets in production units.
d) Sending work orders to work centers.
e) Issuing work orders for manual operations.

Production dispatching activity is situated after detailed production scheduling as can be seen from the Figure 2-3. The production dispatching uses detailed production schedule as its input.

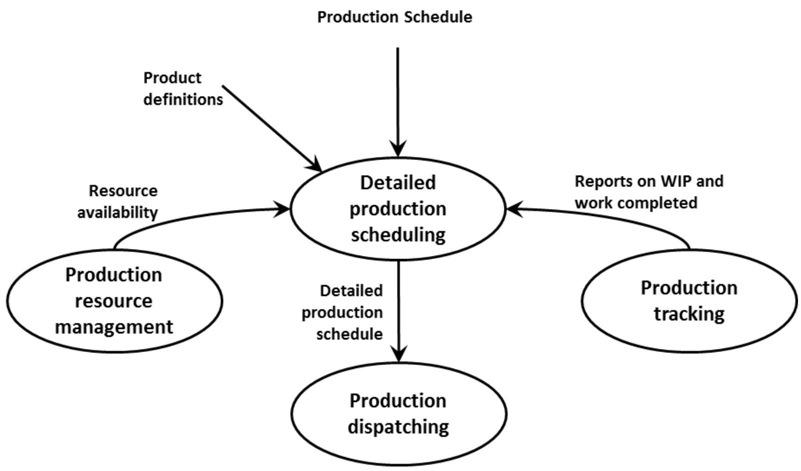

Figure 2-3 ISA-95 detailed production scheduling activity model interfaces (ANSI/ISA-95.00.03-2005)

Production Dispatch List

Production dispatch list is a set of production work orders with order elements assigned to the specific work centers or units. The production dispatch list is an output of the actual production dispatching activity defined by the part three of the ISA-95 standard. Dispatched work may be machine setup, grade change switchovers, equipment cleaning, run rate setup, or production flow setup. (ANSI/ISA-95.00.03-2005, p. 41.)

2.4 Software Technologies in MES

The basic and also widely used software technologies in the MES domain include primary office automation products. In many cases Microsoft Excel is used as a standalone solution. Also many more specialized reporting systems often provide import/export functionalities through Excel worksheets. (Gifford 2011). Those office automation products and their functionalities can be extended by using macros or other script based programming methods.

2.4.1 Service-oriented Architecture

Service-oriented Architecture (SOA) is an architecture level software design pattern, consisting of independent and interoperable operations and processes. The idea of independent services is that they can be flexible and the overall architecture is highly scalable. Basic components of SOA are (Gifford 2011):

- Services
- Service bus
- Services registry
- Message routing
- Service choreography

There are platform specific implementations of technologies that support SOA principles, for example in .NET Framework the Windows Communication Foundation (WCF) can be used as a basis for the SOA applications.

In the MES domain SOA can be used as a basis for software infrastructure that can be extended by custom components at the same time when the core system is standards based MES engine. By using the SOA the integration interfaces are common between different custom components and the overall extendibility is not limited (Komoda 2006).

2.4.2 Unified Modelling Language

The Unified Modelling Language (UML) is a graphical modelling language developed and standardized by Object Management Group (OMG). The latest version of UML is 2.4.1 and it consists of 14 different diagrams which can be used to analyze, design and implement object-oriented processes or systems. Originally the UML was designed to be used for system and software development modelling but the UML is generic enough to be used also for modelling for example business or production processes. (OMG 2011.)

UML diagrams can be used to represent structural information, behaviors and interactions. Diagrams included in the specification are presented in the Table 2-2. For example activity or sequence diagrams can be used to represent production workflows and class diagrams are used when designing software components.

Table 2-2 UML Diagrams and Categories (UML 2011)

Diagram Category	Diagram Name
Structure Diagram	Class Diagram
	Component Diagram
	Object Diagram
	Profile Diagram
	Composite Structure Diagram
	Deployment Diagram
	Package Diagram
Behavior Diagram	Activity Diagram
	Use Case Diagram
	State Machine Diagram
Interaction Diagram	Sequence Diagram
	Communication Diagram
	Interaction Overview Diagram
	Timing Diagram

2.4.3 OPC Integration

OPC, originating from the words Object Linking and Embedding for Process Control, is a series of standards specifications about open communication protocols used in industrial automation (OPC Foundation 2013; OPC Task Force 1998). OPC has become the de facto standard in its field and therefore many integration solutions are leveraging OPC in some way. Today OPC Foundation, who is responsible for developing the OPC standards, is manifesting the new meaning of OPC which is open connectivity via open standards (OPC Foundation 2013).

OPC is used for example in Supervisory Control and Data Acquisition (SCADA) systems to exchange process data between Programmable Logic Controllers (PLC) and Human Machine Interface (HMI) systems. OPC can be used to transfer real-time data or to get historical data. (OPC Task Force 1998.)

Prior to OPC there was a need for device specific drivers for every software component that had to communicate with a hardware device. When OPC protocol is used there is only need for one device specific driver which provides the standard OPC server interface and which is usually provided by the device manufacturer. Application developers who use devices need only to implement the OPC client interface to communicate with the OPC servers. (OPC Foundation 2013.)

The first OPC standard, the Data Access Specification was originally based on Microsoft’s COM/DCOM (distributed component object model) technologies. The first DCOM-based version of the OPC is still the most widely used and supported technology, but the new specifications offers more open and Microsoft independent solutions like OPC-UA (OPC Foundation 2013). Even though OPC has simplified the development of automation software and communication there are some problems and drawbacks originating mostly from the use of DCOM technologies (Todorov 2007). The configuration of DCOM is considered very difficult and the security is questionable in many cases (Davis & Zhang 2002). The OPC-UA tries to solve all these problems, which were in older versions of OPC specifications.

2.4.4 Database Systems and Integration

There are different solutions for storing information in the MES and ERP systems. Different kinds of information set certain requirements for performance. Data historians are usually used for storing real-time process data, which is usually time-based. All the greatest MES providers offer their own process data historians, like GE Proficy Historian, Wonderware Data Historian and Simatic IT Historian. These databases are optimized for storing data of thousands or event hundreds of thousands process signals changing at a high rate. This rate can be in millisecond scale and the amount of the sequential data can be very high. Data historians usually provide ways to create calculations and filters for the data being stored. Calculations provide a way to save the relevant data in a more efficient form and decrease the need for computing power during the data analysis or the reporting phase. (Jankowski et al. 2011.)

Even with the sophisticated data analysis and calculation tools the process data in historians is rarely directly useful for end-user reporting in MES systems. Common approach is to use an actual MES database to store higher level MES information. Usually parts of this information are based on historian data. But also the MES database could contain information about the actual process and equipment models and the rules or algorithms that are used to aggregate the data to calculate for example some key performance indicator (KPI) values. To calculate these values the MES system might need information from multiple data sources and source commonly being the data historian. Common platform for these MES databases is a relation database like Microsoft SQL Server (Snoeij 2012).

2.4.5 Microsoft .NET

Microsoft .NET framework is a software component library for Microsoft Windows operating systems supporting wide range of different programming languages, including C# and VB.Net. .NET Framework consists of two major parts: class libraries and runtime environment called Common Language Runtime (CLR). The first version of the .NET Framework was published in 2002, now the latest stable version is 4.5, which was released on August 15, 2012. Windows Workflow Foundation (WF) is a part of .NET Framework and was introduced in .NET 3.0. The other well-known and widely used parts of .NET Framework are Windows Communication Foundation (WCF) and Windows Presentation Foundation (WPF). (MacDonald 2010.)

Windows Communication Foundation

WCF unifies all common communication technologies available in .NET Framework. WCF is also the standard way to build service oriented applications with .NET programming languages. It has features compatible with Web Service specifications and thus allows platform independent communications. Web Service specifications are published by W3C and have implementations in many different programming languages. WCF provides tools for both server and client side communication handling and allows multiple standard and proprietary security features. A subset of WCF features are also implemented in Microsoft Silverlight. (Chappel 2010.)

Windows Workflow Foundation

WF provides workflow engine that contains common features for workflow execution and scheduling. Workflows created using WF can be exposed as a WCF service and thus invoked from external systems. Workflows are considered long running sequential activities, but exceptions are possible. The WF contains a graphical design tool, which can be hosted in other .NET programs. The design tool allows programming performed by people without programming experience. The rehostability can be leveraged by implementing custom workflow platforms where workflows can be used to control the execution flow. (Chappel 2009.)

2.4.6 Reporting in MES

One of the most important features of MES systems is the reporting. Reporting provides information about the production operations. Commonly the information is about the output of the manufacturing operations but there can also be estimations about future in the reports. Reporting is a way to evaluate key performance indicators like OEE (Overall Equipment Effectiveness) (Phillips & Both 2010). The actual information collected to the MES system is generally useless if it can't be accessed as a form of clear reports.

Data linking and genealogy reports provide a powerful tool to track production chain from raw materials to finished products. These kinds of genealogy reports might be regulated and demanded by laws in certain manufacturing domains. Standards and certificates of operations quality might also demand certain types of reports. (Suomen Automaatioseura ry 2001.)

Reporting can be implemented in different levels to follow for example ISA-95 hierarchy. Reporting can be dynamic or scheduled to be run on certain intervals. The scheduled reporting is common approach in many commercial MES products (Simatic IT 2011; GE Intelligent Platforms 2010B;

2.5 Manufacturing Execution System Platforms

There are multiple widely used MES platforms in the market today and the following chapters will introduce a one of them in more detail. The platform to be reviewed was chosen because of the request and preferences of the customer. There are at least over 160 commercial MES products (Logica 2011; Snoeij 2012) that could be reviewed but for the scope of this thesis it is not possible or reasonable to review other systems except the one the customer already uses. An up to date comparison and review of over sixty common MES platforms can be acquired from MES product survey 2012 made by Logica (Snoeij 2012).

2.5.1 MES Products and Custom Implementations

Many software consulting companies have implemented their own MES products and also major ERP-system providers have objectives to enter also into MES market. The offerings from different vendors range from simple reporting applications to full production management suites bundled with ERP and automation layer products. (Snoeij 2012.)

There are many custom implementations of MES systems in use as stated in the introduction chapter of this thesis. In cases where MES system is shipped as a part of plant or production system, there might be significant cost benefit to implement fully custom MES. In commercial MES platforms there are license fees that are usually directly proportional to the number of delivered systems (Snoeij 2012). Custom systems are usually owned by the company and therefore there are no license fees but of course the maintenance and life cycle management can be more expensive compared to commercial products.

Custom implementation usually needs more skilled and technically aware people during implementation and possible modification phases. The custom systems should be implemented with standards in mind and those systems should provide standard based interfaces at least to external connections, because of the inevitable expansion and change in production environments (Gifford 2011.)

2.5.2 GE Proficy, an Example of MES Product Family

According to the customer request the General Electric Intelligent Platforms Proficy product family was selected for more detailed review. General Electric Intelligent Platforms Proficy product family includes wide range of industrial automation products. Proficy product family is divided in two categories: automation software and operations management software. Automation software category contains HMI/SCADA products and OPC-connectivity software among some other software products to support operators work in plant and also provide communication layer for higher level MES-systems. The operations management software category includes the actual MES platform, Proficy Plant Applications. Proficy Historian is a data historian that supports real-time data collection from multiple data sources. Proficy SOA and Workflow are also parts of Operation management category.

Proficy HMI/SCADA iFIX and Cimplicity are supervisory control and data acquisition products, which are used to create operator displays that can be used to control and monitor industrial processes. Proficy I/O Drivers contains OPC-server software that can provide connectivity to wide range of different PLC’s and automation systems.

Proficy Plant Applications consists of four different performance modules, which are: batch analysis, efficiency management, production management and quality management.

GE Proficy Workflow

GE offers a product called Proficy Workflow for workflow functions. The Proficy Workflow is a large scale MES and integration product which operates on top of Proficy SOA. The framework is highly standardized and based on ISA-95 and ISA-88. Proficy Workflow operates mainly on layer three in ISA-95 scale but its operations can be extended all the way to process cell level two by using the ISA-88 extensions of Proficy Workflow. Proficy Workflow can also reach to ERP level functionalities by leveraging the extendibility provided by custom service providers. Models in Proficy Workflow can be divided into two different categories, which are process models and resource models. (GE Intelligent Platforms 2009.)

Resource models describe physical resources of an enterprise. Physical resources can be personnel, material and the actual equipment model. The equipment model is used to model the whole enterprise with desired granularity. Personnel model can be used to describe the employees of the organization. Material model describes the properties and definitions of materials used in production. (GE Intelligent Platforms 2009.)

Equipment hierarchy model used in Proficy Workflow is presented in Figure 2-4. In the equipment hierarchy model the blue objects are based on ISA-95 standard and the orange objects are extensions from ISA-88 standard. (GE Intelligent Platforms 2009.)

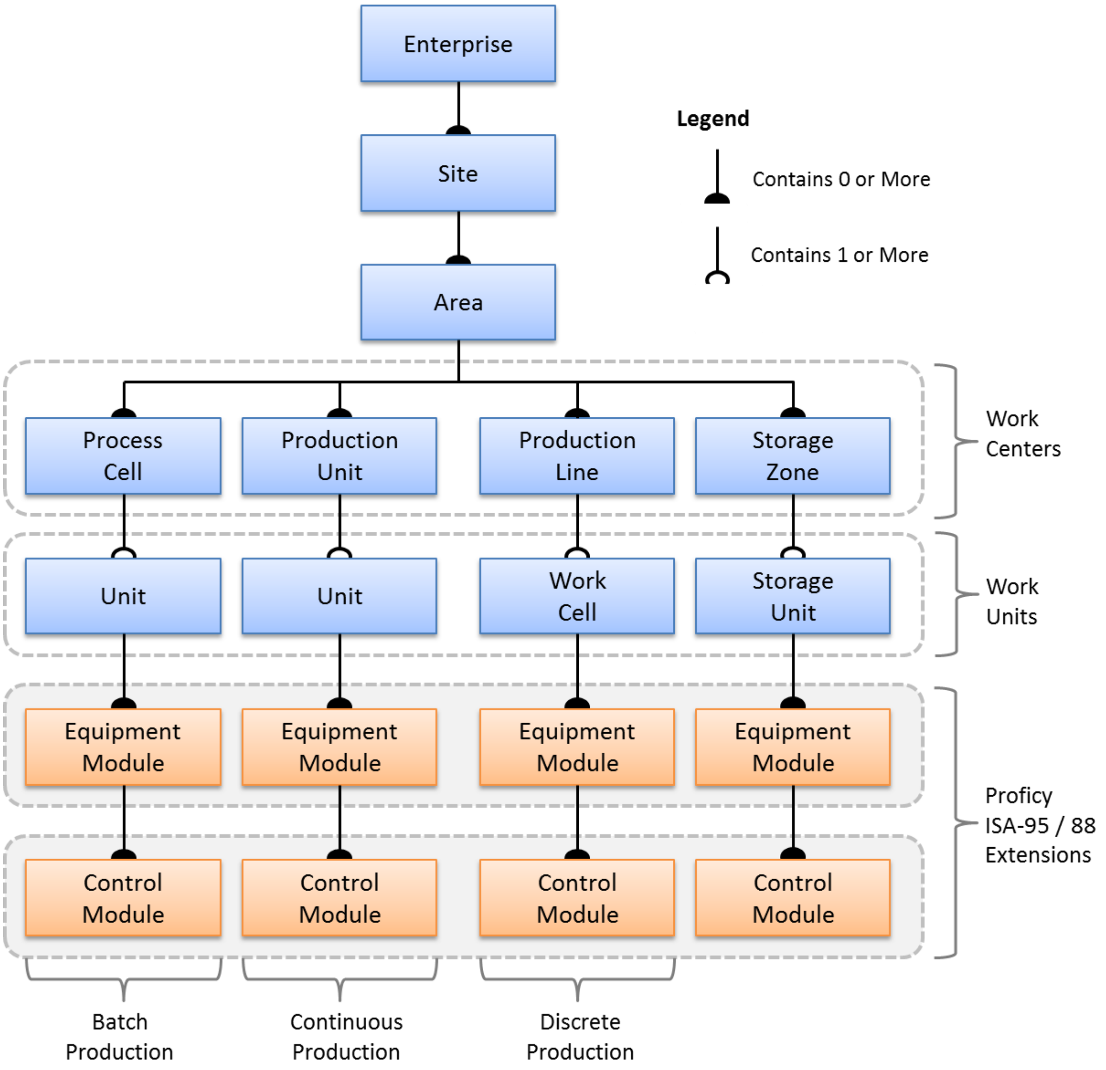

Figure 2-4 ISA-95 Equipment Hieararchy Model with ISA-88 Extensions

The model contains different equipment for different production types, which are batch production, continuous production and discrete production. Equipment in the model can have custom properties, which each has the standard attributes defined in the Table 2-3 ISA-95 core object attributes.

Table 2-3 ISA-95 core object attributes (ANSI/ISA–95.00.01–2000

Attribute	Description	Example
ID	A unique string identifying the property	2011-01-12-UNIT2-Yb21
Description	Additional description of the property	Number of anodes inside unit 2.
Quantity	The quantity of the property	3
Quantity Unit Of Measure	The unit of measure in which the quantity is measured	anodes

Process models are related to production processes and are also based on ISA-95. Process models in the Proficy Workflow use naming conventions from new ISA-95 standard drafts, those new names and their old counterparts are described in Table 2-4. The basic difference between old and new naming conventions is that the new ones use generic term *work* instead of term *production*. The new terminology aims to provide support for other work types than production; these other types can be for example maintenance, quality or inventory. (Gifford 2011.)

Table 2-4 Old and New S95 Terminology (ANSI/ISA-95.00.01-2000; GE Intelligent Platforms 2009, p. 10)

Old Name	New Name
Process Segment	Work Process Segment
Product Segment	Work Definition Segment
Product Production Rule	Work Definition
Production Request	Work Request
Production Response	Work Response
Production Schedule	Work Schedule
Production Capability	Work Capability
Production Performance	Work Performance

The main components of the process models of Proficy Workflow are work request, work definition, work definition segment, work process segment and work data class. The main components are separately configurable, which can be seen from the Figure 2-5.

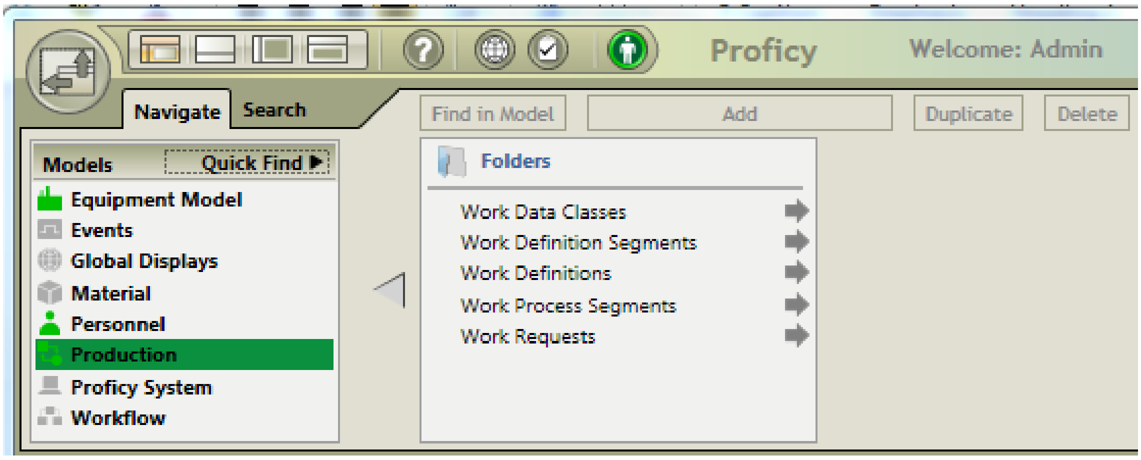

Figure 2-5 Main components of production model in Proficy Workflow

Implementation of the ISA-95 process models in Proficy Workflow is described in Figure 2-6. The sequence for creating a production or other work process goes as follows:

Table 2-5 Sequence for creating work process in Proficy Workflow

1. Create Work Data Classes
2. Create Work Process Segment and associate desired Work Data Classes to it

3. Create Work Definition Segment based on Work Process Segment created in the step 2
4. Create Work Definition and associate desired number of Work Definition Segments
5. Work Requests can be now made based on this work definition

In the Proficy Workflow the general idea is to create generic work process segments for example to a certain machine production unit and then create separate work definition segments to define more specific information. Work definition segments can be used for example to produce different products with different work parameters and resource requirements without the need of rewriting the general requirements set by the work process segment.

The work data is the class definition for the work response data. In Proficy Workflow the work data can only be associated with a work process segment. Each work definition segment is associated with only one work process segment, while single work process segment can be associated with multiple work definition segments. (Gifford 2011, pp. 148-164; GE Intelligent Platforms 2009.)

