TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MARKUS HIRVONEN

STREAMLINING MANUFACTURING DATA INTEGRATION

Master of Science Thesis

Examiner: David Hästbacka
Examiner and topic approved
31st May 2017

ABSTRACT

Markus Hirvonen: Streamlining manufacturing data integration
Tampere University of Technology
Master of Science Thesis, 63 pages
November 2017
Master's Degree Programme in Automation Engineering
Major: Information Technology in Automation
Examiner: David Hästbacka, D.Sc. (Tech.)

Keywords: agile, analysis, CNC, data, IIoT, industrial, Industry 4.0, information, integration, internet, IoT, JIT, lean, machine tool, machinery, machining, MTConnect, NC, network, OPC UA, plug and play, plug and produce, remote, smart, things, UPnP

Industrial connectivity is lagging years behind consumer device connectivity. Both management and technological trends demand more data for decision making and one of the major hindrances is poor connectivity interfaces. Open Platform Communications Unified Architecture (OPC UA), a general industrial protocol for data transfer and information modelling, seeks to rise above first-party manufacturer data standards by providing a common way of implementation for industrial device connectivity. MTConnect, a free open-source data standard for mainly numerical control (NC) machine tools, seeks to do the same in the manufacturing industry. Can they both be used to ease the integration of devices in the comparable manner that Universal Plug and Play (UPnP) has successfully done for consumer devices?

This thesis explores the differences of OPC UA and UPnP, the whys and how's of manufacturing data collection focusing on NC machine tools from both management and technological perspectives, MTConnect and OPC UA capabilities, and finally seeks to answer the fore mentioned integration ease question. This pursuit is driven by global megatrends like Industry 4.0, Smart Manufacturing, Industrial Internet, Agile Manufacturing, Business Intelligence, Lean and JIT. As a part of this thesis, a prototype application using MTConnect and OPC UA is made to investigate if they have brought industrial data transfer standardization as far as UPnP has done in the consumer space.

It was found that OPC UA and UPnP share many aspects technologically, but the differences are found in the depth and spread of standardization. Multi-device intercommunication is inherently a part of UPnP, but is something that has been largely neglected from OPC UA until 2015. The OPC UA - MTConnect companion specification allows easier integration of MTConnect devices into a factory-wide OPC UA network, but in a smaller environment MTConnect is easier to implement alone without OPC UA. The prototype proved that connectivity between OPC UA and MTConnect is effective albeit more time-consuming than implementing a mere MTConnect integration in a situation, where industrial devices are only outputting MTConnect data.

TIIVISTELMÄ

Markus Hirvonen: Tuotannon dataintegraation virtaviivaistaminen
Tampereen teknillinen yliopisto
Diplomityö, 63 sivua
Marraskuu 2017
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Automaation tietotekniikka
Tarkastaja: TkT David Hästbacka

Avainsanat: agiili, analyysi, CNC, data, esineiden, etä, IIoT, Industry 4.0, integraatio, internet, IoT, JIT, kone, lean, MTConnect, NC, numeerinen, ohjaus, OPC UA, teollinen, tieto, tuotanto, työstö, työstökone, UPnP, valvonta, verkko, älykäs

Teollisen liitettävyyden standardointi on vuosia jälkijunassa verrattuna kuluttajalaitteiden liitettävyyteen. Niin hallinnolliset, kuin teknologisetkin trendit vaativat enemmän dataa päätöksentekoa varten ja tätä hidastaa eniten huonot liitettävyysrajapinnat. Open Platform Communications Unified Architecture (OPC UA) on teollisuuden yleisprotokolla, joka pyrkii nousemaan valmistajakohtaisten datastandardien yläpuolelle, tarjoten yhteisen toteutustavan teollisten laitteiden liitettävyyteen. MTConnect, ilmainen ja avoin numeerisesti ohjattujen työstökoneiden datastandardi pyrkii tekemään samoin valmistavassa teollisuudessa. Voiko näitä kahta yhdistää siten, että laitteiden integrointi helpottuu verrattavalla tavalla kuin Universal Plug and Play (UPnP) on onnistuneesti toteuttanut kuluttajalaitteiden kanssa?

Tämä diplomityö tutkii miksi ja miten valmistavassa teollisuudessa kerätään dataa, keskittyen NC-koneisiin niin hallinnollisesta, kuin teknologisestakin näkökulmasta. Työssä tutkitaan MTConnect:n ja OPC UA:n mahdollisuuksia sekä pyritään vastaamaan edellä esitettyyn kysymykseen laitteiden integroinnista. Tätä pyrkimystä tukee maailmanlaajuiset megatrendit kuten Industry 4.0, Smart Manufacturing, Industrial Internet, Agile Manufacturing, Business Intelligence, Lean ja JIT. Diplomityön osana tehdään prototyyppisovellus käyttäen MTConnect:a ja OPC UA:ta, jonka perusteella päätellään, ovatko nämä teollisten laitteiden datastandardointi tullut yhtä pitkälle kuin mitä UPnP kuluttajapuolella.

Työssä havaittiin että OPC UA ja UPnP jakavat monia teknologisia osia ja että niiden eroavuudet löytyvät pääosin standardisoinnin syvyydestä ja kattavuudesta. Monen laitteen kanssa kommunikointi on sisäänrakennettua UPnP:ssä, kun taas OPC UA:sta tämä mahdollisuus on paljolti laiminlyöty vuoteen 2015 saakka. OPC UA:n ja MTConnect:n välinen kumppanispesifikaatio auttaa laitteiden integraatiota tehtaanlaajuisessa verkossa, mutta pienemmässä ympäristössä yksin MTConnectin käyttö on helpompaa. Tuloksista selvisi, että OPC UA:n ja MTConnect:n välinen liitettävyys on toimivaa, mutta enemmän aikaa vievää verrattuna pelkkään MTConnect-integraatioon tilanteessa, jossa teolliset laitteet käyttävät syöttävät vain MTConnect:n mukaista dataa.

PREFACE

This thesis is done for a InSolution Oy, a Finnish company based in Tampere, who have graciously allowed me to do my thesis on the side of my work. Thanks to Juha Katajisto and Mauri Ranta for company oversight, technical guidance and business insight. Thanks also to Arttu Tamminen for technical guidance and help with interfaces. Thanks to the examiner of this thesis, Dr. David Hästbacka, for being patient with me.

Cheers to all my friends whom I have neglected during the process.

I would like to thank my spouse Raisa Saarijärvi for being awesome and encouraging during the process, while reminding me to take time for rest and relaxation during this thesis.

Tampere, 25.11.2017

Markus Samuel Hirvonen

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

AMT	Association of Manufacturing Technology
AMQP	Advanced Message Queuing Protocol
ARP	Address Resolution Protocol
CNC	Computer Numerical Control
CTF	Critical Tool Failure
DNC	Direct Numerical Control
DS	Design Science
GUI	Graphical User Interface
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IT	Information Technology
LDS	Local Discovery Server
MT	Machine Tool
MTU	Maximum Transmission Unit
MQTT	Message Queue Telemetry Transport
NC	Numerical Control
OCF	Open Connectivity Foundation
OLE	Object Linking and Embedding
OPC	Open Platform Communications
OPC UA	OPC Unified Architecture
OS	Operating System
OSI	Open Systems Interconnection
PnP	Plug and Play
MT	Machine Tool
REST	Representational State Transfer
SCADA	Supervisory Control and Data Acquisition
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
TUT	Tampere University of Technology
UA	Unified Architecture
UA TCP	Unified Architecture Transmission Control Protocol
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WP	Work Piece
XML	Extensible Markup Language

1. INTRODUCTION

The manufacturing industry has been dragging its' feet in an age of change as customers demand more product customization and companies demand more monitoring capabilities (Booth 1995; Stjerna 2017). At the same time, they are haunted by an old problem of consumer electronics, where networking adapters had to be custom-made for each different vendors' equipment. Universal Plug and Play (UPnP) brought differing standards together under an umbrella and has made connecting devices simple in the consumer space (Miller et al. 2001). By 2015, the adoption rate of UPnP has reached over 2 billion devices, available across all major operating systems like Windows, iOS, MacOS, Linux and Android. (Lofgren 2015).

Problem is that the same level of integration ease has not been achieved in the manufacturing industry due to lack of consensus and standardization of discovery and data transfer between vendors, which Schiele et al (2010) mentions as one of the main factors for interoperability (Figure 1).

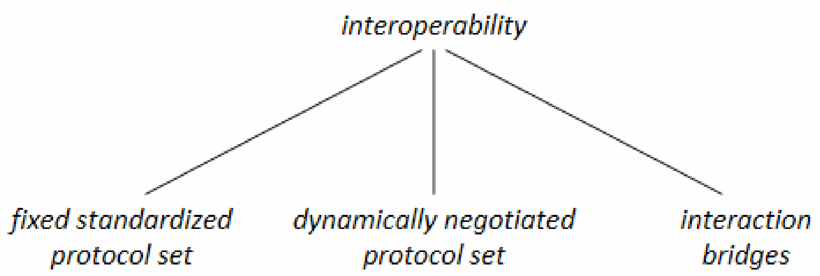


Figure 1. *Possibilities for interoperability (Schiele et al. 2010).*

This essentially slows down the cycle of technological development in the industrial sector since adapters must be custom-built for each different data protocol (Lindström 2015). This makes the projects more expensive and effectively hinders innovative commitments in an environment where pay for performance is valued (Conlin 2003).

The Business Intelligence (BI) (Gilad & Gilad 1986) space has long been swamped with numerous theses and scientific papers (Vuori 2005; Lawton 2006; Ramakrishnan et al. 2012; Ponomarjovs 2013; Tyrväinen 2014; Keskinen 2017; Kiran 2017) on how to process vast amounts of data to make better decisions on business operations. Furthermore, business intelligence has been taking roots in all large industries due to competitive and institutional pressures, resulting in data collection and its' analysis (Ramakrishnan et al. 2012). Many mention integration difficulties as one of the key problems. (Ziegler & Dittrich 2004; Lawton 2006; Haas 2006; Jarke et al. 2014). Accordingly, many researches have also been committed to addressing the integration problem, albeit with application engineered solutions (Haas 2006; Kalibataitė 2014).

Many have come to the conclusion that the Open Platform Connectivity Unified Architecture (OPC UA) might be the answer, increasingly in the recent years (Hannelius et al. 2008; Palm et al. 2015; Lindström 2015; Rouhiainen 2015; Hoefling et al. 2015; Seilonen et al. 2016; Iatrou & Urbas 2016; Kozar & Kadera 2016; Rentschler et al. 2016; Hoffmann et al. 2016), likely due to focused funding from various global, regional and national actors. The OPC Foundation seeks to bring protocol consensus to heavy industries with the platform independent, secure and extensible OPC UA and its' support for object-oriented information modelling capabilities (Mahnke et al. 2009). This allows separate industries to keep their own specific data standards such as the extensible markup language (XML) based MTConnect in the specific segment of numerically controlled (NC) machine tools (Che & Liu 2013).