Work definitions are created by defining the type of the work definition, where the type can be: production, maintenance, quality, inventory, mixed or other. Related production plant can be also set if known. The actual definition is created by combining zero or more work definition segments to the work definition.

A work request is always created against a single work definition. Work requests contain for example scheduling information and have a status. The status can be one of the following: draft, unscheduled, pending, running, completed, failed, cancelled, unknown. Work response is the output of the execution of a work request. A work response contains separate segment responses with their work data. Work responses are part of the completed work request.

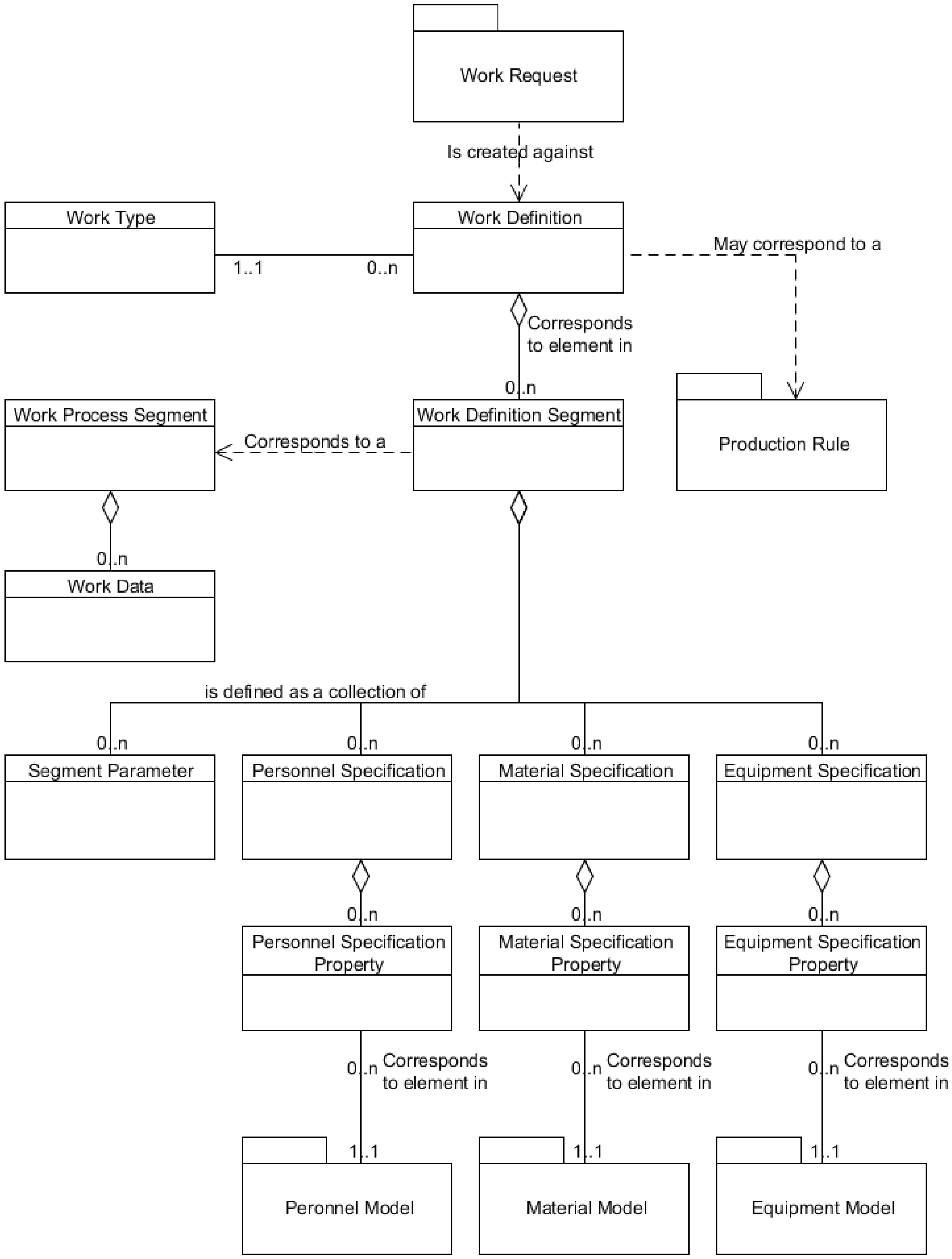

Figure 2-6 Implementation of ISA-95 process models in Proficy Workflow

Proficy Workflow also provides a configurable event and schedule engines that can be used to trigger workflows. Events can be time based, linked to different data sources or published by custom service providers. Schedules are commonly triggered by events and schedules can also contain input parameters that can be passed to workflows.

3 THE RESEARCH PROBLEM AND THE ENVIRONMENT

This chapter introduces the case company and the research problem. The production environment and processes related to the research problem are introduced in this chapter also.

3.1 The Case Company

The case company and the customer for this project is Outotec Oyj. Outotec Oyj is a Finnish company providing technological solutions for multiple industries, including metals and mineral processing and utilization of alternative energy sources. Outotec operates globally and its sales in 2010 were nearly a billion euros. A part of the Outotec's solution offerings are tankhouse processes and related automation and information systems. The tankhouse technology can be used for nickel, zinc as well as copper production processes.

3.2 Copper Electrolysis Processes and the Target Environment

Electrowinning and electrorefining are electroextraction techniques for purifying copper with the help of electric current. The production facilities using these electroextraction processes are called tankhouses. The basic features of a tankhouse crane will be also reviewed in this chapter, because cranes are in important role in the scheduling point of view.

3.2.1 Copper Electrorefining Process

Copper electrorefining (ER) is used to produce over 99.99% pure copper. In the electrolysis process there are impure source material blocks called anodes, which are about 99% pure copper. During the process cycle the pure copper is deposited to the surfaces of the cathodes. Anodes and cathodes are in electrolyte solution, which is an acidified aqueous solution containing different additives. Anodes and cathodes are laying in production cells. Simplified structure of a single cell is described in Figure 3-1, where brown objects are anodes and grey ones are cathodes. Cells are divided into sections, each section containing usually from 15 to 30 cells and the whole cell area usually contains from 10 to 30 sections.

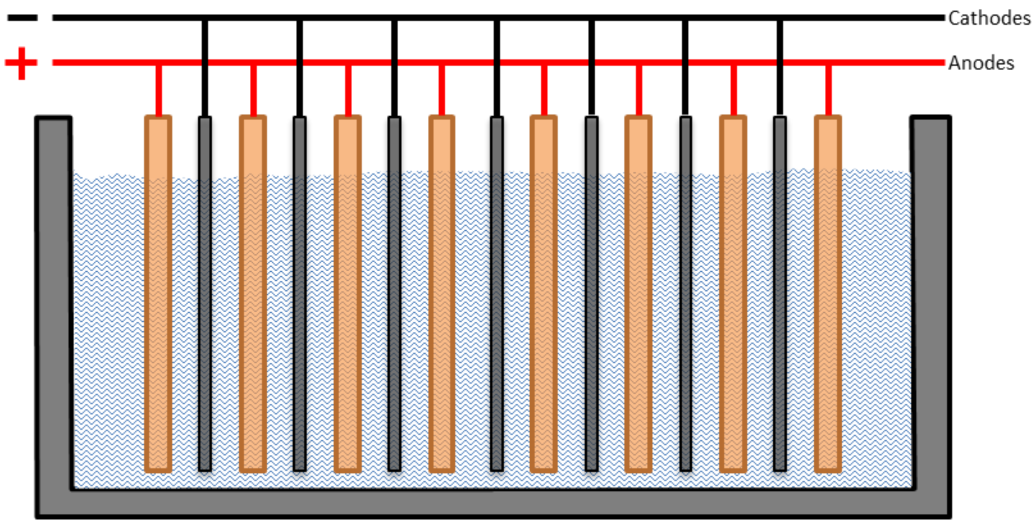

Figure 3-1 Simplified structure of a cell, the brown ones are anodes and grey ones are cathodes.

A cell farm area is only a one part of a production plant called tankhouse, which is used also when talking about EW production plants. Simplified structure of a tankhouse is presented in Figure 3-2. The common machines in tankhouse are anode preparation machine (APM), cathode stripping machine (CSM), anode scrap washer (ASW) and cranes. Outotec stripping machines are called full deposit stripping machines where the abbreviation is FDSM. Usually the tankhouse contains one or two cranes, which operate on different section areas called aisles. The tankhouse may also contain multiple APMs and CSMs.

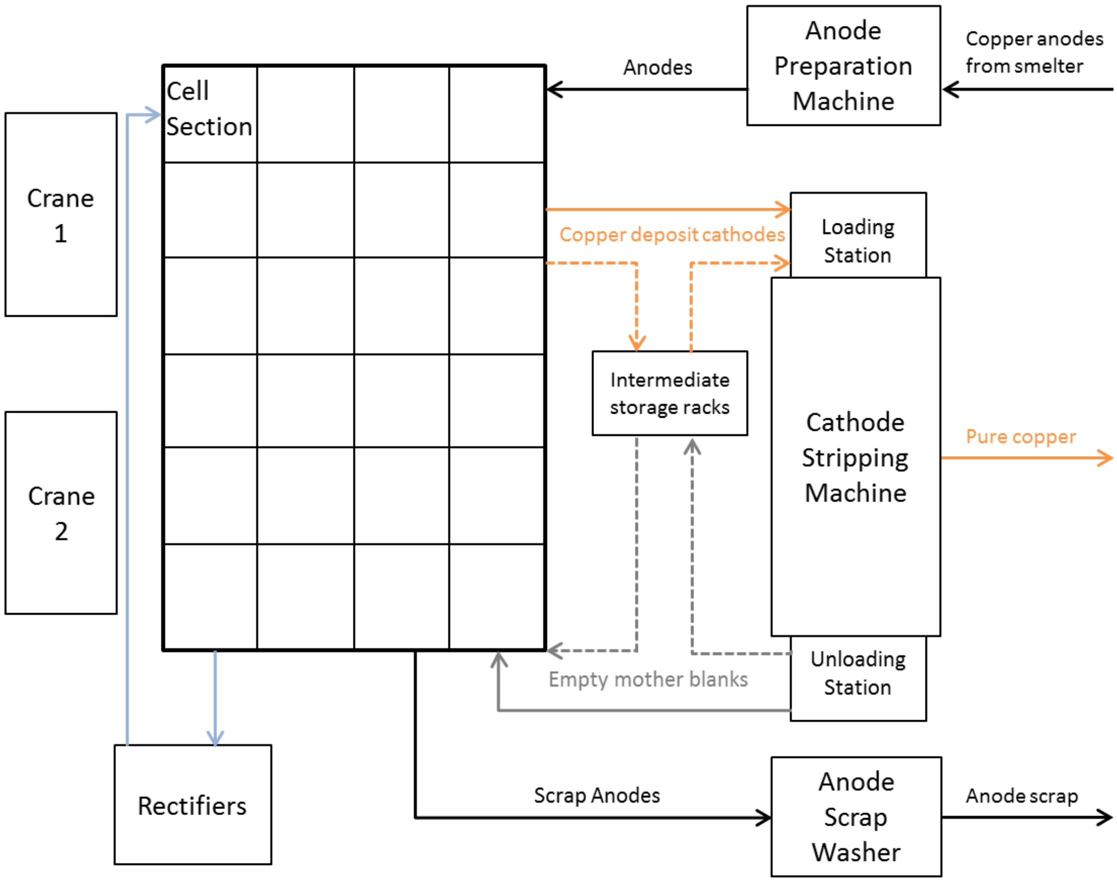

Figure 3-2 Simlified structure of a ER tankhouse

The cranes are in the key role when orchestrating the operations of a whole tankhouse, because all the material and object transfers are done by the cranes. The other machines have conveyors so that the cranes can load or unload the machines.

When the process is initially started the cranes are used to fill the cells with anodes from APM and with empty mother blank cathodes from CSM. After each cathode cycle, the copper deposit (CD) cathodes are transferred to CSM by the cranes. And after the CSM has stripped the CD off the cathodes, they are transferred back to empty cells. The transfer of CD cathodes or mother blanks can go through intermediate storage racks, which can be used for example as a buffer. After each anode cycle, the anodes are consumed and the left overs, called scrap anodes, are transferred to ASW by the cranes. The ASW is used to wash acidic solution off from the scrap anodes. The solution might also contain precious metals, which should be collected. The anode scraps from ASW are usually sent back to smelter. (Schlesinger et al. 2011; Hartman 1992 p. 2255; Young et al. 2008 pp. 357-370).

The production is turned on or off by using bypass switches. Usually in ER tankhouse there are bypass switches for each cell section. When the cells are harvested the current is usually cut off or bypassed from the whole cell section. This means that all the cells in the cell section are not in production for the whole time of the harvest of that section. In ER process the production cycles are tracked on per cell section basis, because of the bypass switch layout. (Schlesinger et al. 2011.)

The tankhouse has many other utilities, which are not described here in detail. Solution process utilities and other auxiliary processes are important for the ER process but not in the scope of this thesis. The process itself is depending on multiple factors, which are for example the rectifier currents, electrolyte temperature and amount of different additives in the electrolyte solution. (Schlesinger et al. 2011.)

3.2.2 Copper Electrowinning Process

The copper electrowinning (EW) is an electroextraction technique that is used in commercial copper refineries. In the EW process the copper is dissolved into electrolyte solution, from where it is extracted to the surface of cathodes. The process is based on electric current that passes through the inert anodes and the solution to the cathodes. The EW tankhouse structure presented in Figure 3-3 is quite similar to ER structure, but the EW tankhouse is missing APM and ASW. The cells are not grouped and thus the process is controlled per cell basis and production cycles are also counted for each cell separately. (Schlesinger et al. 2011.)

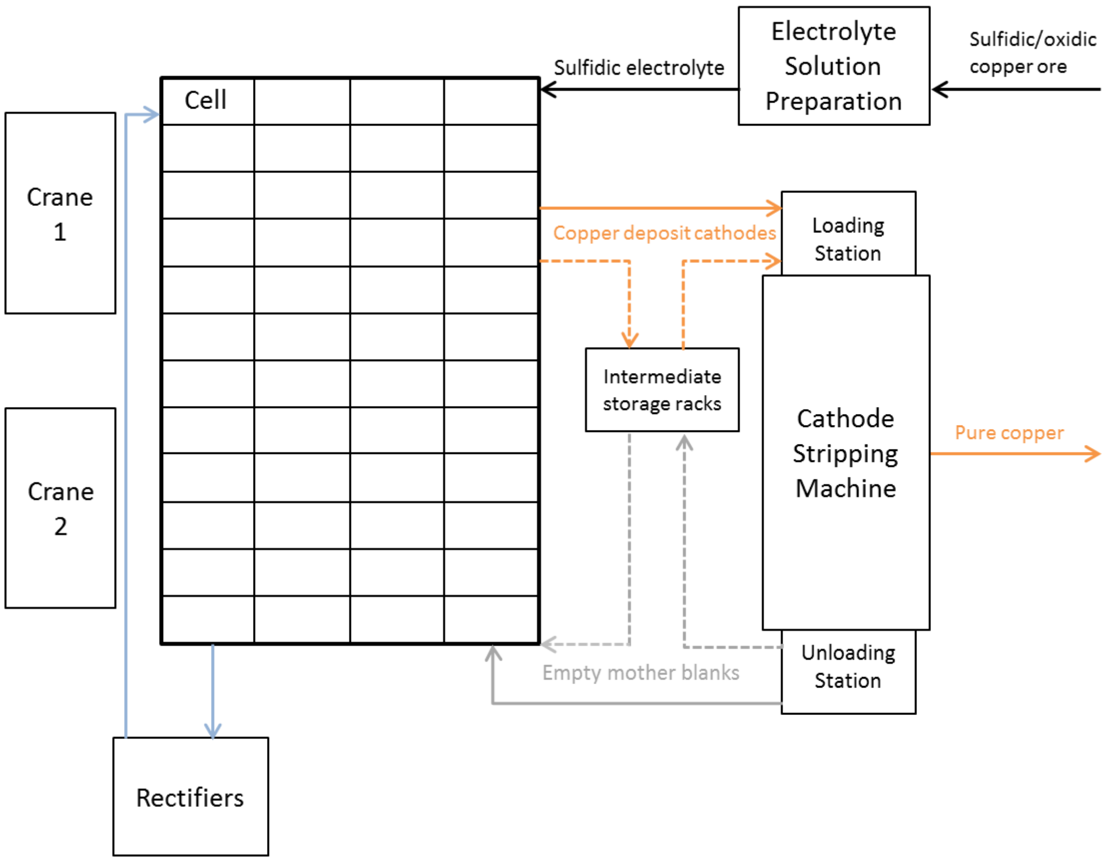

Figure 3-3 Simlified structure of a EW tankhouse

The inert anodes are passive and are unloaded only during maintenance, which takes place around every six months. There can also be few EW process cells in ER tank-houses to control the amount of dissolved copper in the electrolyte solution. (Young et al. 2008 pp. 357-370.)

3.2.3 Tankhouse Cranes

As stated previously in 3.2.1 cranes are in important role when orchestrating the production in a tankhouse. Typically tankhouse crane is of type overhead travelling crane containing a bridge with a trolley and a hoist. The bridge travels on rails attached to the tankhouse wall structures and the trolley is a unit travelling on the bridge rails. Trolley usually contains a cabin for the operator. The bale used to crab the anodes and cathodes is attached to the main hoist which is also part of the trolley.

Usually the tankhouse cranes are semi-automatic or fully automatic and the controlling can be done with a remote control or from operators panel located in the crane cabin. Cranes are usually controlled with PLC's and the PLC's are usually connected to the plant wide fieldbus that can be used to do the interlocking and other communications between the systems. There are no standard ways to give the cranes any production tasks or commands from external systems. The optimization of the movement from place A to B in a tankhouse is done in the crane PLC.

3.3 Scheduling of Copper Electrolysis Processes

The ER and the EW processes are different and thus the scheduling of those processes differs also. The first implementation will be for ER plant. This chapter will contain more detailed description of the ER process and for the EW process the differences to ER will be listed to consider future implementations.

The main measurable advantages that can be achieved with improved scheduling is a higher time efficiency for the whole plant and optimized growing times for cathodes. The waste percent can be also improved by optimizing the cathode growing times. The time efficiency for a cell section is the ratio between the active production time, also known as the laying time, and the calendar time. The scrap percent is the ratio of initial anode weight and residual anode weight after the production cycle. The cathode growing time is the actual time the cathodes have been laying in the cell and the current has been going through the cathodes and anodes. There are an optimal growing time for cathodes which depends on multiple parameters of the process. The most important parameters being the cathode laying time and the current density.

The time efficiency is decreased every time there are no anodes or cathodes in the cell or the current is not going through the anodes and cathodes. Because of the common layout of the electric circuits in an ER tankhouse there is a bypass switch only for every cell section. This means that all the cells in a section are bypassed even when only one of them are under harvest. If the optimal growing time is reached it might be necessary to bypass the section to achieve the optimal quality and output of copper. If the scheduling is done improperly it might be so that the crane is already harvesting an another section or is otherwise unable to start the harvesting immediately after the section in question is ready for harvesting, which also decreases the time efficiency. On the other hand in the ER process if the section is bypassed before it has reached the optimal growing time the residual anode weight is higher, meaning that the scrap percent will increase.

The influences of the upgraded scheduling can be seen from the key performance indicators introduced in the previous chapters. Especially the time efficiency should be monitored.

3.3.1 Scheduling of Copper Electrorefining Process

The scheduling of the copper electrorefining process is also the optimization of the grow times of cathodes and on the other hand optimizing the time how long anodes are kept in a cell. The most important machine to be controlled for the scheduling point of views is the crane.

In the copper ER process the major goal is to achieve optimal output of copper with minimum energy consumption. It is clear that there are variations of operating procedures how the scheduling is done and executed. Current way to do the scheduling in the copper electrorefining plants is offline calculations without feedback from the automation systems. The actual schedule can be based on different cycles which are parts of the process. Generally there are two different kinds of production cycles, anode cycle and cathode cycle. In different production plants there are different ways to calculate these and the actual full production cycle can consist of different number of these cycles.

When implementing generic solution for scheduling it is essential to create algorithms that can be customized to support customer preferences and various customer specific environments. On the other hand, this system could be one important driver to start production process re-engineering if end users were to test other ways to create a schedule (Wu et al. 2004). One of the challenges is to get the end user to adopt the new improved operating procedures (Saha et al. 2000). Two different ways to divide the schedule are presented in the following paragraphs (Stelter & Bombach 2004; Young et al. 2008 pp. 357-370).