Ideally, OPC UA could work as a discovery and service layer between any two industrial devices with minimum effort, be it a MTConnect device or a TwinCAT programmable logic controller (PLC); same way as UPnP works with consumer devices – plug two computers to a router via an ethernet cable and you can connect and share data with no configuration needed outside of security permissions (Miller et al. 2001). Even this industrial and consumer separation is crumbling with Microsoft adding OPC UA support to Azure cloud services (Microsoft introduces new open-source cross-platform OPC UA support for the industrial Internet of Things) and OCF introducing UPnP+ which aims to expand the user base over to industries and beyond (Lofgren 2015).

1.1 Research questions and limitations

This thesis aims to answer three research questions: First, this thesis will explore UPnP and OPC UA differences in data transfer and discovery. Secondly, investigation is made on how MTConnect data can be mapped in OPC UA. Thirdly, assessment is done whether MTConnect and OPC UA allows ease of manufacturing information integration comparable to the UPnP.

Since the base of this thesis is founded on a business case from a company working with industrial informatics and automation, there are some limitations regarding the examined technologies and industrial environment. Integrating MTConnect machine tools into a data analysis software is the main priority for the business case. This thesis does not address other possible technologies as they have already been addressed by a previous thesis to InSolution Oy made by Rouhiainen (2015), which pointed this thesis to OPC UA. Finding a way to utilize OPC UA would bring added value to the results as it was highly recommended by the fore mentioned thesis.

1.2 Methodology

Design Science (DS) is a methodology that is fundamentally a problem-solving process that creates a viable solution as a result. It seeks to create innovative ways in which the analysis, design, implementation, management and use of information systems can be efficiently accomplished. (Hevner et al. 2004). Design science has seven guidelines (Table 1) that should be addressed in some manner for a design science research to be complete.

Table 1. *Design Science Research Guidelines (Hevner et al. 2004).*

Guideline	Description
#1: Design as an Artifact	Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
#2: Problem Relevance	The objective of design science is to develop technology-based solutions to important and relevant business problems.
#3: Design Evaluation	The utility, quality and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
#4: Research Contributions	Effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations and/or design methodologies.
#5: Research Rigor	Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
#6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
#7: Communication of Research	Design science must be presented effectively both to technology-oriented as well as management-oriented audiences.

This thesis implements design science guidelines in following ways:

- A prototype application will be made
- An overview of the relevant technological and business trends is explored to emphasize the problem relevance
- The prototype application will be evaluated in its’ ease of implementation
- Results must be verifiable
- Rigorous methods are used and referred to in design and evaluation
- Prototype design and implementation are done as an iterative search process
- The results are communicated in both technological and management terms.

In response to DS guideline seven, this thesis approaches the topic from both technological and management perspectives. Machine tools and related connectivity technologies, UPnP, OPC UA and MTConnect are introduced as well as current management trends that drive modern business decision making.

A prototype application will be made to address DS guideline one. The findings from this prototyping will be concrete evidence of the viability of OPC UA in integrating manufacturing devices to BI applications. Prototype development starts by documenting common requirements in numerically controlled machine tools. In each step of the way we evaluate if the examined has potential to reduce complexity in integrating devices to a data collection application by comparing it to the widely successful UPnP.

2. INTERCONNECTIVITY NEEDS IN THE MANUFACTURING INDUSTRY

This chapter seeks to answer DS guideline two with a nod to guideline seven, problem relevance and research communication in technological and management terms. Within are the reasons why the topic of this thesis is important and relevant to current business problems. The technological side addresses information integration, which Haas (2006) identifies as the database community's answer to the management problems.

2.1 Numerical control machine tools

This section introduces the context of this thesis, manufacturing, and further probes into the focus of this thesis, numerical control machine tools. El-Hofy & Youssef (2008) describes manufacturing as "the industrial activity that changes the form of raw materials to create products.". Figure 1 depicts several common manufacturing practices including machining and its' traditional and non-traditional processes.