Harvesting schedules

A general way to divide scheduling is to create a schedule consisting of three cathode cycles which equals to one anode cycle. Typical length of a cathode cycle in this model is around seven days, which yields to anode cycle of length 21 days. The Figure 3-4 shows the three cathode cycle model and the operations in relation to different cycles. General harvest for one section according to this model goes as follows: after first cathode cycle the cathodes of the section are harvested, after second cycle the cathodes are also harvested and after third cathode cycle both anodes and cathodes are harvested. There are also different conventions to replace the anodes. The term harvest can have different meanings and in case of cathode harvest it means that cathodes with copper deposit are transferred from cell to a cathode stripping machine and new blank cathodes are transferred to the same cell. (Schlesinger et al. 2011.)

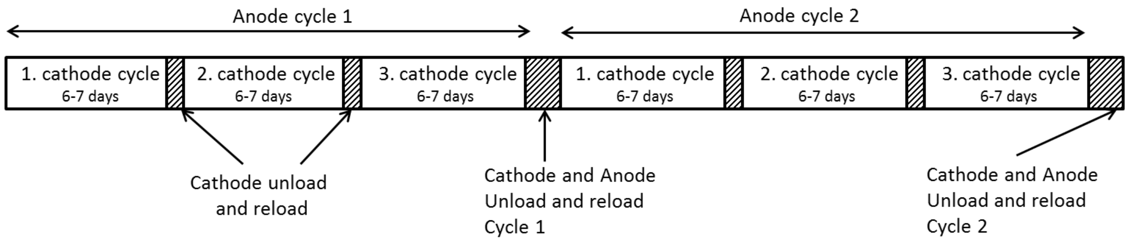

Figure 3-4 Three cathode cycle model

Other common harvesting model is presented in Figure 3-5. The two cathode cycle model contains only two cathode cycles per anode cycle. During an anode harvest the end anodes from either left or right side are changed but not both at once.

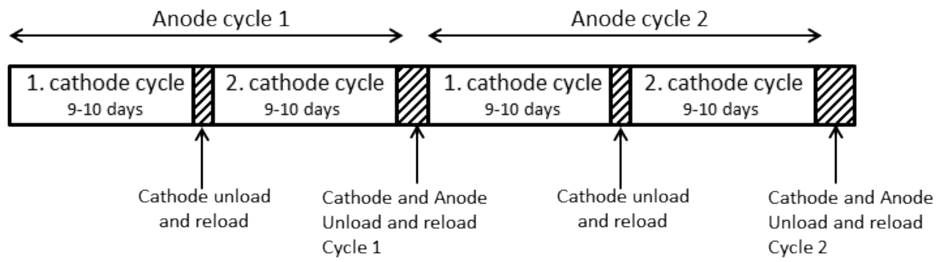

Figure 3-5 Two cathode cycle model

In the case of anode harvest there are more degrees of freedom to do the harvest. In general there are at least four different harvest types: harvest all anodes, harvest middle anodes (leaving one end anode in both ends of the cell), harvest all anodes but one from the left end of the cell and harvest all anodes but one from the right end of the cell. The reason for not to harvest all anodes at once, comes from the layout of the cell (Figure 3-1). The end anodes of the cell are consumed only on one side, therefore they need twice the time that those in the middle. (Schlesinger et al. 2011.)

3.3.2 Scheduling of Copper Electrowinning Process

The basics for the scheduling of EW process are the same as in ER process. The most significant difference is that in EW process there are no cell groups or sections which are harvested in a single run. In the EW process the cells can be harvested separately from each other and there are usually no cell sections as in ER. The scheduling system should take this difference into account. In the EW process the fumes are toxic to humans and usually the cranes are fully automatic and the crane is not operated from the trolley cabin.

4 DESIGN AND IMPLEMENTATION

This chapter contains first the introduction to TIMS and design of the system to be implemented. After the design there will be a description of the implementation. Specific focus being in the Proficy Workflow configuration and usage of ISA-95 based production and equipment models. The extent of the documentation of the Proficy Workflow implementation is wide because this thesis work is also meant to be an introduction this Workflow technology.

4.1 Tankhouse Information Management System (TIMS)

Tankhouse Information Management System (TIMS) is a modern manufacturing execution system developed by Outotec and especially designed for managing processes in ER and EW tankhouses. TIMS is a part of larger scale product family called Integrity and therefore the term Integrity can be used in place of TIMS in some diagrams. In other words: TIMS is an Integrity system for the tankhouse domain. Integrity as a system comprises everything that is extra to the conventional PLC’s and process control, including for example custom code in the PLC’s for reporting, communication between automation and MES layer, integrations to external systems, different databases, Integrity SOA framework and the Integrity user interface. The system contains commercial products from other providers like for example GE.

TIMS consists of different modules, including: production, efficiency, quality and maintenance. The new addition to the modules is a scheduling module which is the main focus and development point in this research. The Integrity and the TIMS are under development at the time of writing this thesis and thus there is still possibility to influence the overall architecture of the system. The general Integrity system architecture is shown in the Figure 4-1. The architecture shown in the Figure 4-1 is from the starting point of this thesis project and does not contain anything related to the scheduling or Proficy Workflow or Proficy SOA.

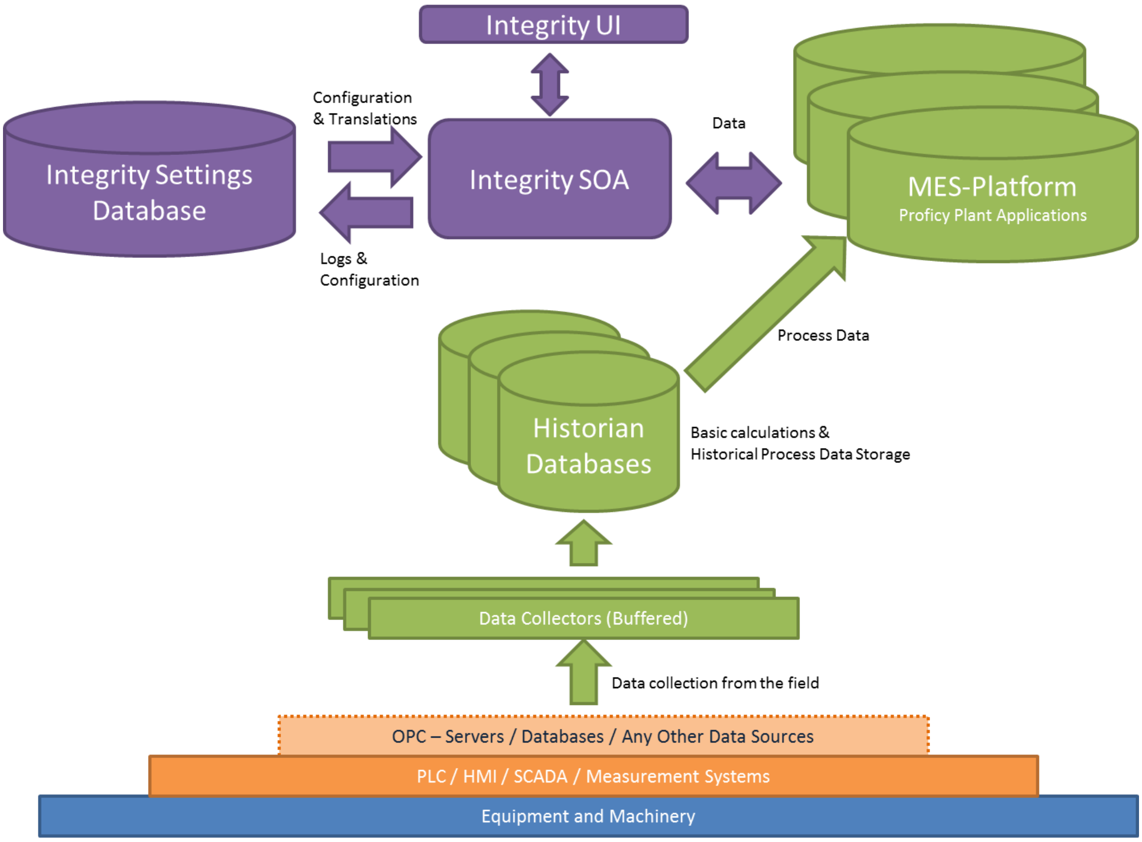

Figure 4-1 Integrity System Architecture at the time of starting this thesis work

Integrity uses the GE Proficy Plant Applications as its MES-platform or MES engine. The Proficy Plant Applications is used for MES functionalities in the background but no user interface components of the Proficy Plant Applications are revealed to the end-user.

The TIMS user interface is a custom made web-based solution designed usability in mind with latest .NET technologies. The interface can be customized per user basis. The end-users can create their own reports and content with graphical online tools.

The Integrity SOA is a set of services, common data objects and interfaces that provide the basic functionalities for the communication and configuration of the system. The services include for example translation, logging, integration and settings. The SOA framework can be extended by adding new services or new interfaces to existing services.

Proficy Historian is used for the historical plant database and the raw process data is stored there. The OPC is used as a standard way to connect to the field devices and other data sources.

4.2 Scheduling in TIMS

The current state of the scheduling in the tankhouse systems is fully manual and offline. The normal scenario is that the schedules are created with for example Microsoft Excel by the production managers and then distributed to all of the operators in tankhouse. The manually created schedule has a high possibility of being outdated if there are any anomalies in the production. In many cases those manually created schedules are not as detailed as they could, leaving room for operators to work according their own possibly inefficient preferences.

From the information system's point of view the heterogeneous nature of different plants and sites is one major problem, because the interfaces and implementations of old systems do not provide equal information that is needed for feedback and state information monitoring.

Scheduling in TIMS will be done in two different levels, which are the tankhouse level ("master schedule") and the field device level ("crane program"). In the future there might be need for connection to the higher level ERP scheduling, which might include for example work shift information. At the point of starting this thesis work the TIMS system did not include any scheduling related functionalities.

4.2.1 TIMS Scheduling Module

TIMS Scheduling module is the largest entity which consists of sub functionalities. As a commercial product it is an optional part of the whole TIMS system. At the time of writing this thesis the scheduling functionalities are based on control of the tankhouse cranes but the production dispatching is done manually by the operators. The cranes can be used to control the tankhouse production operations as a whole, because the cranes operate as a link between different machines in the tankhouse.

4.2.2 Selection of Dispatching and Scheduling Engine

Considering the current TIMS platform it is recommended to seek solutions and tools for implementation within Proficy family. Proficy Workflow is targeted to cases like these with integration and custom communication with systems in different levels of process hierarchy. Proficy Workflow is a potential product and this would also be a pilot project for Outotec to utilize this technology. On the other hand, Proficy Workflow is relatively new product which might bring problems with stability and reliability. Proficy Workflow provides tools for communicating with Proficy Plant Applications and OPC client for communicating with cranes in the field. The program implemented with Proficy Workflow can be customizable with end-user and thus lengthen the lifetime of the whole software solution being implemented. Proficy Workflow's SOA architecture provides a standard way to implement communication with existing Integrity SOA platform. Proficy SOA could also be used to communicate with existing offline scheduling or planning tools to implement the scheduling functionalities. Based on these features and requirements the Proficy Workflow 1.5 SP4 will be used as the engine for this whole dispatching and scheduling module.

4.2.3 Tankhouse State Information in TIMS

Scheduling is based on different variables usually available from process control system. The most important variable is the growing time, which is the actual time the cathodes have been laying in the cell and the current has been going through the cathodes. The other important variable is the current density, which is directly related to the rectifier current. The states of the bypass switches are also monitored, because the bypass switches indicate whether the section is in production or not.

4.2.4 The Master Schedule

One of the challenges regarding the scheduling function is the amount of information to be presented in TIMS UI and also the ability to modify existing schedules and create new ones. It is clear that the scheduling functionalities need to be divided in to different tabs.

4.2.5 The Crane program

The crane program is the crane level part of the TIMS scheduling module. Crane program consists of daily harvesting schedule for the crane. The tankhouse can contain one or more cranes which operate simultaneously, but usually within different sections. The crane program needs to be generic in the way that it can be extended to support more than one crane.

4.3 Selection of the Technologies

The TIMS system is built on top of GE Proficy Plant Applications MES platform and the production dispatching and scheduling functionalities need to communicate seamlessly with different GE Proficy products. The user experience and interface must be in accordance to customer design guidelines and in line with already implemented features. One of the good design principles is to provide single user interface solution to all features instead of multiple different and this principle should be followed in this case also. The scheduling module will be a first feature in TIMS that uses a two way communications with field devices and therefore there are no predefined solutions or interfaces that have to be used. The scheduling module will be part of generic product which should be customizable to different production environments without excess reconstruction or programming.

4.3.1 Selection of User Interface Technologies

The user interface should be part of the TIMS Silverlight user interface and no other end-user interfaces should be used except of configuration interface that could be separate. The design and implementation of the user interface is out of the scope of this thesis however the functionalities of the user interface will be documented in the following chapters.

4.3.2 Selection of Database Technologies

The data storage architecture of scheduling module should follow the general Integrity guidelines and thus use Microsoft SQL Server 2008 R2 as a basis for non-real time data storage and Proficy Historian for real time data storage. Crane program and schedule data will be classified as the non-real time data and all the feedback from the crane PLC will be classified as real time data and stored in Proficy Historian.

All SQL database operations should be executed through T-SQL stored procedures and transaction with stored procedures should be implemented with built in .NET framework's database classes. Direct interaction with database tables or other objects than stored procedures should be denied to ensure the integrity of the database. GE's built-in Historian OPC collector should be used as a data insertion to Historian if applicable but a custom collector could be also implemented if needed.

4.3.3 Selection of Integration Methods to Existing System

The basis for the communication between different parts of the system is Integrity SOA which is implemented with Microsoft's Windows Communication Foundation technology. Required SOA service needs to be implemented according to Integrity design principles. The role of the custom service is to provide data to user interface and forward user input to Proficy Workflow. The custom service will be implemented with Microsoft .NET 4.0 and C# as the other Integrity services are implemented with the same technologies.

Communication with cranes' PLC's will be done with OPC and the server program to provide OPC connectivity will be GE's Industrial Gateway Server (IGS). GE's IGS is basically rebranded version of well-known Kepware KEPServerEX (GE Intelligent Platforms, 2010A).

4.3.4 Selection of Computer Hardware

Integrity software infrastructure allows distributing the system on two or more server computers. Because of the hardware requirements of Proficy Workflow, Proficy Plant Applications, Proficy Historian and Microsoft SQL Server (GE Intelligent Platforms 2010B; GE Intelligent Platforms 2010C) the system should be divided to at least two computers.

4.3.5 Summary of Selected Technologies

Proficy Workflow version 1.5 will be used as an engine behind scheduling module and all custom software components will be designed and implemented according to Integrity design guidelines with Microsoft .NET Framework version 4.0 and C# programming language. Two different database systems will be used. These include Microsoft SQL Server 2008 R2 and Proficy Historian 4.0. Selected technologies are presented in Figure 4-2, where orange boxes are custom made components or parts and blue ones are commercial products. Selected programs and solutions are divided to two server computers.

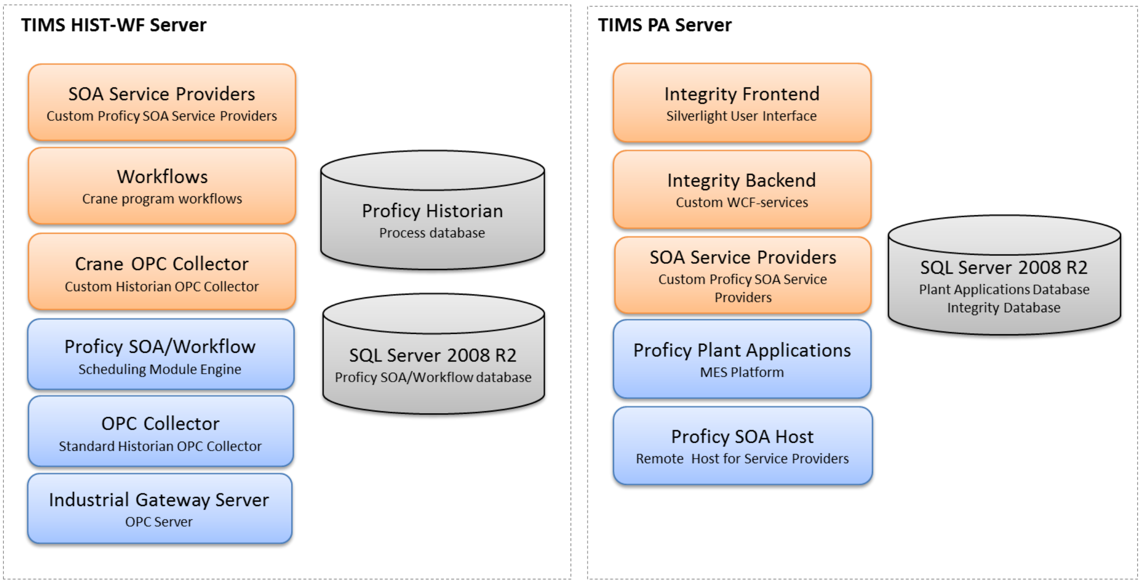

Figure 4-2 Summary of Selected Software Technologies and Hardware Configuration

4.4 Implementation of the Scheduling Module

The implementation of the scheduling module was divided into two parts as discussed in the chapter 4.3. The initial goal is to implement the lower level crane program functionality first and then create the higher level master scheduler, which can then create daily crane programs. The creation of the schedules is done fully or semi-automatically depending on the customer preferences. There is also a possibility to use old scheduling tools for master schedule and to implement only the integration between the old system and new lower level crane program.

4.4.1 Design of the System Architecture

The architecture design in this context addresses concerns mainly with software components but also some high level hardware decisions. There are certain characteristics of the general architecture design that have to be met. These characteristics include the full integration with existing infrastructure and flexibility to support different production environments with different number of cranes and tankhouses containing varying number of sections and cells. The high level architecture picture of the designed system is presented in the Figure 4-3. The architecture design was based on best practices found from installation manuals and guides for these Proficy systems.

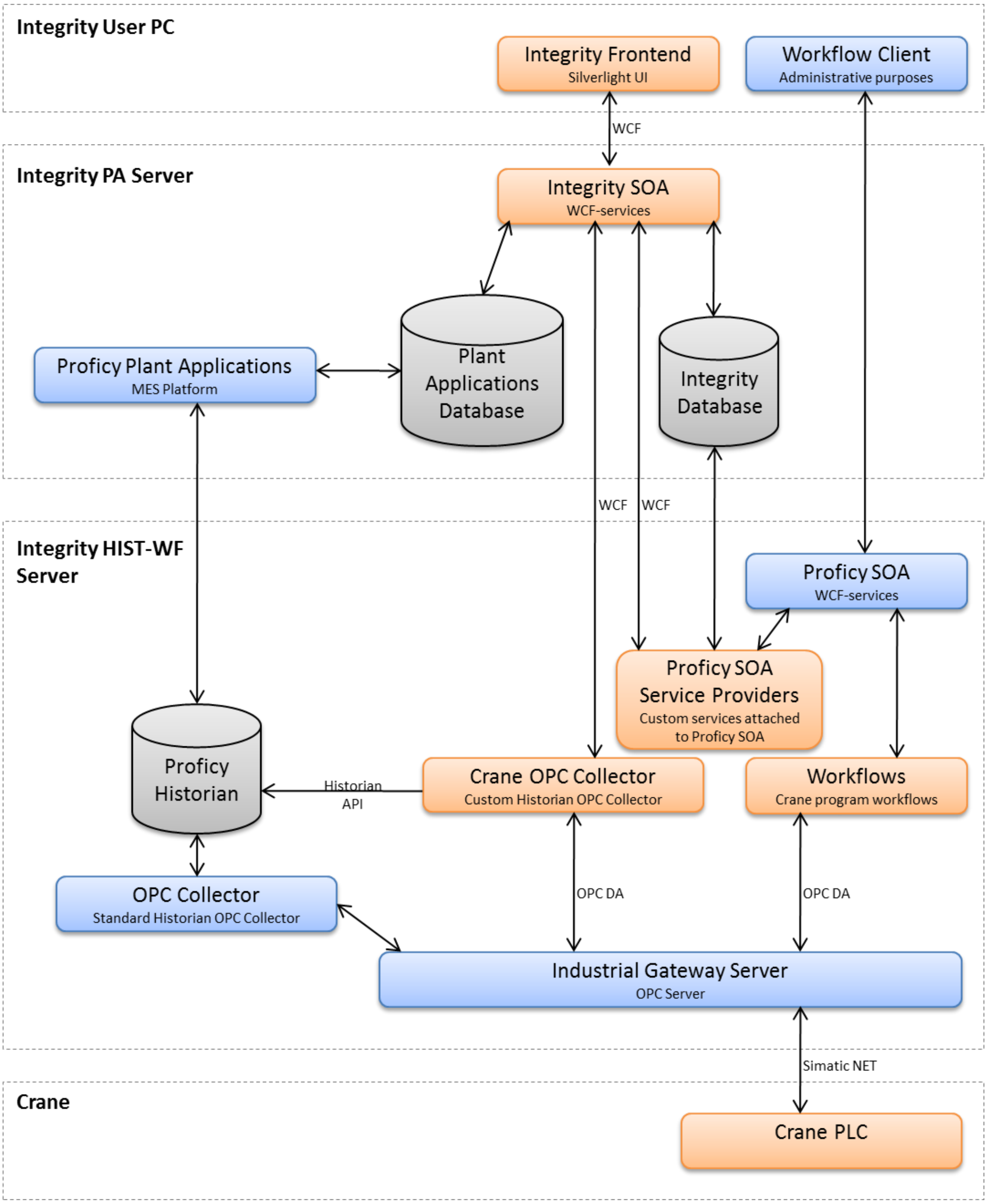

Figure 4-3 TIMS Architecture with Components Required for Scheduling Functionalities

In the Figure 4-3 the blue parts are commercial products and orange parts are custom software that needs to be implemented. The commercial products require configuration and programming that needs to be designed and implemented during the project. More detailed descriptions of each component shown in the Figure 4-3 will be in the chapter 4.4.

4.4.2 Design of the Crane Program

The basis for the crane program design is to leverage Proficy Workflow's resource and process models that are compatible with ISA-95 standard and also to use standard OPC communication. For the basis to the implementation it is crucial to identify and define the interface between PLC and TIMS. In the following chapters the operation of the crane PLC is described.

Crane Program in PLC

The crane program in PLC is basically three arrays of integer values, specifying sections, cells and task types for one day and one crane. The lengths of the arrays are fixed to 100 elements. The limitation is set by the PLC and can be extended if needed. The task array specifies the task to be executed for the cell in the specified section. Possible tasks are listed in the Table 4-1. Values from one equal index in each array constitute one task for a crane. Indexing starts from the zero and ends to the 99, in case there is less than 100 tasks in the crane program the values in the excess indexes should be set to zero in all of the three arrays.

Every task has a name to be shown in the UI and an actual id number that is written to the crane PLC through OPC. Tasks are associated with process types but in this first implementation EW-types are not needed.

Table 4-1 Crane task types with OPC values and descriptions.

OPCID	Name	Description	Process Type
0	No Load	No load	Both
9	Cathode reload	All Cathodes	ER
10	Anode & cathode reload (leave first anode)	Leaves first anode, lifts all cathodes	ER
11	Anode & cathode reload (leave last anode)	Leaves last anode, lifts all cathodes	ER
12	Anode & cathode reload (leave both side anodes)	Leaves first and last anode, lift all cathodes	ER
13	Anode reload (leave first anode)	Leaves first anode, does not lift cathodes	ER

14	Anode reload (leave last anode)	Leaves last anode, does not lift cathodes	ER
15	Anode reload (leave both side anodes)	Leaves first and last anode, does not lift cathodes	ER
16	Anode & cathode reload (all)	Empties a cell	ER

Next there will be presented an example of a crane program. In this example we want to create a crane program that reloads cathodes to the section number six and after that does a full reload (cathodes + anodes) to section number eight. We assume that each section contains 16 cells and they are harvested in order starting from cell number one. The crane program according to the example is presented in Table 4-2. In the table the task type value is the actual value written to the crane PLC, the task type values correspond to the OPCID's in Table 4-1. Index is the actual index of the OPC array.

Table 4-2 Example crane program

Index	**0-15**	**16-31**	**32-99**
Cell Value	1..16	1..16	0 (in each index)
Section Value	6 (in each index)	8 (in each index)	0 (in each index)
Task Type Value	9 (in each index)	16 (in each index)	0 (in each index)

Operation logic and sequence of the crane is described in Appendix 5 Diagram 1. Execution of the new crane program begins when the crane receives information about new schedule. The information is provided by updating the schedule status OPC tag according to Table 4-3. When the status changes to 1 there must be the actual program set in the correct OPC arrays. The crane operator can accept or reject the program and the schedule status is changed accordingly either to loaded or rejected. After the schedule is accepted the program is executed semi-automatically by the operator and the crane.

Table 4-3 Crane program statuses in PLC/OPC

OPC Value	**Schedule Status**	**Set/Updated by**
0	Created	TIMS, actual usage in database only
1	Updated	TIMS
2	Loaded to PLC	Crane PLC
3	In progress	Crane PLC
4	Completed	Crane PLC
5	Rejected	Crane PLC

Crane Program in TIMS

In the initial design phase the main operations of the crane program in TIMS can be divided into two separate workflows, being schedule status monitor and schedule dispatcher. To support the ISA-95 process models other two workflows are created for the purpose of creating a new work request and a crane program scheduler workflow. The designed operations of scheduler workflow and create new work request workflow are described in more detail in the chapter 4.4.3. The designed sequence of the dispatching operation is described in Appendix 5 Diagram 4 and the sequence for status monitor workflow is shown in the Appendix 5 Diagram 3. The final implementation of the crane program is required to have support for multiple cranes. However, the first version implemented during this thesis contains only one crane.

The schedule status monitor workflow will be executed in the case of value change in OPC tag SCHEDULE_STATUS (see Appendix 3 Crane Machine Data). The program status can be updated to the database using only this workflow. Using this approach it can be ensured that the status is updated correctly and only when the status change has actually reached the PLC.

The schedule dispatcher in the Appendix 5 Diagram 4 is the actual workflow that handles update request from other systems. The implementation should support multiple different triggers for this workflow. In the beginning the trigger will be only called from the Integrity Scheduling service, handling requests from Silverlight UI. In the future, however, it is highly likely that this workflow could be triggered by timer or some other external system. The actual schedule will be stored in the common Integrity database. This way the information is stored in only one place and there are fewer problems with synchronization. The workflow will get input information about the crane and the date of the schedule in question. This way the schedule can be identified and updated to correct crane.

4.4.3 Design of the Master Scheduler

The higher level master schedule is not in the scope of this thesis and it will be implemented later, but some design principles and research results will be presented in this chapter. The master schedule should provide higher-level framework to support the functionalities defined in the chapter 4.4.2. All the information about scheduling and scheduled operations is transferred inside work requests. Only exception to this rule is the detailed contents of the schedule. Detailed contents of the schedule needs to be stored in a separate Integrity database due to initial design decisions made with user interface and service backend operations. There would also be a possibility to store all the information within the work request. In that case the same information would be stored in two different locations, not being a good design practice. Main function of the master scheduling is to create the work requests. This master scheduler could be executed every day in certain time or by a request of an end-user.

4.4.4 Implementation of the Crane Program

The crane program functionality was implemented as designed in chapter 4.4.2. The implementation is divided according to different models used in Proficy Workflow. The models are equipment model, production model and workflow model. The security settings and user model of the implementation are not in the scope of this thesis.

Equipment Model

The equipment model was implemented using ISA-95 compatible default tools provided by Proficy Workflow. The equipment model is described in the Appendix 4 Figure 2 and high level linking to ISA-95 terminology is shown in Table 4-4. The production environment is modeled all the way to the work unit level. Cranes are located under single production line named Crane system and single crane is presented as a work cell.

Table 4-4 Equipment model for the Crane Program

Name	ISA-95 Type	Notes
Enterprise	Enterprise	Generic enterprise without properties
Site	Site	Generic site without properties
Tankhouse	Area	
Crane System	Production Line	

Production Model

The purpose of the production model is to define a process, used when updating the crane program to the PLC. The output of the whole process in initial phase is a work request. When the work request is completed there will be work response information linked to the corresponding work request. The production model was created according to general steps described in Table 2-5 Sequence for creating work process in Proficy Workflow.

The production model contains a single work data class named Crane Task Response. This data class can be used to track execution information about the task. In the initial phase this class is added to the model but it is not used, because there are no use cases for the response data. The response data could contain for example information about the actual execution order of tasks and possible reject reasons. The actual response tracking is done in Proficy Plant Applications, from where the information is accessible also by Proficy Workflow.

In the production model there is a single work process segment named Crane Process Segment. The work type of the process segment is production, because this segment defines the actual production start process for a crane. The Crane Process Segment contains also one equipment specification, which is only a class level specification. The class specified is Crane and the actual equipment will be selected during creation of a work request.

The next level in the production model is the work definition segment, which in this case is named Crane Program Definition Segment. The work definition segment extends the production model by adding three new work parameters to it. These parameters can be used to store the contents of the actual crane program but as noted in the design phase, these are optional. Thus, the most reliable and up to date information about each program is stored in the separate SQL-database.

Crane Program Work Definition is the last step in the static production model definition sequence. It contains only one work definition segment, being the Crane Program Work Definition Segment defined in previous paragraph. The type of the work definition is production. The work definition gets all its parameters and specification from previously created work definition and work process segments. View of the parameters in this definition is presented in Figure 4-4.

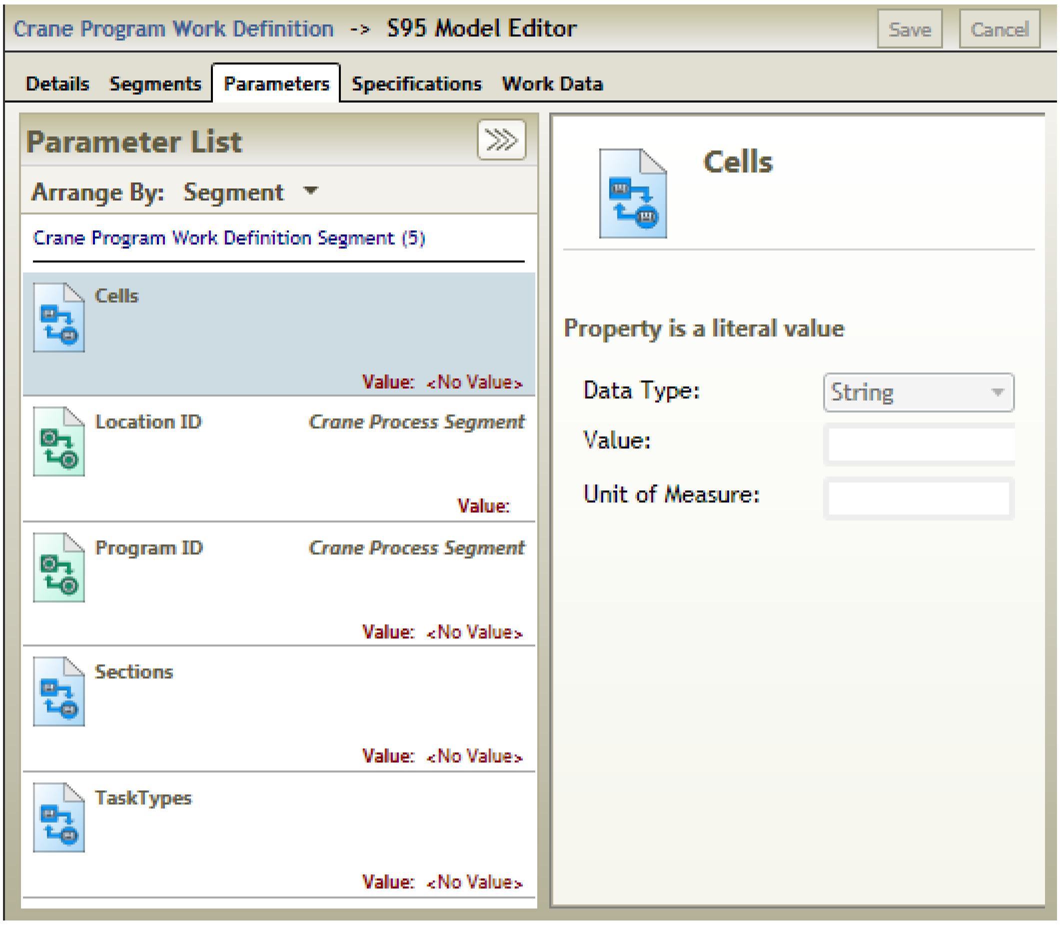

Figure 4-4 Crane Program Work Definition parameters

The actual Crane Program Work requests are created during execution of workflows. More detailed explanations of those workflows are found from following paragraphs. An example of a work request created against production process described above is presented in Figure 4-5. In the figure there are all the common attributes listed and the generic class level equipment specification has been changed to specific crane unit.

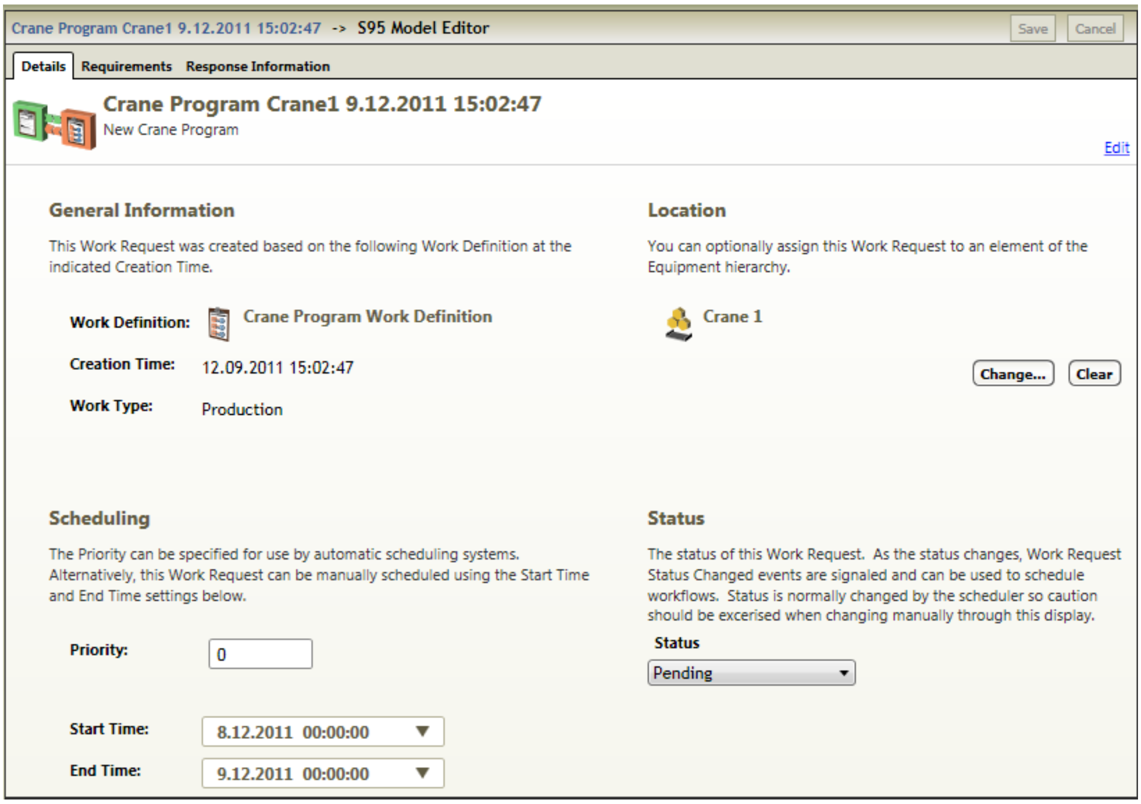

Figure 4-5 Crane Program Work Request

Custom Service Providers

Custom service providers using the Workflow Service Provider API were implemented to support communication to Integrity SOA and handling of complex data types. Custom service providers were implemented with .NET 3.5 C#, because Proficy Workflow 1.5 does not support .NET 4.0. The internal implementations of these service providers are not described in this thesis.

IntegrityServiceProvider is an integration component used for communication between Integrity SOA and Proficy SOA. Basically, this component allows update commands received from Integrity SOA to be sent to Proficy SOA. This component can be extended in the case of other communication needs. Update requests can be seen by Proficy SOA as events, which can be listened by other SOA services.

SchedulerServiceProvider provides multiple functions that are used inside workflows to process custom data types used in Integrity SOA data contracts and to provide SQL-database communication. The workflow editor and model itself does not provide efficient ways to create new objects from custom data contracts and therefore this custom service provider was decided to be created.

Workflow Model

The workflow model implements the actual working logic set by the production model. The production model itself is useless without the execution provided by workflow model. The Workflow model in crane program consists of three workflows, all having multiple sub processes. The internal implementations of each sub processes are not described in detail, but one example of inner implementation is given.

The execution cycle of the crane program creation process begins by invoking the Crane Program Work Request Generator Workflow presented in the Appendix 6 Diagram 1. The workflow can be invoked by multiple ways, but the most common way is to use an event and schedule combination. The event is created by the IntegrityServiceProvider that provides communication to Integrity SOA. Based on the input parameters the workflow will fetch the information from database and create a work request.

The schedule dispatcher workflow presented in Appendix 6 Diagram 2 will be executed from higher level Crane Program Scheduler workflow. The main purpose of the schedule dispatcher is to fetch program information from database and format it to arrays that can be uploaded to crane PLC. All the OPC linking and communication are automatically selected from the crane unit properties defined in equipment model.

Appendix 6 Diagram 5 is an example of the inner implementation of a sub process. The first step in the sub process is a call method activity, which is used to call a function from the custom made SchedulerServiceProvider. Next steps in the sub process are used to validate the crane program object and write to error log in case of invalid crane program.

Monitor schedule status workflow presented in Appendix 6 Diagram 4 is used to track changes in schedule status. The workflow is triggered by the change in schedule status OPC tag. The workflow will update the status first to SQL database and after that the workflow will update the status to the corresponding work request. If the schedule dispatcher workflow updates the program status or the crane PLC updates the status this workflow will update status to the database in both cases.

4.4.5 Implementation of the Master Schedule

The master schedule is out of the scope of this thesis but the high-level implementation suggestion is presented in the Appendix 6 Diagram 3. The implementation is directly from designed structure. The actual implementation does not contain any execution scheduling logic for this workflow.

4.4.6 Implementation of Databases

The implementation of the databases can be divided into the two parts. The first part is a database for the crane program and second part is the scheduling database. Crane program database contains entries for daily crane programs with ability to store template programs/tasks to ease creation of new crane programs. Scheduling database links information from crane programs together with other tasks, including production pauses and other tasks that are vital for production scheduling. All database operations must be executed using T-SQL (Transact-SQL) stored procedures. Stored procedures are used to provide consistent interface to the database and to ensure the integrity of the data. T-SQL stored procedures provide also performance benefit, because only the data needed for the end result are returned to the calling application and all the intermediate processing is done inside the database engine. This approach is also consistent with the one used in Proficy Plant Applications (Wells 2007, p. 8).

Crane program database is designed according to the requirements of the UI. The detailed structure of the database is in the Appendix 1. CP_HarvestType describes the lowest level harvest task type and the content is basically the same as in Table 4-1. Table CP_OperationType is used to store template classes for crane tasks. Table CP_CraneTask contains rows that use CP_OperationTypes for basis of the task.

4.4.7 Implementation of the Communication between Systems

The reliability of the communication between the MES-level and the automation system level were the major concern from the beginning of the design. The problems were caused by unreliable wireless connection between mobile crane and stationary wireless bridge device. The crane operates in a large area and there are dead zones where the wireless signal can’t reach the crane. Some physical changes were made to the communication system, but the reliability was not at acceptable level.

Possible solutions to the problem would be installing the OPC-server program to the crane enabling the OPC-Historian collector automatically to buffer the data collected during connection failures. This solution was rejected, because there is not enough room in the Crane device cabinet to fit a PC. Other solution was to implement buffering in PLC by using acknowledgement design pattern. This acknowledgement design pattern was selected to be used in this case.

The acknowledgement works so that new values are written only when the old ones are acknowledged by the reader. This pattern is sufficient in this scenario, because there are no real-time requirements since the data is used for reporting purposes only. This system should be able to perform correctly even in situations where the crane is stopped in dead space for longer periods of time. This is enabled by the fact that there is only count limit of values in the buffer but no time limit. The Appendix 7Appendix 6 Diagram 1 describes a simplified operating logic of the buffering section of the crane PLC. Basically, the crane executes any given task and compares whether there are old tasks in the send buffer or if not, it can publish the task data immediately to the data block which is visible to OPC Server.

4.4.8 Custom Integrity OPC Collector

Integrity OPC Collector is a custom made OPC to Historian collector, implementing the reading data and acknowledging it to the data source. The buffering decision described in the previous section rendered the basic Proficy OPC-collector useless and forced to implement custom made acknowledge capable OPC-Historian-collector. The OPC-communication was implement by using OPC .NET API 1.3 provided by OPC Foundation. Proficy Historian connection was made with the Historian software development kit (SDK). The Integrity OPC Collector is part of the Integrity SOA and gets its initial settings and configuration data from the Integrity Settings Service. This configuration is designed to be as generic as possible, which enables the collector to be used also in other similar cases requiring collection of buffered data or at least to be used with multiple cranes.

The operating logic of the historian collector is described in the left section of the Appendix 7Appendix 6 Diagram 2. There is also a second sequence describing the operating logic of the standard OPC collector. Integrity OPC Collector monitors for changes in a given OPC tag, which in this case is the MSG_NUM_TO_TIMS. After the detection of the value change the collector will read values of the tags configured as collectable. Values of the tags are written to the Historian database with timestamp got from the timestamp tag.