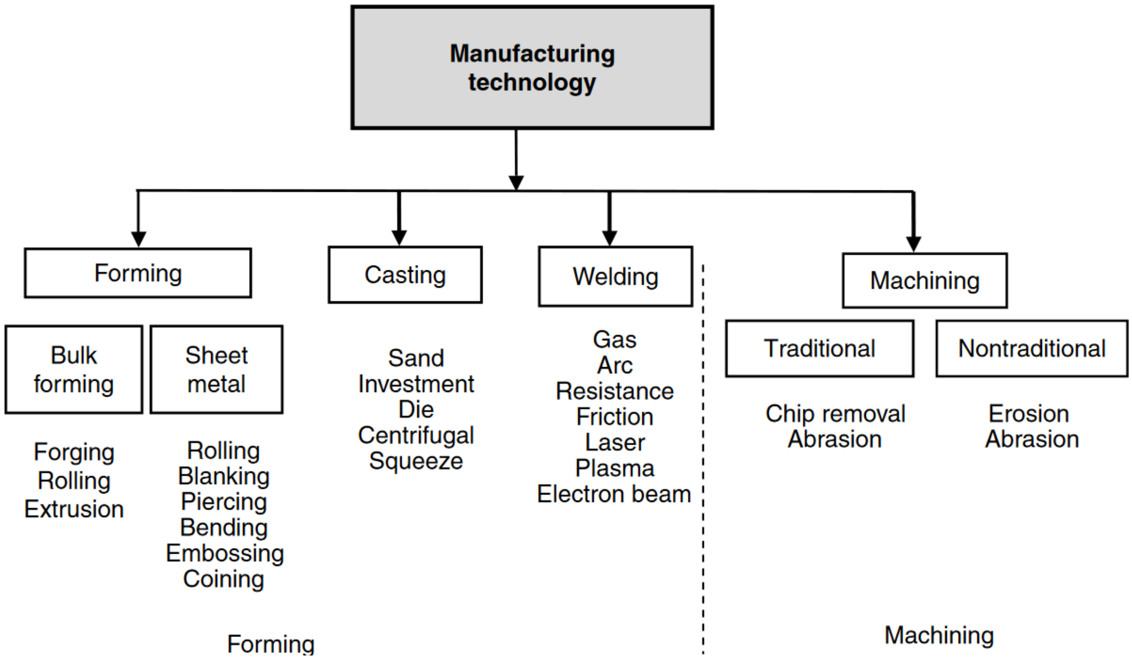


Figure 1. *Classification of manufacturing processes (El-Hofy & Youssef 2008).*

Metal cutting machines, also known as machine tools, remove unwanted material from a workpiece to make a product of exact size, shape and surface quality. Many of the parts made with additive or molding manufacturing methods are often followed by a machining finishing. This is emphasized by Figure 2. (El-Hofy & Youssef 2008).

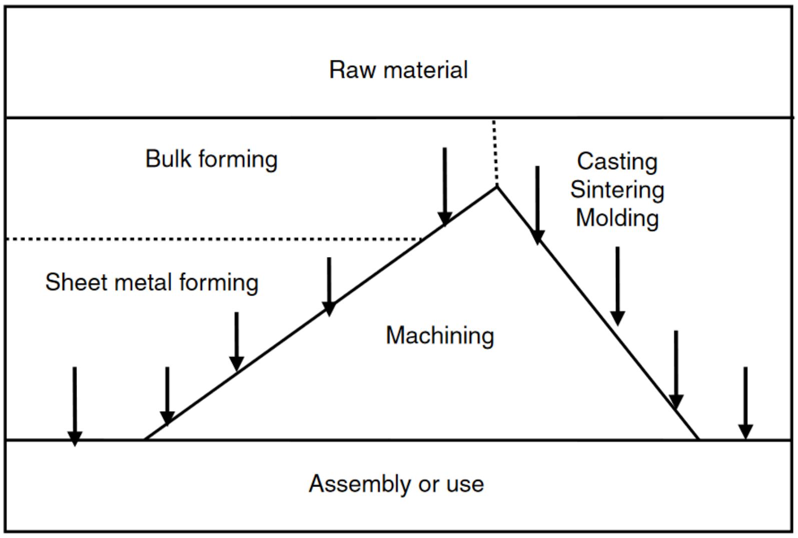


Figure 2. *Definition of manufacturing (El-Hofy & Youssef 2008).*

This practice of removing unwanted material from objects was first adopted by prehistoric animals with tools made from bone, stick or stone. Humans later evolved to use metal tools, water, steam and even electricity to aid in this manufacturing effort. The first actual machine tool, a drilling machine made of wood, was invented by the Egyptians back in 4000 BC. More advanced tools were later built over the ages by visionaries such as Leonardo da Vinci, Maudslay, Sellers, Fitch, Spencer, Whitney, Brown, Sharpe, Fellows, Pfauter etc. (El-Hofy & Youssef 2008). Figure 3 depicts various machining methods.

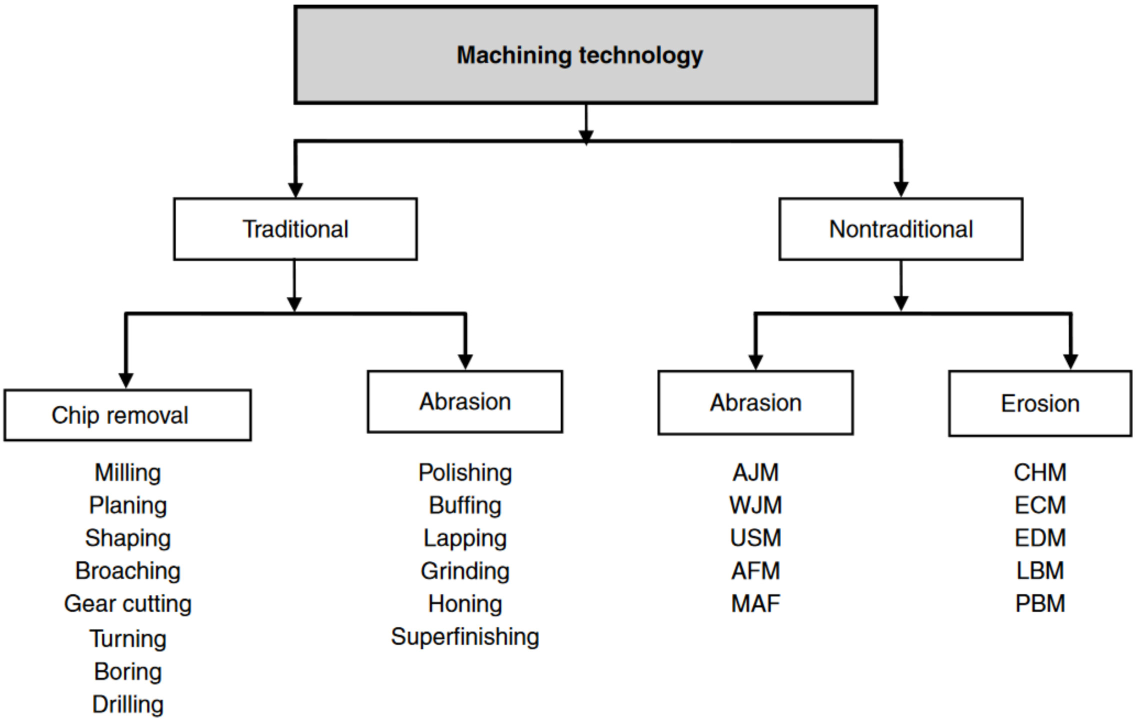


Figure 3. *Classification of machining processes (El-Hofy & Youssef 2008).*

Further developments in automatic mechanisms, such as numerical control (NC), computer numerical control (CNC) and direct numerical control (DNC), and measuring devices, raised product accuracy and reduced manual labor along with costs.

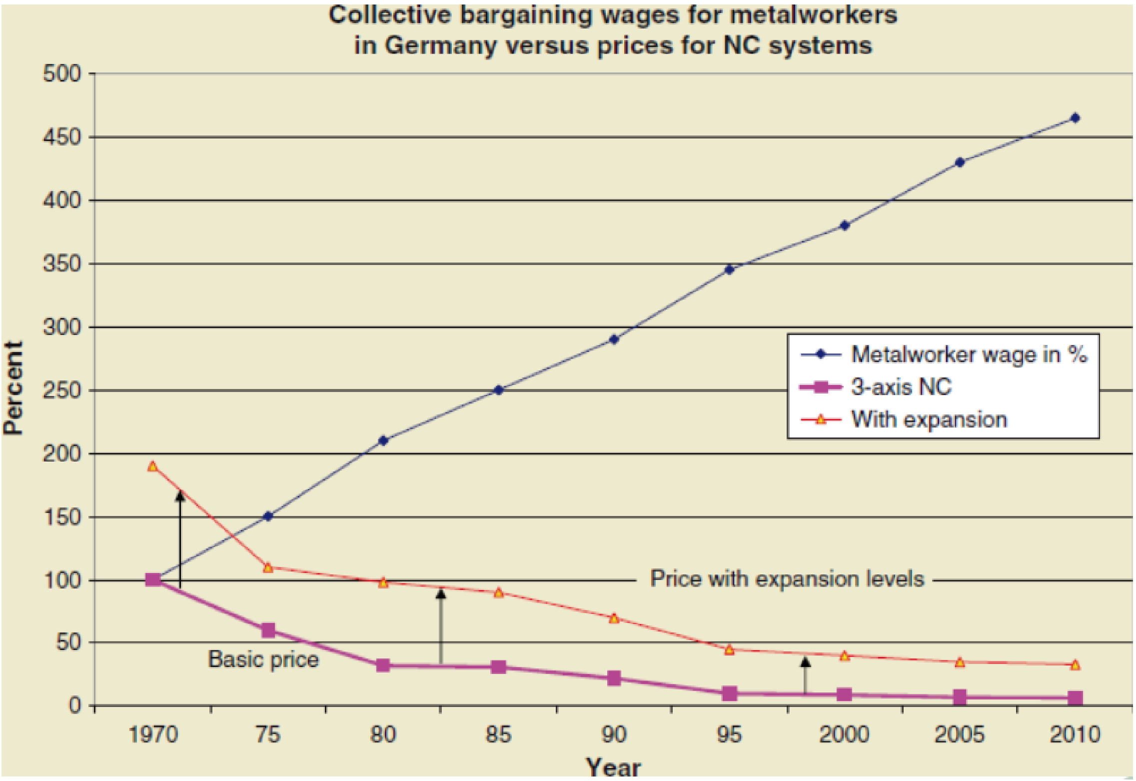


Figure 4. *Price trends for numerical controllers with a constantly increasing range of functions in comparison with trends for collective-bargaining wages in the German metalworking industry (Kief 2011).*

Numerical control allowed the use of pre-recorded recipes. The first NC machine tool was demonstrated in Massachusetts Institute of Technology (MIT) during year 1952. (El-Hofy & Youssef 2008; Suh et al. 2008). The first NC systems were built using relays and were hard-wired. As electronics parts developed and got cheaper, the performance of NC machine tools increased greatly (Figure 5) along with its' hard-wired logic functions turning into software that can easily be copied (Kief 2011).