The basic GE Proficy OPC Collector will be used for collecting the non-task related data, including running counters and other maintenance data. This data is not buffered and it is not considered as critical if some of the values are lost due to connectivity problems. The two different collection sequences are presented in the Appendix 7Appendix 6 Diagram 2. The Integrity OPC Collector is also used to synchronize the PLC’s clock with TIMS.

5 RESULTS AND DISCUSSION

5.1 Overall Design and Infrastructure

The theoretical part focused on introducing the most common standards and software design practices to be used in the empirical part of the thesis work. These design practices included for example integration and communication standards and patterns. The well-known ISA-95 standard based production and equipment models were also used extensively when implementing GE Proficy Workflow based solutions.

The overall design and infrastructure is based on Service Oriented Architecture and the results integrated seamlessly to the existing commercial and custom software components. Custom made software was used only when no suitable commercial or standard solution was available.

The hardware infrastructure was mostly based on the previous design decisions made before the thesis work project. The previously selected hardware infrastructure was suitable for the new features and no significant changes or redesign was required during this thesis work.

5.2 Problems during Implementation

The reliability of Proficy Workflow was the major concern from the beginning of this project. In following sections there will be more discussion about the problems in general and problems specially related to Proficy Workflow. The problems reported in the following chapters are based on release version of Proficy Workflow 1.5 SP4 being the most recent version of the product at the time of starting the implementation phase of this project.

When building industrial grade software, required to be as robust as possible, there should be a way to reset the system in case of critical malfunction. In many cases problems could be solved by restarting computers and other devices. Restarting is a simple operation and does not usually require special technical knowledge and thus can be executed by system operator. Unfortunately, Proficy Workflow has communication features that may get disabled and require manual actions to get everything working as they should.

These features are related to Workflow’s event model. Whenever event trigger source is unavailable the event itself gets disabled. There is no any automatic ways to check or enable the events. Possible event sources are for example OPC or Plant Applications Service Provider for Proficy SOA. This is a major problem especially with system Workflow, having only background tasks running on workflow engine.

One example of problematic operation concerning this implementation is related to communication failures with event source servers. For example, connection to the OPC Server is interrupted, because the OPC Server Service will not start during computer startup. Proficy Workflow’s server service will start and detect that crane program schedule monitoring event data source is unavailable, because the source is an OPC tag. The event gets disabled and only indication of this is in Event Editor, available on in Workflow user interface. In this implementation there are no Workflow user interfaces available to operator. OPC Server gets up and running but the event is still disabled. Next if user uploads new schedule to crane, there will be no feedback or acknowledgement that the schedule was actually uploaded. This way the communication is interrupted as long as someone manually enables those events from the Event Editor.

There have also been problems with system backups using Microsoft Windows volume shadow copy service with Proficy Workflow running. The instance of the Workflows directory service will block the shadow copy service and the backup execution might be interrupted.

There were also some other minor issues which caused problems. To name a few, import and export functionality of the Proficy Workflow does not support manual edit of configurations, causing problems when trying to move configuration to new environment. The import and export error messages can be unclear and not descriptive enough to find the problem.

5.3 The TIMS Scheduling Module

The main focus of this thesis work’s empirical part was to implement production dispatching and scheduling tool to be used copper electrorefining plant. The following chapters will note the results related to this empirical part of the thesis. Only the back-end features and functionalities of the scheduling tool was part of this thesis work.

5.3.1 The Tankhouse Master Schedule

The tankhouse master schedule was not implemented during this thesis work due issues faced during specification and planning. During multiple meetings and discussions with customer it was decided that only estimated harvest times for sections could be calculated. The main problems with the fully automatic and generic online implementation were related to reliable and up to date information of the status of the tankhouse sections and cells. The required status information is described in 4.2.3.

As discussed with customer there are differences of information availability about bypass switches and some systems even provide a reliable information about cathode cycles per section basis. However in the first commissioning planned for this system, the cathode cycle information was not directly available and could not be implemented in Proficy Plant Applications.

The scheduling of the whole tankhouse relies on the information of multiple variables that should be user specified and also fully different scheduling schemes are used as described in chapter 3.3.1. If the information about the cathode cycle and cathode crow times cannot be guaranteed to be correct at all times, there is a need for manual correction feature. The schedule should also take planned and unplanned maintenance task in to account. The manual correction would be very laborious to use, in case of hundreds or even thousands of cells, without intelligent scheduling tools. Without these custom functionalities, which are highly user interface dependent, the usability and overall benefit of the system would be unacceptable low. In the case of communication error or data loss between crane and MES system the recovery to correct operation might not be even possible if the cathode cycle tracking were based on relative calculations from different crane tasks, which means that if a single task for certain cell is missed in some moment of the plants lifetime, all the subsequent calculations are based on incorrect presumptions.

Proficy Workflow would have been responsible for all the logic of the schedule creation and updating at least in the first implementation case. When taking into consideration all the problems with connectivity and reliability of the Proficy Workflow platform it was decided to postpone the full implementation of the master schedule. Also the integration to existing offline scheduling tools is still under development and no results can be presented as a part of this thesis work.

5.3.2 The Crane Program

The crane program is implemented as designed and proven to be functional during FAT. The implementation has gone through multiple iterations of different development tasks and customer needs and specifications have changed multiple times along the time of the development. The user interface is a part of the common Integrity environment and the backend was implemented according to general Integrity design principles by using standard approaches and technologies.

5.3.3 The Scheduling Database

The scheduling database is a central place to store all the data related to the scheduling and dispatching. The scheduling database was designed and implemented using best practices for programming and database design. The interface to different systems is T-SQL stored procedures, which hide the actual implementation from end user and limit the operations to well-defined and easily testable set. By using the T-SQL stored procedures the security and access control can be easily configured in a standard way. The scheduling database is part of Integrity database and contains only the crane program implementation as a working and tested entity. The scheduling database does not contain working master schedule implementation as designed in the beginning.

5.4 Testing

A set of tests was performed to guarantee the validity and reliability of the implemented system. The testing was required by the customer and the test cases and test plan was done part of this thesis work.

5.4.1 Testing in General

The implemented solution was tested in three different levels, which were development time unit testing, larger scale module testing and integration testing with all parts of the system in their own places. In the future there will be at least factory acceptance tests and possibly some other minor tests, due to changes in some parts of the implementation. The testing was performed mainly in office environment during development, but also workshop testing was done.

Tests described in this thesis were performed by the author, but in addition to these test external software testing professionals were used for testing purposes. The planning and reporting of external testing is out of the scope of this thesis.

The following chapters describe what and how tests were executed but do not reveal exact testing results. The results were used to improve the implementation and software bugs were fixed and verified by reruns of the same tests.

5.4.2 Unit Testing

In this context the unit testing stands for testing of small parts containing single functionalities in different parts of the implemented solution. The initial design and implementation took unit testing into account, which made the execution of tests easier.

All T-SQL stored procedures were unit tested separately. In .NET C# backend all methods exposed through WCF interface were also unit tested with simple test client setup. Other internal methods were tested if seen critical for the entity.

The Proficy Workflow implementation was tested per workflow basis and by using simulated OPC tags and also with PLC simulation implemented in similar hardware that the actual PLC's controlling the cranes have.

The reporting part related to Proficy Plant Applications used simulated data as basis for calculations and reports. The OPC communication with real PLC was tested manually with third party OPC client.

5.4.3 Module Testing

Testing plan for the module test is presented in the Appendix 8. The testing plan was created to support module testing during development but also to be a base for the more comprehensive FAT (Factory Acceptance Test) being executed in the future. The test plan contains also some optional parts, which were skipped during the initial testing but will be part of the FAT. The initial testing was completed in co-operation with automation specialist responsible for PLC implementation and in the supervision of project manager. According to test plan the tests are divided into two different categories, connectivity tests and functional tests. In the end of the test plan there are predefined test cases that can be used to test the functional requirements.

The connectivity tests part aims to reveal problems and possible misconceptions with interface definitions between PLC and MES layers. The main task in connectivity tests is to check the PLC/OPC data block definitions and compare the addresses in OPC server configuration with the actual PLC data block implementations.

Functional tests are executed according to test cases defined in Appendix 8 chapter 2. The functional tests mainly focus on correct operation and output in Proficy Historian, Proficy Plant Applications, Proficy Workflow and also in PLC HMI. In the case of FAT the expected output values should be defined also in the test document, but for the purpose of development time testing, it was decided that it is not necessary information.

5.4.4 Integration Test

The tests performed in chapter 5.4.3 Module Test can be also categorized to be integration tests, because of the test setup contained all components of the actual system working together. Therefore there was no need to perform separate integration tests.

5.5 Evaluation of the Results

The theoretical part was based on standards and publications from distinguished non-profit organizations, which strengthens the reliability of the statements made. The initial specification of the technologies to be used and researched was set by the customer and this had a strong effect on all the choices made throughout the thesis. The customer preferences and orientation of my own organization had an effect also on the theoretical part of the thesis.

Empirical research and implementation was first and foremost made according to the design principles set by the customer, which might not had been the best practices. Especially the decision to use GE Proficy Workflow. GE Proficy Workflow 1.5 SP4 was first of all too heavy platform for fairly simple solution to be implemented but on the other hand this was a pilot project to use the Proficy Workflow.

The MES platform part of the thesis was mainly based on marketing materials provided by the system vendor, which may affect the reliability of the information. The actual MES platform in Integrity is Proficy Plant Applications and the possibility to actually test other systems were highly limited and thus the comparison would have been based on marketing material of the system providers

The reliability of the software products can be divided to reliability of commercial and custom software. During development and testing phase the custom software proved it to be reliable but especially the version of Proficy Workflow used is not reliable enough for large scale production use.

6 CONCLUDING REMARKS

This thesis introduced the theoretical background of the manufacturing execution systems in terms of most common standards and software technologies. The research problem was related to the scheduling and dispatching operations of copper electrolysis production processes. One of the important aspects of this thesis work was to test a new GE Proficy Workflow based approach to integration and communication related challenges.

6.1 Conclusions

This was a first of a kind implementation to provide automatic scheduling tool for copper electrorefining process. This was also a pilot project for the customer to leverage Workflow technology to configure and create integration and communication solution using ISA-95 standard based GE Proficy Workflow product. Previously the electrorefining process was scheduled by manually transferring the information from planning systems to the crane operator. This new scheduling tool allows the production planners to automatically transfer the information to the machine.

The high level Tankhouse Master Schedule was not implemented but initial implementation recommendation was given as part this thesis. The Master Schedule was not implemented due to technical issues faced during design and planning face with the reality that the other systems related to the first commissioning project does not provide required information for the automatic schedule creation.

One important part of the theoretical research was to evaluate the modern SOA and Workflow based software development tools for the customer. GE Proficy Workflow, the product to be evaluated and tested was selected by the customer before starting this thesis project thus there were no possibilities to influence this decision. The ideology and structure of the GE Proficy Workflow 1.5 SP4 seemed to be very well suited for this kind of integration and production dispatching solutions. The actual real-life performance and the lack of robustness of the Proficy Workflow were not at acceptable level.

6.2 Further Development

During the software development most of the problems were related to the GE Proficy Workflow, a commercial platform used for the implementation. Because of these problems the extended usage and further development of Proficy Workflow based solutions was put on hold. Questions and descriptions about the problems faced during the development process with Proficy Workflow were sent to the platform developers. The questions were sent to GE but no relevant answers were got back.

The communication interface between the crane and TIMS should be updated to support more flexible insertion of new tasks. This update requires significant changes in the crane PLC but also major changes in the MES level. The tankhouse level master scheduler should be implemented also to provide fully automatic schedule creation. The user interface should be developed to better support the new master schedule with functionalities to edit the master schedule. The crane scheduling user interface should also include more detailed monitoring features of the progress of the execution.

The production plants where this software tool can be used may be located in any part of the world, which means that it is certain that the operators and end users have their own ways of doing things. This implementation can be thought as a pilot project and therefore it is essentially important to get end user feedback about the system to further develop it to better respond to the users' needs and preferences. It must be also kept in mind that this system should be emphasizing the best practices, which might contradict with someone's normal way of doing things.

REFERENCES

ANSI/ISA–95.00.01–2000, 2010. Enterprise-Control System Integration. (Standard) Instrument Society of America, Research North Carolina, USA 15th May 2000. 142 p.

ANSI/ISA–95.00.02–2001, Enterprise-Control System Integration Part 2: Object Model Attributes. (Standard). Instrument Society of America, Research North Carolina, USA 17th Oct. 2001. 104 p.

ANSI/ISA–95.00.03–2005, Enterprise Control System Integration Part 3: Activity Models of Manufacturing Operations Management. (Standard). Instrument Society of America, Research North Carolina, USA 6th Jun. 2005. 104 p.

ANSI/ISA–95.00.05-2007, Enterprise-Control System Integration Part 5: Business-to-Manufacturing Transactions. (Standard). Instrument Society of America, Research North Carolina, 10th Jan. 2007. 124 p.

Automation Resources, inc 2010, GE Fanuc Acquires Mountain Systems Inc., Providers of Advanced Manufacturing Execution Systems for Essential Industries. [WWW]. [Accessed 19.9.2012], URL: http://www.automation.com/content/ge-fanuc-acquires-mountain-systems-inc-providers-of-advanced-manufacturing-execution-systems-for-essential-industries

Buhulaiga, E.A. 2007. SOA for Manufacturing. Orlando, Florida, USA 16-18.9.2007, 2007 Plant-to-Enterprise Conference. 39p.

Chang, S. & Hsieh, F. 1992. Order and Production Scheduling/Rescheduling for flow shops. Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France, May 1992. 6 p.

Chappell, D. 2009. The Workflow Way: Understanding Windows Workflow Foundation. Microsoft Corporation. 27 p.

Chappell, D. 2010. Introducing Windows Communication Foundation. Microsoft Corporation, January 2010. 37 p.

Davis, A. & Zhang, D. 2002. A Comparative Study of DCOM and SOAP. Multimedia Software Engineering, 2002. Proceedings. Fourth International Symposium, February 25th 2002. pp. 48-55.

Fazlollahtabar, H. & Mahdavi, I. 2009. Applying Stochastic Programming for Optimizing Production Time and Cost in an Automated Manufacturing System. Computers & Industrial Engineering, 2009. July 6 – 9, 2009. pp. 1226 – 1230.

GE Intelligent Platforms. 2009. Implementing S95 with Proficy Workflow. Technical White Paper. GE Fanuc Intelligent Platforms Inc.

GE Intelligent Platforms. 2010A. GE Intelligent Platforms Launches Industrial Gateway Server 7.5 for Data Acquisition, Visualization and Control With Proficy® Software. Charlottesville, VA. Jan 19, 2010. Press Release, available on: http://www.genewscenter.com/content/Detail.aspx?ReleaseID=9452&NewsAreaID=2

GE Intelligent Platforms. 2010B. Proficy Plant Applications 5.0 Help. GE Intelligent Platforms Inc.

GE Intelligent Platforms. 2010C. Proficy Workflow Getting Started. GE Intelligent Platforms Inc. August 2010. 101 p.

General Electric Company 2012. Our Company. [WWW]. [Accessed 19.9.2012], URL: http://www.ge-ip.com/

Gifford, C. 2007. Integration Standards Converge into Manufacturing Application Framework. Utrecht, Netherlands, 2007 European Plant-to-Enterprise Conference. 39 p.

Gifford, C. 2011. When Worlds Collide in Manufacturing Operations: ISA-95 Best Practices Book 2.0. New York, USA, International Society of Automation (ISA)

Hartman, H. 1992. SME Mining Engineering Handbook 2nd Edition. Society for Mining, Metallurgy, and Exploration, Inc. 2255 p.

ISA-88, 2010, ISA-88: the international standard for flexibility in production. [WWW]. [Accessed 27.10.2011], URL: http://www.isa-88.com/

Jankowski, T., Davis, G., Holmes, J., & Kemper, G. 2011, Increasing Data Historian Efficiency, Cement Industry Technical Conference, 2011 IEEE-IAS/PCA 53rd, on 22-26 May 2011, pp. 1-14

Kashyap, N. & Emerson, D. 2010. A Workflow Driven Approach to Mfg. Operations Management (MOM), Chandler, Arizona, USA, MESA International 15.6.2010.

Komoda, N. 2006. Service Oriented Architecture (SOA) in Industrial Systems. Industrial Informatics, 2006 IEEE International Conference on 16-18 Aug. 2006, (updated August 8th 2007). 5 p.

LeBlanc, L. 2002 Production Workflows, MESA International 2002 North American Conference (Presentation). 29 p.

Li, S., Li, Y. & Liu, Y. 2006. Effects of Process planning Upon Production Scheduling under concurrent Environment. Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 – 23, 2006, Dalian, China. 5 p.

Logica. 2011. MES Product Survey 2011 Flyer. 12th edition. September 2011. Logica MES Centre of Excellence. 4.p

MacDonald, M. 2010, Pro WPF in C# 2010 Windows Presentation Foundation in .NET 4. New York, Springer -Verlag. 1180 p.

OMG. 2011. OMG Unified Modelling Language (OMG UML) Infrastructure. Version 2.4.1. August 5th 2011. Object Management Group. 230 p.

OPC Foundation. 2013. OPC - The Interoperability Standard for Industrial Automation & Other. [WWW]. [Accessed 23.04.2013], URL: http://www.opcfoundation.org.

OPC Task Force. 1998. OPC Overview. Version 1.0, October 27th 1998. OPC Foundation. 16 p.

Philips, M., Both J. 2010. Implementing OEE Reporting In Manufacturing. Chandler, Arizona, USA, MESA International, 14.6.2010.

Samaras, I.K., Gialelis, J.V., Hassapis, G.D. & Akpan, V.A. 2009. Utilizing Semantic Web Services in Factory Automation towards Integrating Resource Constrained Devices into Enterprise Information Systems. Emerging Technologies & Factory Automation, IEEE Conference 2009. pp. 1 – 8.

Schlesinger, M., King, M., Sole, K., Davenport, W. 2011. Extractive Metallurgy of Copper, Fifth Edition. Elsevier Ltd, Langford Lane, Kidlington, Oxford, United Kingdom. 356 p.

Scholten, B. 2007. The Road to Integration: A Guide to Applying The ISA-95 Standard in Manufacturing. New York, USA, International Society of Automation (ISA). 235 p.

Scholten, B. 2009. MES Guide for Executives: Why and How to Select Implement and Maintain a Manufacturing Execution System. New York, USA, International Society of Automation (ISA). 160 p.

Shibata, T., Zhang, X., Hai, X., Shmizu, Y. & Fujimura, S. 2007. Generalization of Scheduling Information Acquired Through User Manipulation for Self-construction Production Scheduling System. Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering, Scottsdale, AZ, USA, September 22 – 25, 2007. 6 p.

Siemens AG. 2012, The Company, Siemens 2012. July 2012. Press release. Available on: http://www.siemens.com/press/pool/de/homepage/the_company_2012.pdf

Snoeij, J. 2012. MES Product Survey 2012. Logica MES Centre of Excellence. Meander, Amhem, Netherlands. 622p.

Son, M., Yi M. 2010. A Study on OPC Specifications. Perspective and Challenges, Strategic Technology (IFOST), 2010 International Forum (updated December 13th 2010), pp. 193 – 197

Stelter, M. & Bombach, H. 2004. Process Optimization in Copper Electrorefining. Advanced Engineering Materials 2004(7), Wiley-VCH, Verlag, Weinheim, Germany. pp. 558-562

Suomen Automaatioseura ry. 2011 MES Market Survey Conference on 11.5.2011 Finland.

Suomen Automaatioseura ry. 2001. Laatu Automaatiossa. Suomen Automaatioseura ry, Helsinki, Finland. 1. edition. 245 p.

Tan, V.V. Yoo, D. & Yi, M. Modern Distributed Data Acquisition and Control Systems based on OPC Techniques. School of Computer Engineering and Information Technology, Ulsan, Republic of Korea. 8 p.

Todorov, I. 2007. OPC UA – Road to the future. Wonderware/Invensys, OPC Foundation 34 p. (Presentation)

Yalaoui, N., Camara, M., Amodeo, L., Yalaoui, F. & Mahdi, H. 2009. New Heuristic for Scheduling Re-entrant Production Lines. Computers & Industrial Engineering, CIE, July 6 – 9 2009. 6 p.

Young, C., Taylor, P., Anderson, G., Choi, Y. 2008. Hydrometallurgy 2008: proceedings of the sixth international symposium. Society for Mining, Metallurgy, and Exploration Inc. Shaffer Parkway,Colorado, USA. 1186 p.

Vollmann, T.E, Berry, W.L., Whybark, D.C. & Jacobs, F.R. 2005. Manufacturing Planning and Control for Supply Chain Management. New York, USA, McGraw Hill Companies Inc. 2005 fifth Edition. 712 p.

Wadawadigi, G. et al. 2010, Lean Manufacturing Strategic Initiative Guidebook, Chandler, Arizona, USA, MESA International, 6.12.2010. 93 p.