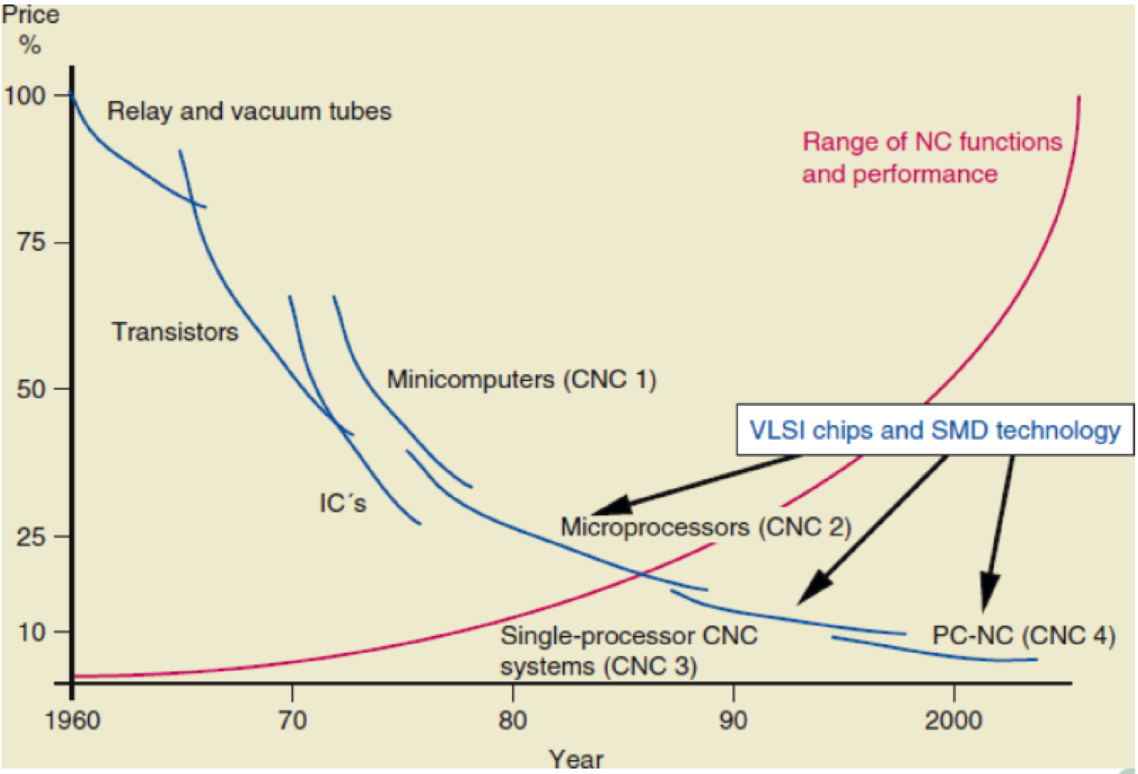


Figure 5. *Development of prices for numerical controllers with the use of electronic components with ever-increasing levels of integration (Kief 2011).*

The most common types of NC machine tools are (El-Hofy & Youssef 2008; Suh et al. 2008):

- NC drilling machines, available in wide range of types and sizes
- NC milling machines, that produce contours and curved surfaces
- NC turning machines, that produce cylindrical shapes
- NC machining centers, that can change tools automatically

As technology has progressed, NC technology has allowed the construction of Flexible Manufacturing Systems (FMS) that connects the machines with robots, autonomous guided vehicles (AGV), automated warehouses and computers (Suh et al. 2008).

The task flow for creating the part can be summarized into offline, online and post-line sections (Figure 6). In offline, decisions are made on what kind of WP should be made and how. In online, the WP is machined by reading programs from memory and controlling the axes' movement. Possible errors such as tool breakage, compensation of thermal deformation et cetera are handled in this phase. Post-line operations include computer-aided inspection. Some systems measure the completed part and compare it with the model to compensate for mistakes.

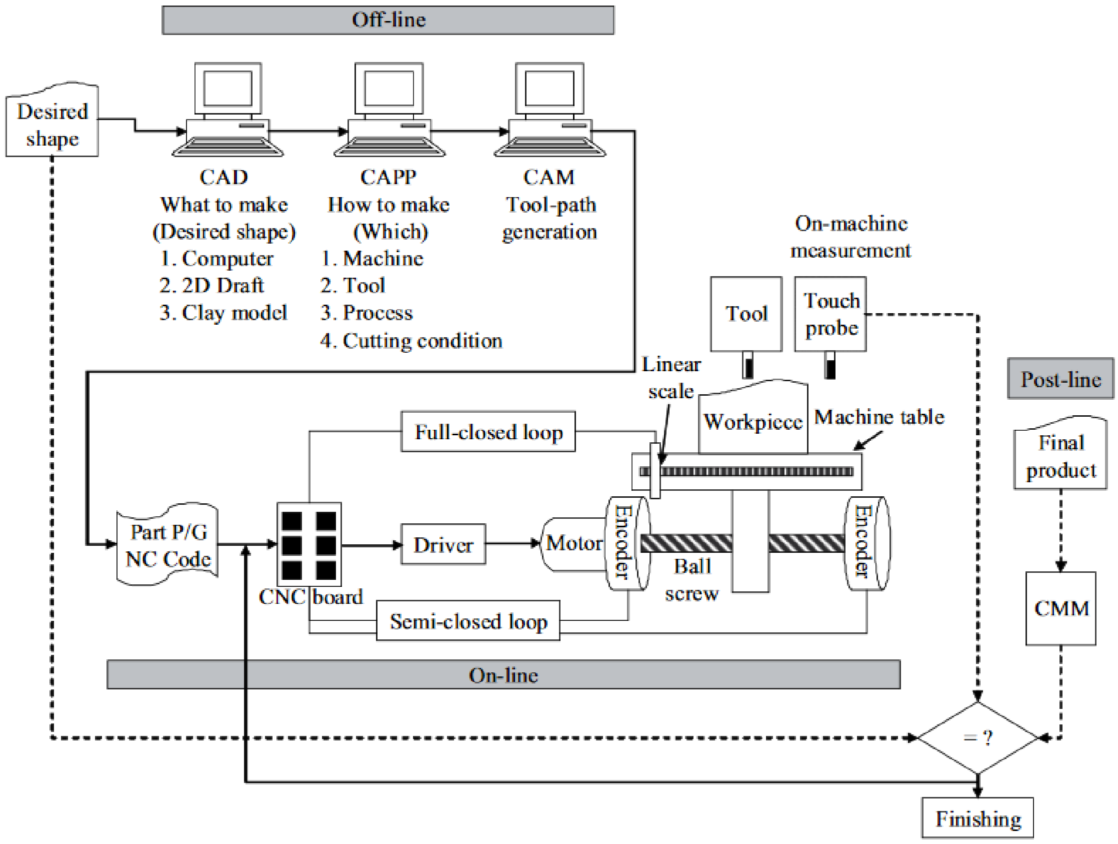


Figure 6. *The architecture of NC machine tools and machining operation flow (Suh et al. 2008).*

Control of a modern NC machine tool is done with three independent control loops for each of the three axes. These can be either a semi-closed loop control, closed loop control or hybrid loop control depending on where the position data is collected from. A typical control loop of an NC machine is depicted in Figure 7.

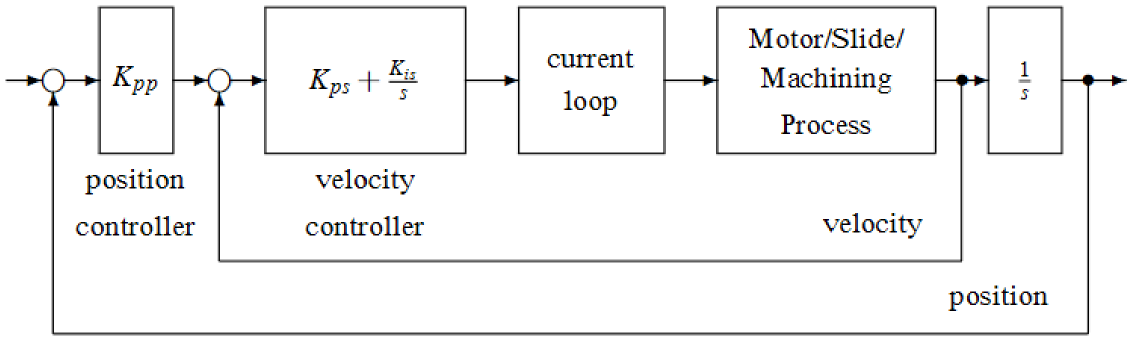


Figure 7. *Control system block diagram of a NC machine tool (Suh et al. 2008).*

2.2 Management trends

As described in the introduction, at the base of every business decision there are always three goals: survival, growth and profitability (Pearce 1982). This section covers manufacturing trends that drive modern industrial demands and decision making. Most of these trends originate from the 1970s where western industries struggled to understand and compete with the suddenly successful Japanese companies that were rapidly taking over the world export market (Corrêa 2001).

2.2.1 Agile Manufacturing

Agile Manufacturing is not a new term. It was envisioned in 1991 by a think tank at Lehigh University, made of industry executives from Chrysler, Motorola, AT&T and 10 other large manufacturing companies (Think Tank Suggests Agile Manufacturing for U.S. Success, 1992). Agility as a business concept was defined by Goldman et al (1995) as being about winning competition and profits with customers as the centrepieces. In manufacturing, this could be solidified into four key strategies:

- Enriching the customer
- Co-operating to enhance competitiveness
- Organising to master change and uncertainty
- Leveraging the impact of people and information

Gunasekaran et al (2001) explain that Agile Manufacturing is often misunderstood as lean or flexible manufacturing. It is a concept that combines the building blocks of agility such as methods and philosophies to create production capability that can simultaneously produce in volume and variety. Inman et al (2011) have researched the relations between agile manufacturing, JIT and Lean, finding many of their parts supportive of each other, like JIT purchasing. In production however, their similarities were non-significant.

Corrêa (2001) further elaborated on Goldman et al (1995) strategies, bringing more focus on planning for change:

- Flexibility is central; and so is change control
- Breaking barriers through customer-supplier negotiation
- The time-phased approach
- Proactivity achieved by using scenarios: the role of “contingency models”
- The consideration of dynamic trade-offs and dynamic paths of improvement
- The re-planning process – triggered by relevant events and time.

2.2.2 Business Intelligence

The collection and analysis of data has become ever more significant and prevalent in today's industries as the line between competitive advantage and bankruptcy is getting ever hazier. Data as a resource has only become widely understood in the end of 1990s and many of its' concepts are still young and unstable. Collected data can be structured into information, which can be analysed and compared with prior experience to create knowledge. Knowledge as a form of processed data is the most easily accessible information resource a human can utilize. (Laihonen et al. 2013). Business intelligence (Gilad & Gilad 1986) is one of such practices meant to improve business' decision making. It focuses on five tasks: collection of data, evaluation of data validity and reliability, analysis, storage of data and dissemination. Gilad & Gilad (1986) wrote about its' silent revolution in companies in the United States of America already in 1986.

2.2.3 Just-In-Time (JIT)

According to Stevenson (2012), JIT is widely viewed to be a system for scheduling work in such a way that the number of work-in-progress and the volume of inventory are low. By some accounts, JIT was already used over 60 years ago in Ford's factories in River Rouge, Michigan for organising production. However, instead of only focusing on production, JIT can be viewed as a philosophy that encompasses the entire process from design to after sales, focusing on minimal transactions, minimal waste, minimal space etc., making it synonymous to a true Lean system.

2.2.4 Lean Operations

Lean is a philosophy and a methodology for operations management focused on waste elimination, operation streamlining and continuous improvement (Stevenson 2012). A lean operation uses considerably fewer resources than traditional operations and tend to have better productivity, lower costs and higher quality. Lean systems are also referred to as JIT systems due to their high level of coordination between activities and resource deliveries. Lean was developed by the Japanese automobile manufacturer Toyota in mid-1900s and was further successfully adopted by other companies in mid-1980s. The following methods are widely understood as components of Lean, further iterated in Figure 8:

- Muda. Waste and inefficiency in Japanese.
- Pull system. Replacing material and parts based on demand.
- Kanban: A manual system signalling the need for materials.
- Heijunka. Workload levelling.
- Kaizen. Continuous improvement of the system.
- Jidoka. Quality at the source (autonomation).