Wu, N., Qian, Y. & Yu, Z. 2004. Manufacturing process reengineering for mass customization by using flexibility analysis. Systems, Man and Cybernetics, 2004 IEEE International Conference. 10-13 October 2004. pp. 5062 – 5067.

Brandl, D. & Emerson, D. 2008. Business to Manufacturing Language B2MML Common Schema Documentation Version 0401. WBF – The Forum For Automation and Manufacturing Professionals. October 2008. 35 p.

WBF – The Organization of Production Technology. 2011. [WWW]. [Accessed 08.12.2011]. URL: www.wbf.org

Wells, M. 2007. Professional Services SQL Programming Guidelines, GE Fanuc Automation. 120 p.

Zheng, L., Nakagawa H. 2002, OPC (OLE for Process Control) Specification and its Developments. SICE 2002. Proceedings of the 41st SICE Annual Conference 2(2002), pp. 917 – 920

Appendix 1. Crane Program Database Structure

CP_CraneProgram

Column Name	Condensed T...	Nullable	Iden...
ID	int	No	☑
TS	datetime	No	☐
IsInUse	bit	No	☐
CraneID	int	No	☐
Status	int	No	☐

CP_CraneTask

Column Name	Condensed Type	Nullable	Iden...
ID	int	No	☑
OperationType	int	No	☐
OperationDesc	nvarchar(100)	Yes	☐
PeriodNum	int	Yes	☐
ProgramID	int	No	☐
OrderNum	int	No	☐
CraneID	int	No	☐
Comment	nvarchar(100)	Yes	☐
IsInUse	bit	No	☐

CP_CraneProgramStatus

Column Name	Condensed T...	Nullable	Iden...
ID	int	No	☐
Name	nvarchar(100)	No	☐

CP_OperationType

Column Name	Condensed T...	Nullable	Iden...
ID	int	No	☑
Name	nvarchar(100)	No	☐
Description	nvarchar(100)	No	☐
HarvestType	int	No	☐
IsInUse	bit	No	☐
CraneID	int	No	☐
CellStart	int	No	☐
CellEnd	int	No	☐
SectionNum	int	No	☐

CP_Section

Column Name	Condensed T...	Nullable	Iden...
Id	int	No	☑
Site	nvarchar(50)	No	☐
Section	int	No	☐
Cells	int	No	☐
ProcessType	nvarchar(10)	No	☐

CP_HarvestType

Column Name	Condensed T...	Nullable	Iden...
ID	int	No	☐
OPCID	int	No	☐
Name	nvarchar(50)	No	☐
Description	nvarchar(500)	Yes	☐
ProcessType	nvarchar(10)	No	☐
Active	bit	No	☐

Appendix 2. Crane Task Data

OPC DATATYPE	DATABLOCK REF.	OPC NAME	DESCRIPTION
Short	-REMOVED-	MSG_NUM_TO_TIMS	Crane task number. Counter that is increased when crane receives new target.
Short	-REMOVED-	MSG_NUM_TO_CRANE	Message number from TIMS to PLC, aka. acknowledgement number.
BCD Array	-REMOVED-	TIMESTAMP	Task timestamp. Time when task id was increased. Format [YYMM, ddhh, mmss]
Short	-REMOVED-	ACTION_NUMBER	Action number: • 1 = Lift load • 2 = Leave load
Long	-REMOVED-	WAITING_MANUAL	Waiting time during task in manual mode. In seconds.
Long	-REMOVED-	WAITING_AUTO	Waiting time during task in auto mode. In seconds.
Long	-REMOVED-	WAITING_POWER_OFF	Waiting time when crane power is off
Long	-REMOVED-	TOTAL_TIME	Total time consumed during task in seconds.
Short	-REMOVED-	GROUP_NUMBER	Group number where task happened. 0 = machine or rack.
Short	-REMOVED-	CELL_NUMBER	Cell number where task happened. • Two digit = cell • 1xy = Rack • 2xy = CSM • 3xy = APM • 4xy = ASW
Long	-REMOVED-	HORIZONTAL_MOVE_AUTO	Horizontal movement in auto mode during the task in seconds.
Long	-REMOVED-	HORIZONTAL_MOVE_MAN	Horizontal movement in manual mode during the task in seconds.
Long	-REMOVED-	VERTICAL_MOVE_AUTO	Vertical movement in auto mode during the task in seconds.
Long	-REMOVED-	VERTICAL_MOVE_MAN	Vertical movement in manual mode during the task in seconds.
Short	-REMOVED-	LOAD_TYPE	Load type of the action: • 10 = AnodeL (ER – leaves first anode, lifts all cathodes) • 11 = AnodeR (ER – leaves last anode, lifts all cathodes) • 12 = AnodeB (ER – leaves first and last anode, lifts all cathodes) • 13 = AnodeL (ER – leaves first anode, does not lift cathodes) • 14 = AnodeR (ER – leaves last anode, does not lift cathodes) • 15 = AnodeB (ER – leaves first and last anode, does not lift cathodes) • 19 = All (ER – empties a cell) • 0 = No load

Appendix 3. Crane Machine Data

OPC DATATYPE	DATABLOCK REF.	OPC NAME	DESCRIPTION
Short	-REMOVED-	RUN	Rung tag: • 0 = Automatic mode, in production • 1 = Target mode, in production • 2 = Manual mode • 3 = Auto mode and waiting • 4 = Auto mode and there is an alarm • 5 = Waiting Cell • 6 = Waiting load conveyr • 7 = Waiting ASW • 8 = Waiting APM
Short	-REMOVED-	FAULT	Fault tag, see detailed description in xx
Short	-REMOVED-	SCHEDULE_STATUS	Schedule status: • 0 = Created • 1 = Updated • 2 = Loaded to PLC • 3 = In progress • 4 = Completed • 5 = Rejected
BCD Array	-REMOVED-	TIME_TO_CRANE	Time synchronization, updated before every acknowledgement.

Appendix 4. Implementation Equipment Model

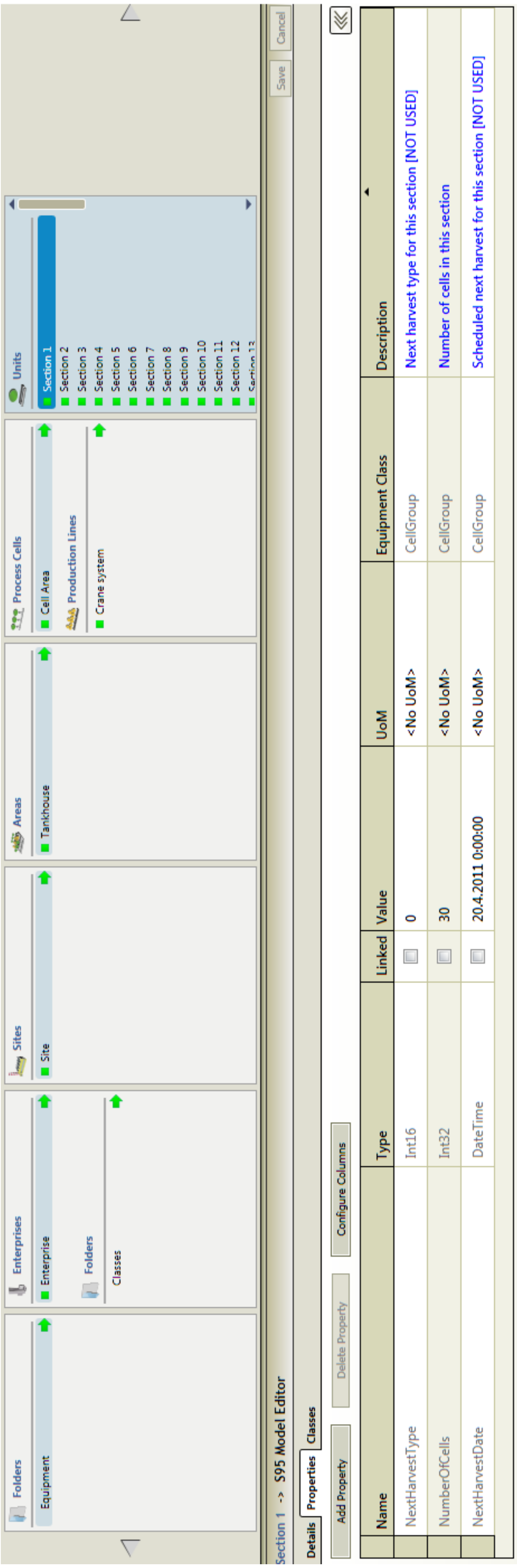

Figure 1 Proficy Workflow equipment model with section properties selected

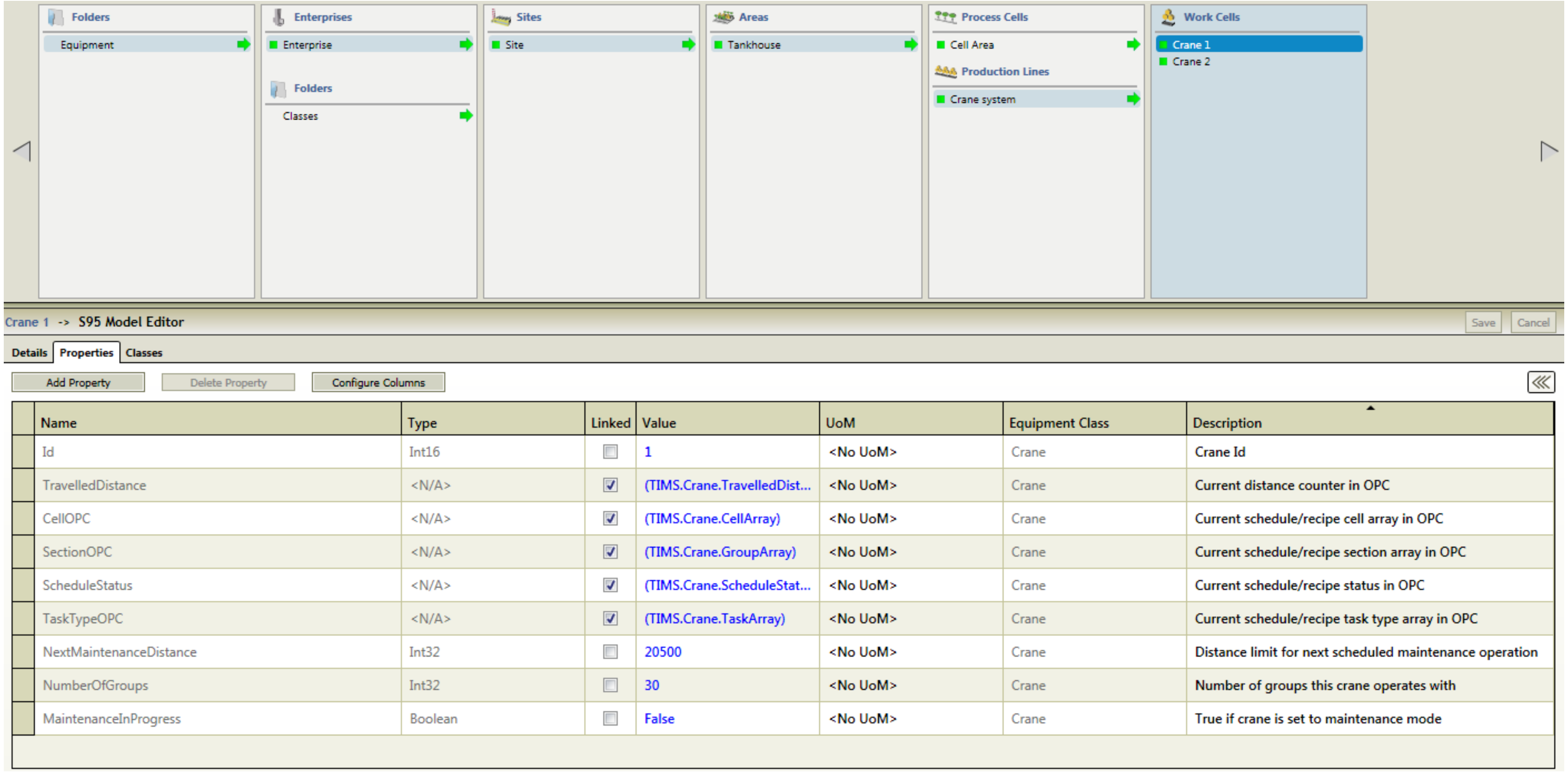

Figure 2 Proficy Workflow equipment model with crane properties selected

Appendix 5. Activity Diagrams for Crane Communication

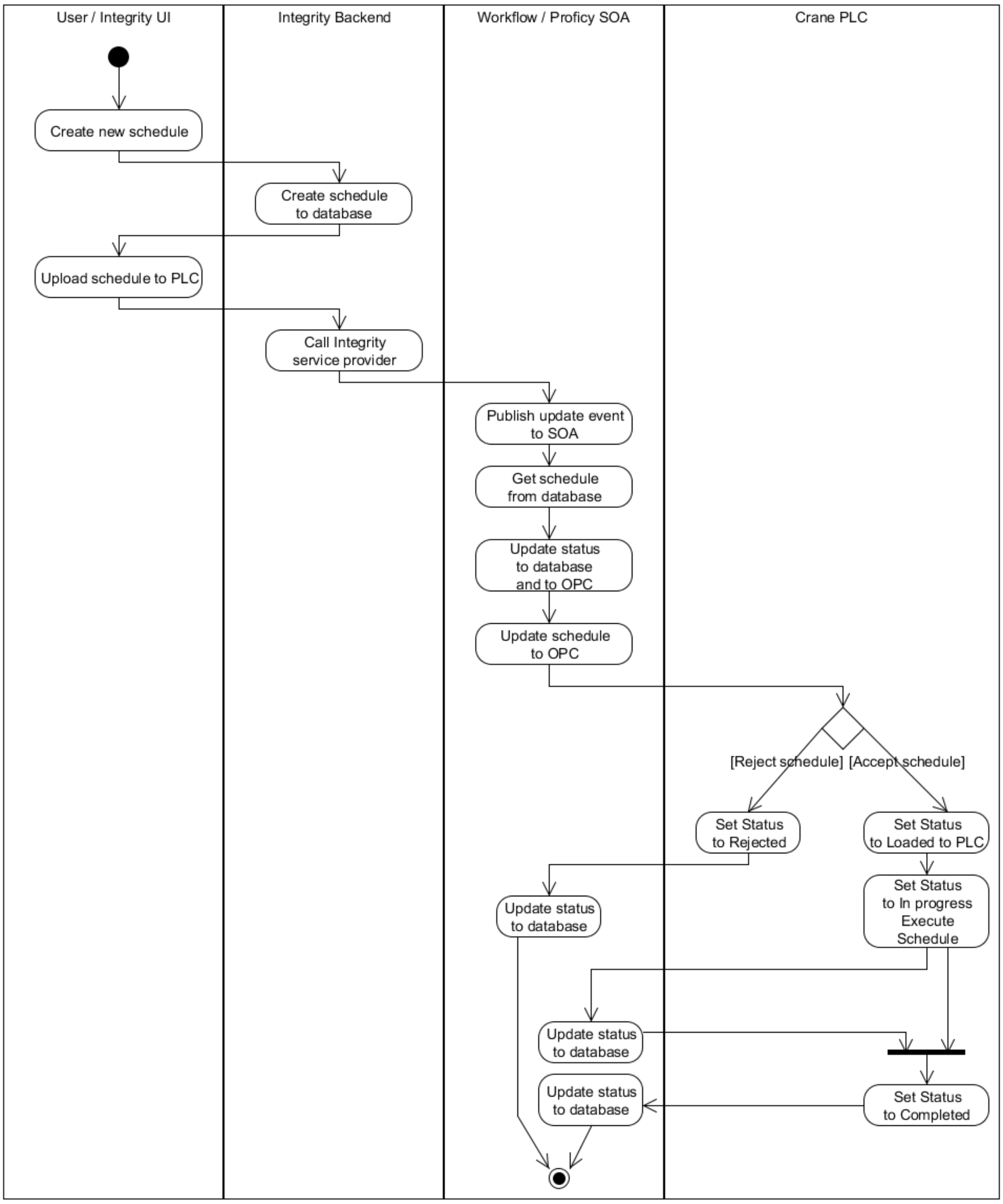

Diagram 1 Activity diagram for schedule update

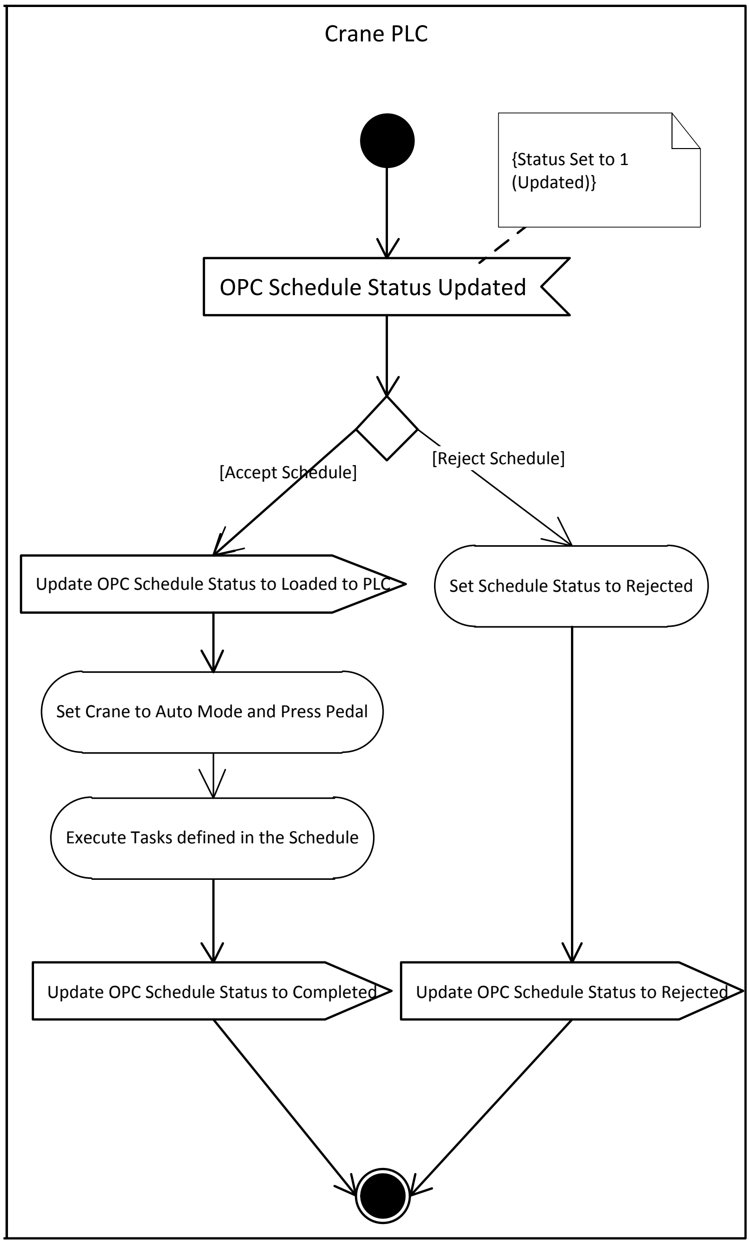

Diagram 2 Crane PLC's activity diagram for schedule handling

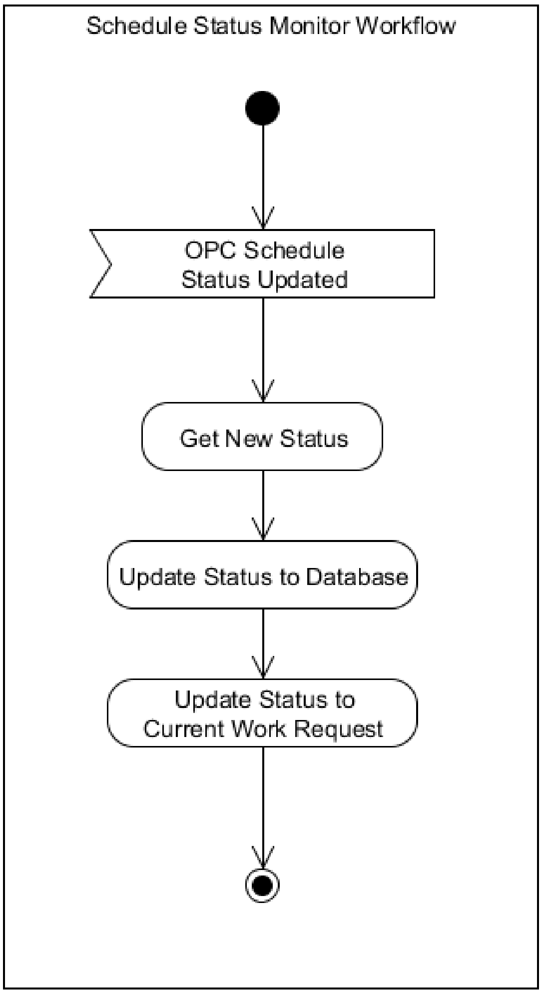

Diagram 3 Activity diagram for schedule status monitor workflow

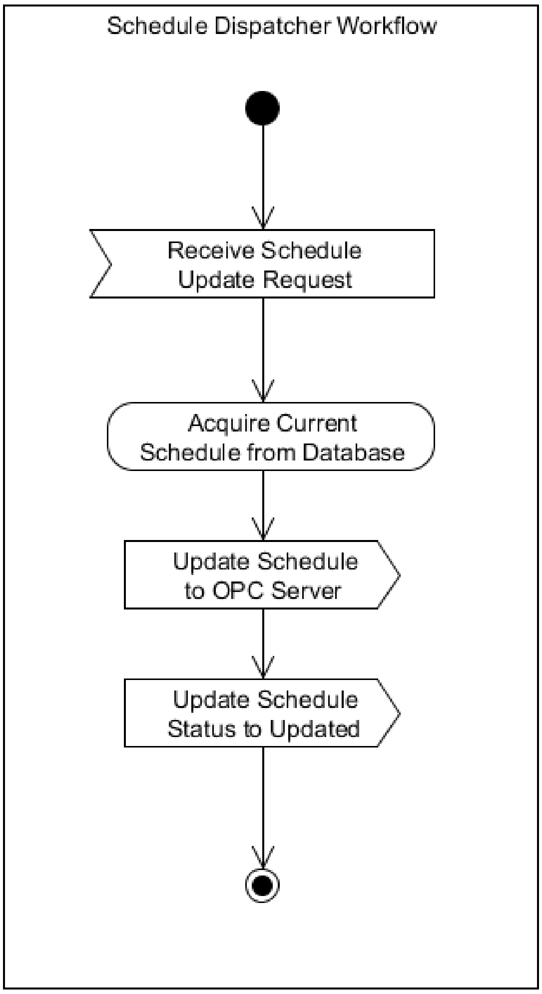

Diagram 4 Activity diagram for schedule dispatcher workflow

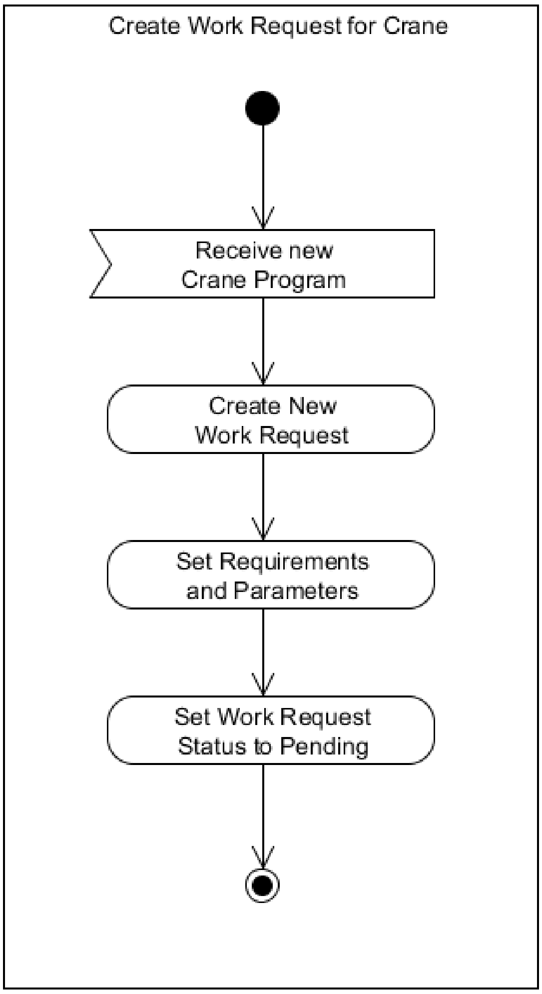

Diagram 5 Activity diagram for create work request for crane

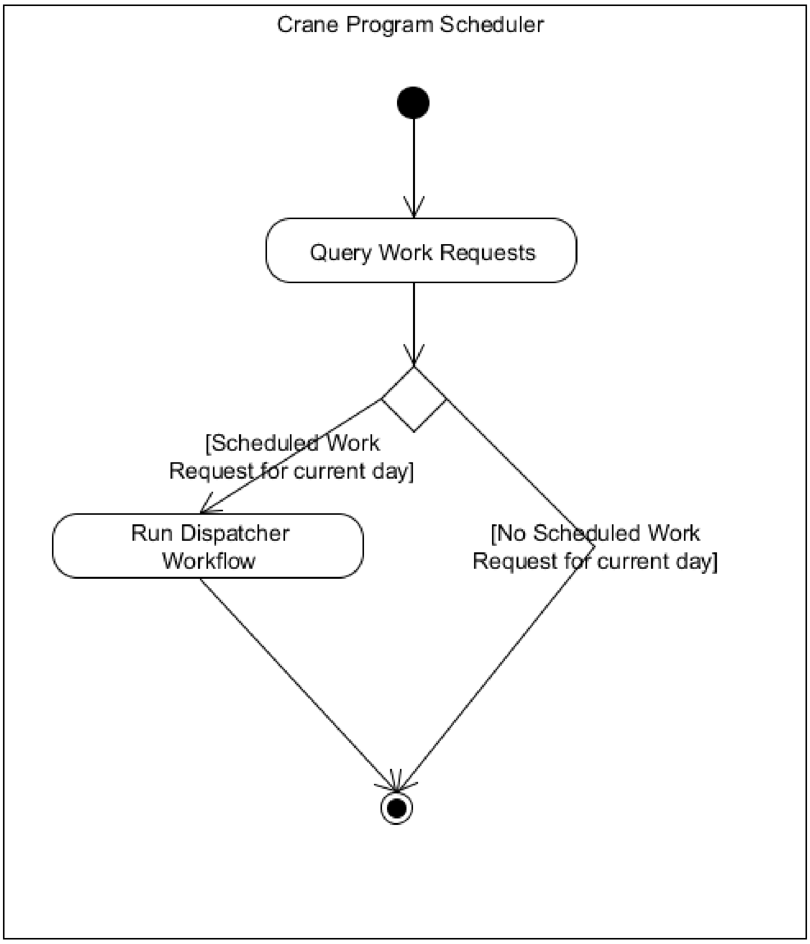

Diagram 6 Activity diagram for Crane program scheduler

Appendix 6. Implemented Workflows

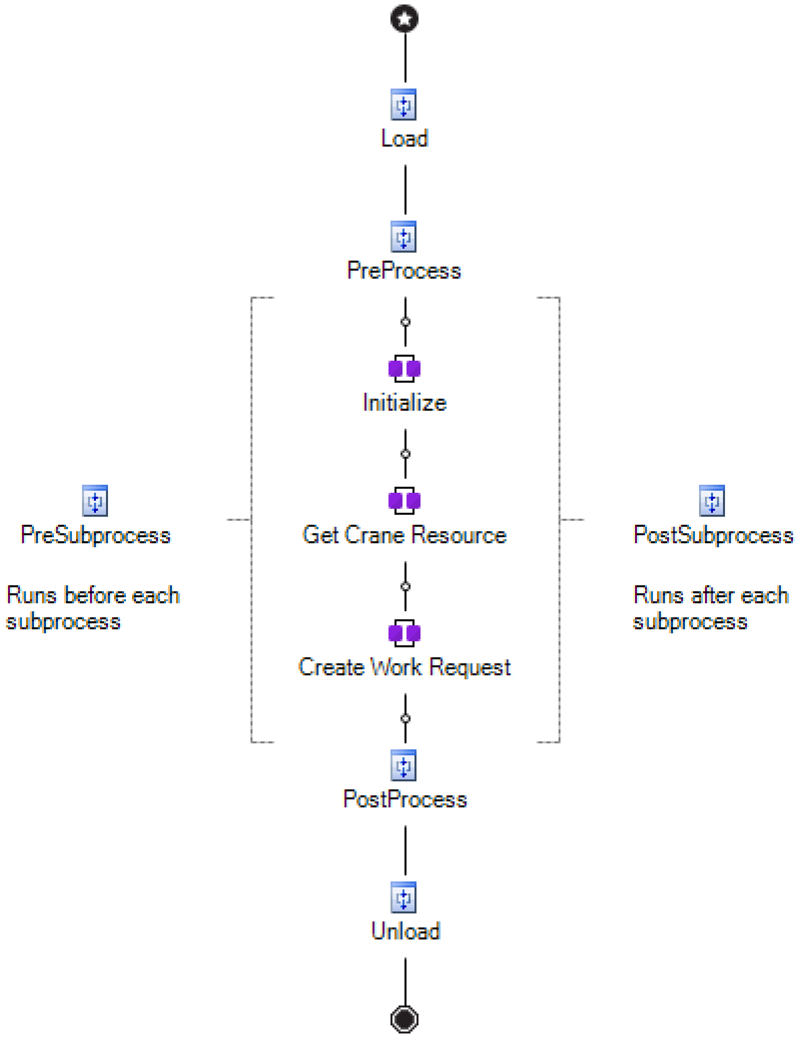

Diagram 1 Crane Program Work Request Generator

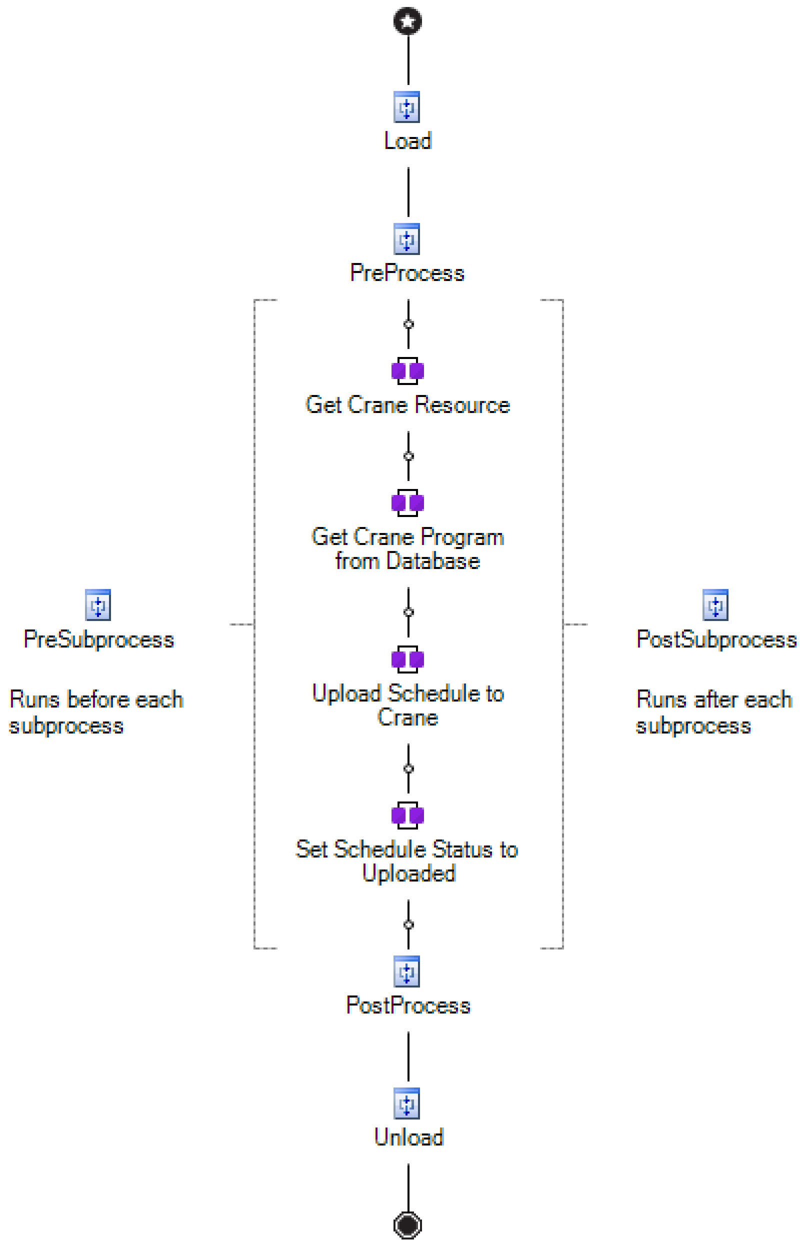

Diagram 2 Schedule dispatcher workflow implementation

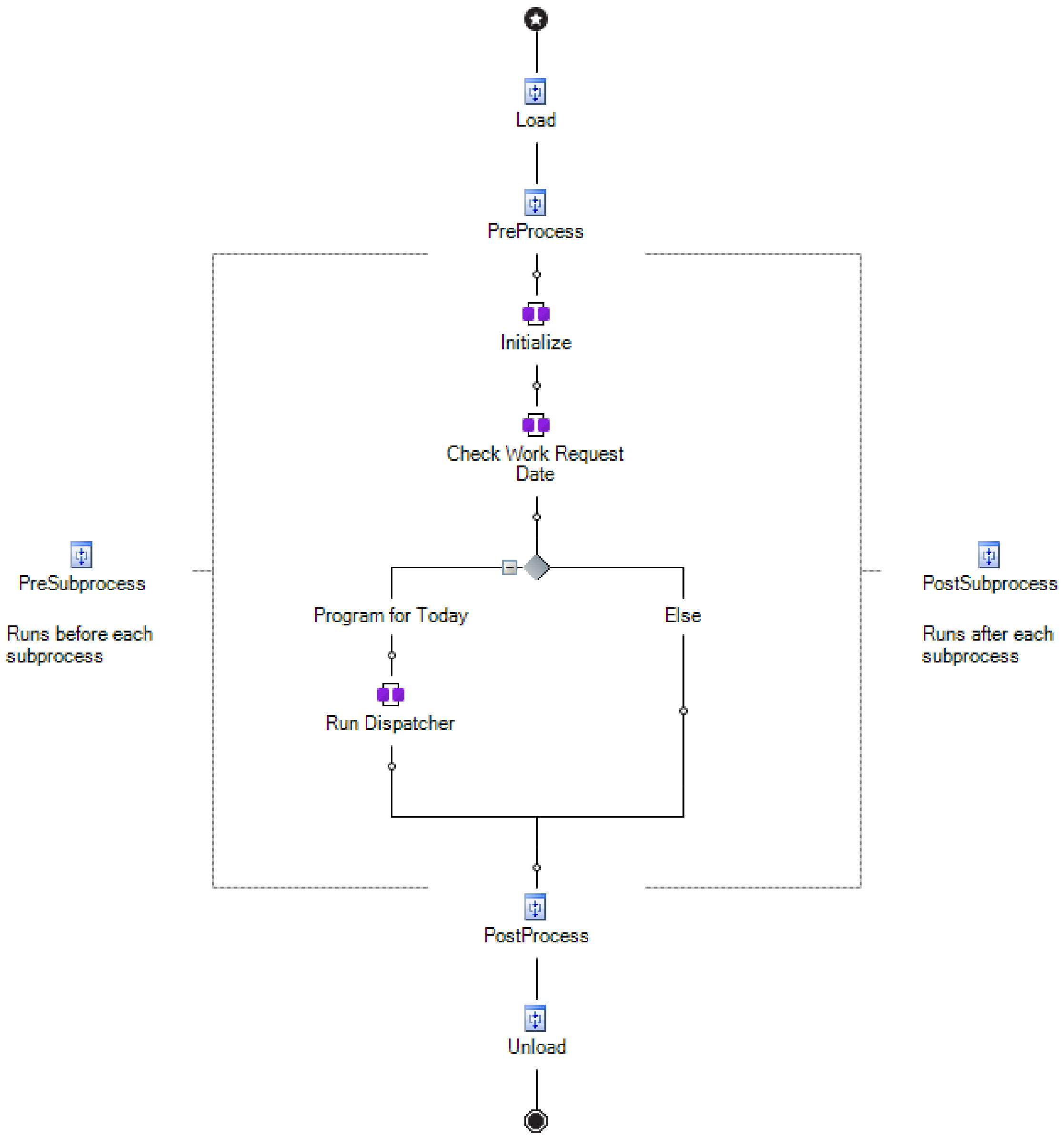

Diagram 3 Crane Program Scheduler

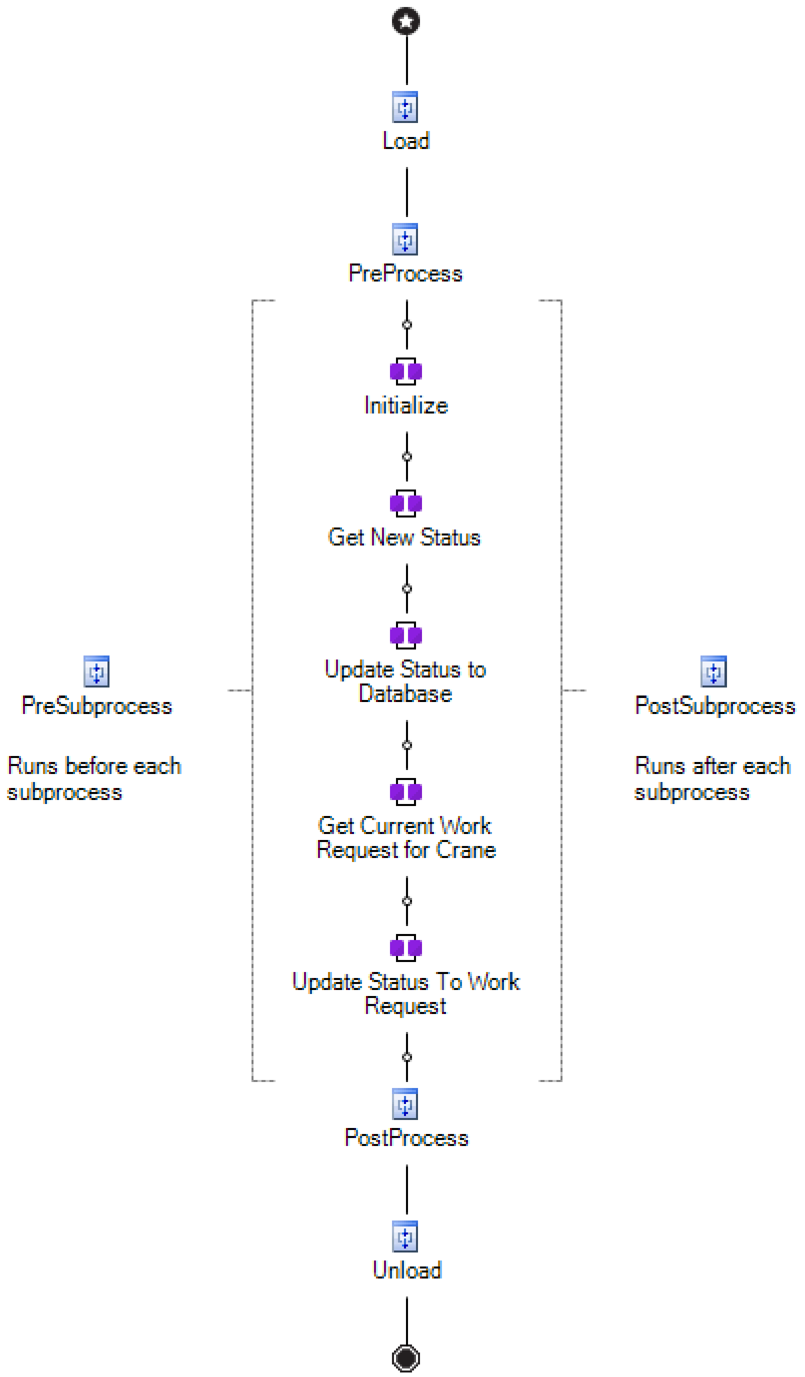

Diagram 4 Monitor schedule status workflow implementation

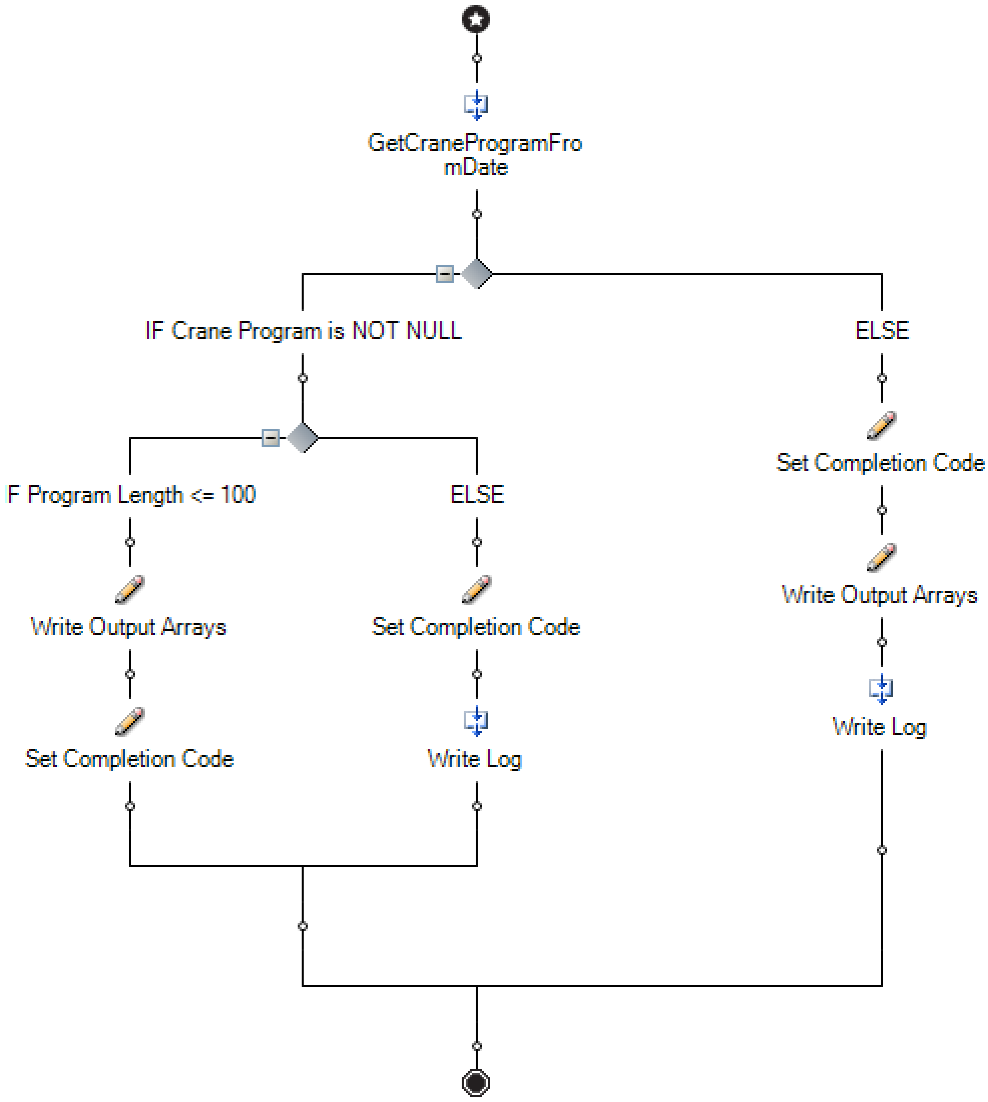

Diagram 5 Get crane program from database sub process

Appendix 7. Buffering and Data Collection Diagrams

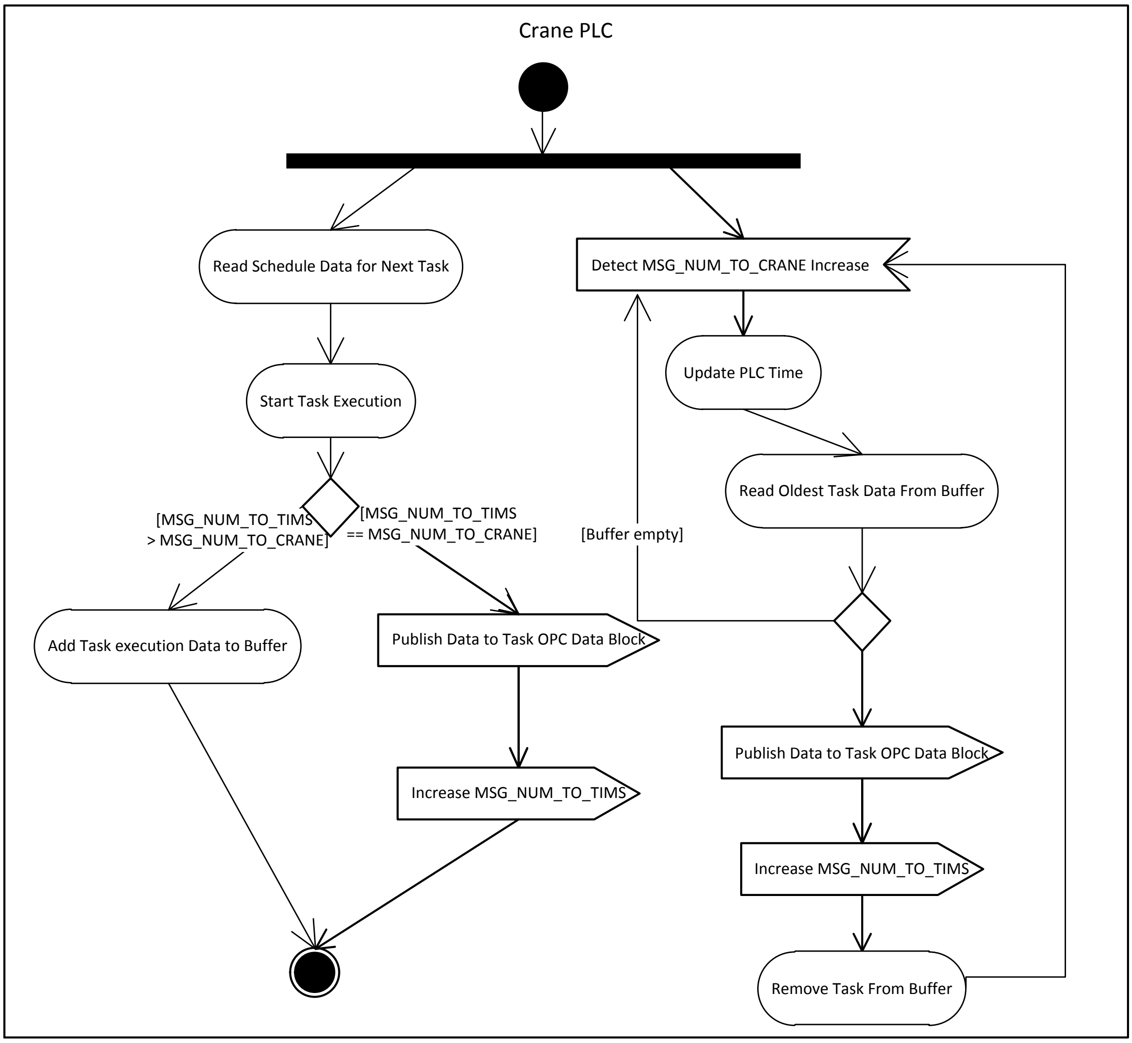

Diagram 1 Sequence diagram of the buffering logic in the PLC

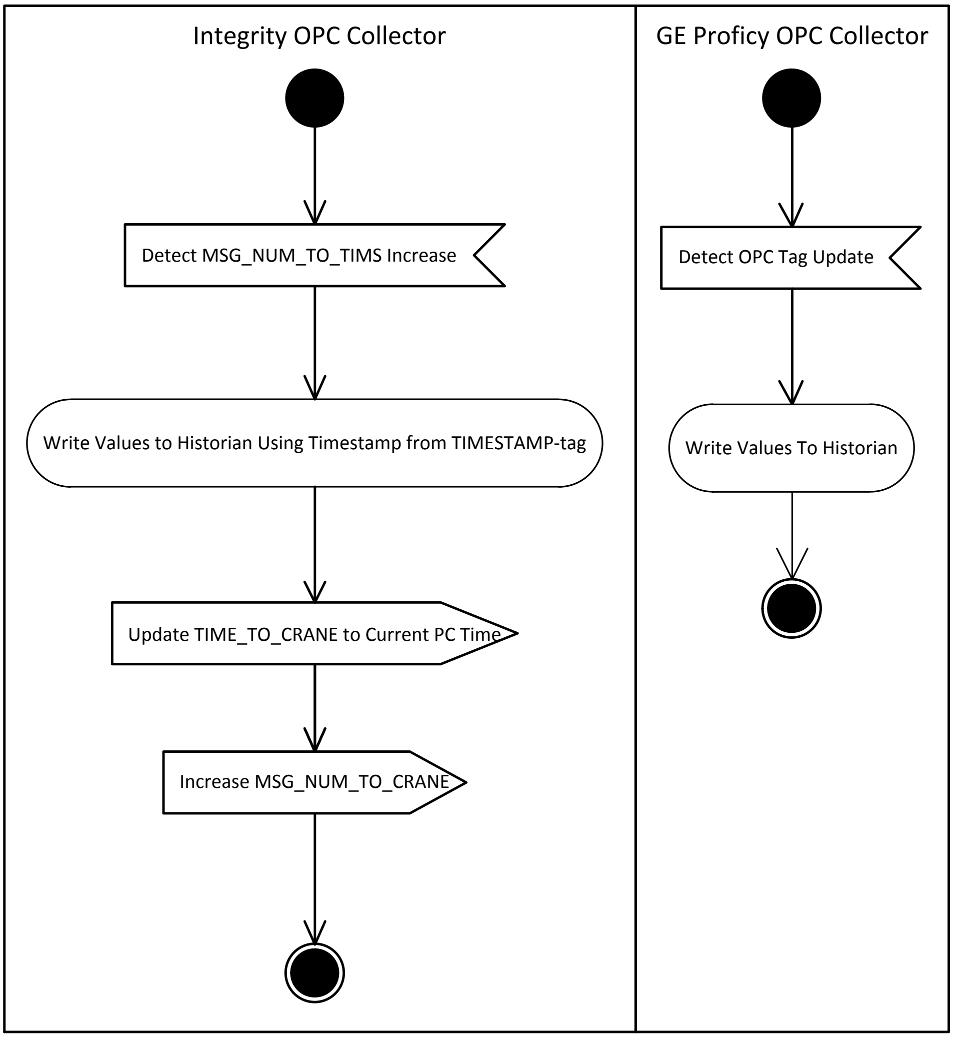

Diagram 2 Crane OPC Data Collection Sequences

Appendix 8. Test Plan for Crane Module Test

1. CONNECTIVITY TESTS

1.1. Crane PLC

Compare the data blocks at the Crane PLC to the datablock descriptions delivered for development purposes.

Table 1-1 Data Blocks

TESTED	DESCRIPTION	NOTES
	DBXX Maintenance data (cumulative distance counters etc.)	
	DBXX Time synchronization data	
	DBXX Machine and task data	
	DBXX Crane program – section data	
	DBXX Crane program – cell data	
	DBXX Crane program – task type data	

1.2. Procify Industrial Gateway Server (IGS) OPC Server

Validate PLC addresses and check the tag descriptions.

Table 1-2 Crane Task Data OPC Item Group

TESTED	OPC DATATYPE	DATABLOCK REF.	OPC NAME	DESCRIPTION
	Short	-REMOVED-	MSG_NUM_TO_TIMS	Crane task number. Counter that is increased when crane receives new target.
	Short	-REMOVED-	MSG_NUM_TO_CRANE	Message number from TIMS to PLC, aka. acknowledgement number.
	BCD Array	-REMOVED-	TIMESTAMP	Task timestamp. Time when task id was increased. Format [YYMM, ddhh, mmss]
	Short	-REMOVED-	ACTION_NUMBER	Action number: • 1 = Lift load • 2 = Leave load
	Long	-REMOVED-	WAITING_MANUAL	Waiting time during task in manual mode. In seconds.
	Long	-REMOVED-	WAITING_AUTO	Waiting time during task in auto mode. In seconds.
	Long	-REMOVED-	WAITING_POWER_OFF	Waiting time when crane power is off
	Long	-REMOVED-	TOTAL_TIME	Total time consumed during task in seconds.
	Short	-REMOVED-	GROUP_NUMBER	Group number where task hap-pened. 0 = machine or rack.
	Short	-REMOVED-	CELL_NUMBER	Cell number where task happened. • Two digit = cell • 1xy = Rack • 2xy = CSM • 3xy = APM • 4xy = ASW
	Long	-REMOVED-	HORIZONTAL_MOVE_AUTO	Horizontal movement in auto mode during the task in seconds.
	Long	-REMOVED-	HORIZONTAL_MOVE_MAN	Horizontal movement in manual mode during the task in seconds.

	Long	-REMOVED-	VERTICAL_MOVE_AUTO	Vertical movement in auto mode during the task in seconds.
	Long	-REMOVED-	VERTICAL_MOVE_MAN	Vertical movement in manual mode during the task in seconds.
	Short	-REMOVED-	LOAD_TYPE	Load type of the action: • 1 = Cath1 (EV – first cathodes) • 2 = Cath2 (EV) • 3 = Cath3 (EV) • 4 = Anode1 (EV – first anodes) • 5 = Anode2 (EV) • 6 = Anode3 (EV) • 9 = Cath all (ER - All cathodes) • 10 = AnodeL (ER – leaves first anode, lifts all cathodes) • 11 = AnodeR (ER – leaves last anode, lifts all cathodes) • 12 = AnodeB (ER – leaves first and last anode, lifts all cath-odes) • 13 = AnodeL (ER – leaves first anode, does not lift cathodes) • 14 = AnodeR (ER – leaves last anode, does not lift cathodes) • 15 = AnodeB (ER – leaves first and last anode, does not lift cathodes) • 16 = All (ER – empties a cell) • 0 = No load

Table 1-3 Machine Data OPC Item Group

TESTED	OPC DATATYPE	DATABLOCK REF.	OPC NAME	DESCRIPTION
	Short	-REMOVED-	RUN	Rung tag: • 0 = Automatic mode, in production • 1 = Target mode, in production • 2 = Manual mode • 3 = Auto mode and waiting • 4 = Auto mode and there is an alarm • 5 = Waiting Cell • 6 = Waiting load conveyor • 7 = Waiting ASW • 8 = Waiting APM
	Short	-REMOVED-	FAULT	Fault tag
	Short	-REMOVED-	SCHEDULE_STATUS	Schedule status: • 0 = Created • 1 = Updated • 2 = Loaded to PLC • 3 = In progress • 4 = Completed • 5 = Rejected
	BCD Array	-REMOVED-	TIME_TO_CRANE	Time synchronization, updated before every acknowledgement.

Table 1-4 Maintenance OPC Item Group

TESTED	OPC DATATYPE	DATABLOCK REF.	OPC NAME	DESCRIPTION
	Float	-REMOVED-	BRIDGE_CUM	Cumulative distance counter of the Bridge in km.
	Float	-REMOVED-	TROLLEY_CUM	Cumulative trolley distance counter in km.
	Float	-REMOVED-	HOIST_CUM	Cumulative hoist distance counter in km.

	Long	-REMOVED-	LIFT_COUNT	Cumulative lift count

Table 1-5 Scheduling Data OPC Item Group

TESTED	OPC DATATYPE	DATABLOCK REF.	OPC NAME	DESCRIPTION
	Short Array	-REMOVED-	GROUP_ARRAY	Array of the groups in one program/recipe.
	Short Array	-REMOVED-	CELL_ARRAY	Array of the cells in one program/recipe.
	Short Array	-REMOVED-]	TASK_ARRAY	Array of the task types (load types) in one program/recipe.

2. FUNCTIONAL TESTS

2.1. Proficy Historian

Data from the IGS OPC server is collected to the Proficy Historian database via standard Historian OPC Collector and custom made OPC collector. The custom made OPC collector is responsible for collecting buffered crane task data. During the tests it should be verified that the collectors works as expected and collect data from the OPC server and store it to the historian database. The standard OPC Collector should be in running state and the custom collector should acknowledge every task to the OPC server and update the time synchronization tag.

Table 2-1 OPC Collectors

TESTED	DESCRIPTION	NOTES
	Time synchronization is working	
	Task acknowledgement is working	
	Data is populated to all configured historian tags	
	IGS OPC server collector is displayed in the Proficy Historian administrator tool.	
	IGS OPC server collector is running.	

Table 2-2 Task Data Historian Tags

TESTED	DESCRIPTION	NOTES
	TIMS.CRANE.TASK.MSG_NUM_TO_TIMS	
	TIMS.CRANE.TASK.ACTION_NUMBER	
	TIMS.CRANE.TASK.TOTAL_TIME	
	TIMS.CRANE.TASK.GROUP_NUMBER	
	TIMS.CRANE.TASK.CELL_NUMBER	
	TIMS.CRANE.TASK.LOAD_TYPE	

Table 2-3 Machine Data Historian Tags

TESTED	DESCRIPTION	NOTES
	TIMS.CRANE.MACHINE.RUN	
	TIMS.CRANE.MACHINE.FAULT	

Table 2-4 Maintenance Historian Tags

TESTED	DESCRIPTION	NOTES
	TIMS.CRANE .MAINTENANCE.BRIDGE_MOTOR_CUM	
	TIMS.CRANE .MAINTENANCE.TROLLEY_CUM	
	TIMS.CRANE .MAINTENANCE.HOIST_CUM	
	TIMS.CRANE .MAINTENACE.LIFT_COUNT	

2.2. Plant Applications

This part of the test is optional, but will be executed if there is time.

TESTED	DESCRIPTION	NOTES
	DOWNTIME REASON TREE Check downtime reasons.	
	CRANE TASK USER DEFINED EVENT Check event triggering when new data is inserted to the historian	
	DOWNTIME EVENT Machine Status or Alarm Number change should trigger down-time event.	
	EFFICIENCY REPORT DATA Efficiency data for reporting purposes should be collected every hour, day, week, month and year.	
	MAINTENANCE REPORT DATA Quality data for reporting purposes should be collected every hour	
	TASK REPORT DATA Task report data should be collected with every crane task UDE. In practice every time the MSG_NUM_TO_TIMS is updated.	

2.3. SQL Server

TESTED	DESCRIPTION	NOTES
	CRANE PROGRAM DATA Check Crane program data in database	
	CRANE PROGRAM STATUS Check Crane program status changes in database	
	TANKHOUSE SETTINGS Check tankhouse settings, for example number of cells in particular section etc.	

2.4. Proficy Workflow

TESTED	DESCRIPTION	NOTES
	EVENTS STATUSES ARE ACTIVE Check OPC Data Change Events statuses	

	CRANE PROGRAM STATUS Check Crane program status changes in database	
	COMMUNICATION WITH INTEGRITY BACKEND Check communication with Integrity WCF Services and Database	
	COMMUNICATION WITH IGS OPC Server	

2.5. Reporting

This part will be executed at a later date. When the data is collected to the historian the evaluation can be done in offline environment. All other but crane reports will be excluded from the tests.

2.6. Integrity Backend

Integrity backend is a collection of Windows Service hosting WCF servers. The reporting service contains queries to retrieve the data needed for the reports. These queries should be tested and approved using Visual Studio at developer's PC.

2.7. Integrity Silverlight Application

Check each report by using the Silverlight GUI.

2.8. Crane Program Editor

Crane program editor is a part of the Silverlight UI and the main function of the editor is to create daily crane harvest programs. The editor allows user to enter template harvest schedules which can be used to create actual crane programs.

TESTED	DESCRIPTION	NOTES
	CRANE PROGRAM EDITOR MAIN VIEW Check the operation of program editor	
	ADD CRANE TASK VIEW	
	TASK OPERATION TYPE VIEW	
	ADD OPERATION TYPE VIEW	
	UPLOAD CRANE PROGRAM	
	COMMUNICATION WITH INTEGRITY BACKEND Check communication with Integrity WCF Services and Database	

3. TEST CASES

Following test cases will be executed during testing. Cases consist of different sequences of operations and crane program definitions. Some tests are abstract because the final operation logic of the crane is not absolutely clear. The testing aims to reveal problems in communication and backend operation logic and does not focus on bugs and problems that might be found in Silverlight UI.

3.1. Crane Program Editor

Create following operation types:

TESTED	NAME	SECTION	CELL START	CELL END	Harvest Type	NOTES
	Test 1	1	1	16	Anode & cathode reload (all)	
	Test 2	24	16	1	Anode & cathode reload (leave first anode)	
	Test 3	12	1	1	Anode & cathode reload (leave both side anodes)	
	Test 4	24	12	4	Anode reload (leave last anode)	
	Test 5	4	2	15	Anode & cathode reload (leave last anode)	
	Test 6	2	1	16	Cathode reload	
	Test 7	2	1	16	Anode & cathode reload (all)	
	Test 8	2	1	16	Cathode reload	
	Test 9	1	15	16	Anode & cathode reload (leave last anode)	

Create following crane programs:

TESTED	NAME	SECTIONS	OPERATION TYPES	NOTES
	Program 1	1	Test 1	
	Program 2	24, 1, 4	Test 2, Test 9, Test 5	
	Program 3	12	Test 3	
	Program 4	2, 2, 2	Test 6, Test 7, Test 8	
	Program 5	24	Test 4, Test 9	

A program is updated to the crane by pressing the update button. This update will be done manually after a new program is created. Crane operator should try to reject and accept different programs in different situations.

3.2. Machine Data Test

Machine data test are executed to trigger downtime events in Plant Applications. Crane operator should try to enter different operator assist faults and if possible some other machine failure faults.

3.3. Connectivity Tests

Connection interruptions should be simulated to test systems ability to recover in a controlled manner. This could be done by disconnecting Ethernet cables etc.