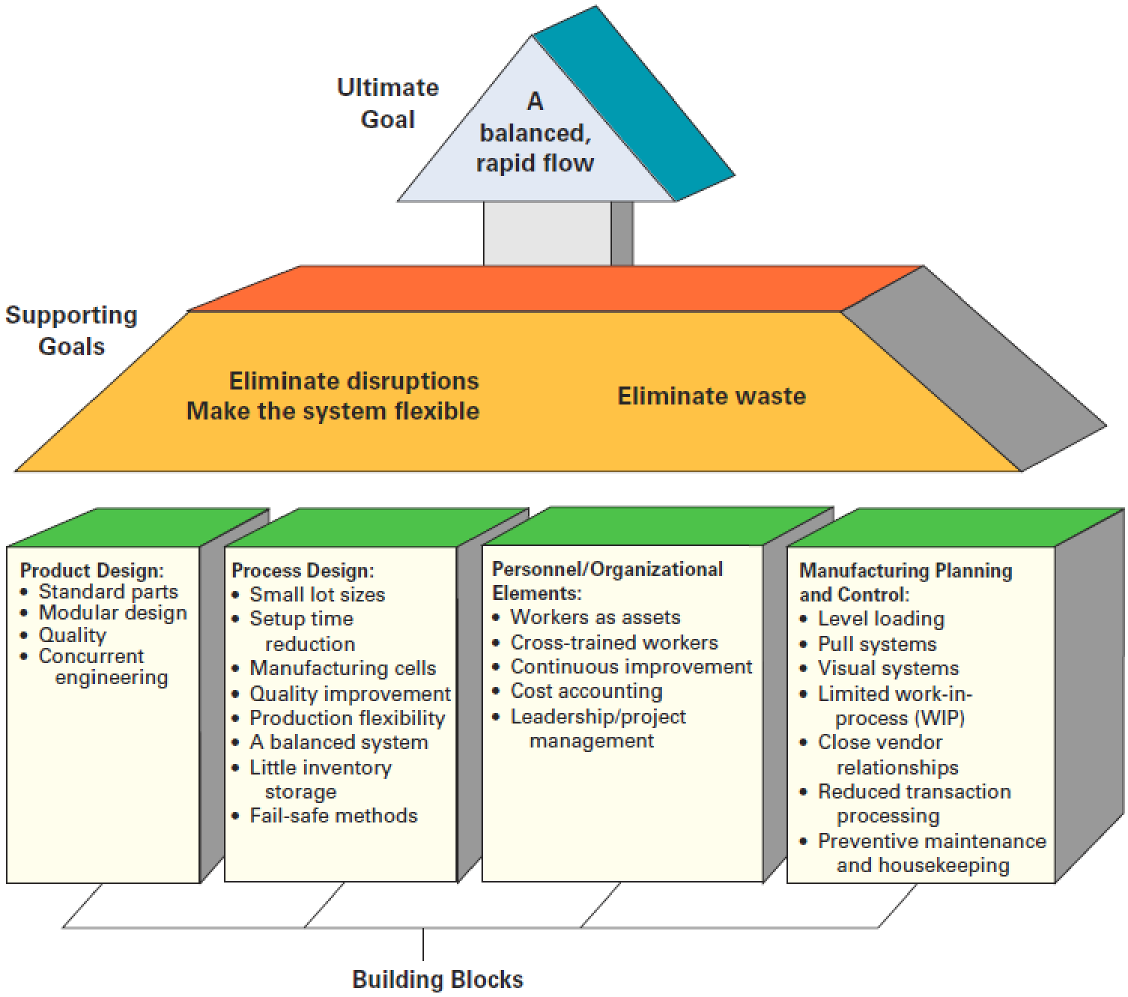


Figure 8. *Overview of Lean system goals and building blocks (Stevenson 2012).*

Muda (orig. 無駄) is a Japanese term literally meaning futility, uselessness, idleness and wastefulness (Stevenson 2012; Suárez-Barraza et al. 2016). It can be divided into seven categories according to Stevenson (Stevenson 2012):

- Inventory, idle resources takes costly floor space
- Overproduction, excessive resource use
- Waiting time, requires space and adds no value
- Unnecessary transportation, requires personnel and increases Work-In-Progress (WIP) inventory
- Processing waste, unnecessary production steps
- Inefficient working methods, low productivity, increased inventory and scrap
- Product defects, creates reworking costs, possible lost sales.

The pull system is critical to prevent overproduce and large inventories. A traditional push system is based on an advance schedule and purchase orders are made by projected demand. A pull system is based on current demand, effectively preventing large inventories. It can use simple signals, also known as Kanban, between operations such as empty bins, to communicate that a certain material needs to be ordered and restocked. (Liker 2004).

Kanban (orig. 看板）is also a Japanese term and it translates to signboard. Kanban was developed by Taiichi Ohno to control production and to implement JIT manufacturing at Toyota in Japan. Kanban's core is about using visual signals to communicate production status and work load so that supervisors can see the schedule status with a glance. It minimized the WIP and reduced the inventory costs. (Gross & McInnis 2003).

Heijunka (orig. 平準化) means the levelling or harmonization of production by both volume and product mix. For example, if a company is fully focused in making one product and the customer unpredictably decides to order another type, production is in trouble and workers would have to work overtime, which raises personnel costs. Heijunka addresses this by balancing the product mix to include other products as well and lowering the amount of the primary product. This increases the flexibility of production and decreases inventory in case that all of the primary products do not get sold. (Liker 2004).

Kaizen (orig. 改善) is a Japanese term meaning improvement and it represents constant improvement in domestic and working life. When implemented at a workplace, it means involving everyone to improve in all levels from top managers to workers. Kaizen is known to improve production values and employee morale and safety (Shettar et al. 2015). It can be applied anywhere, and it doesn't require spending a lot of money. It is driven with total transparency of procedures, making problems and wastes visible to everyone. It focuses on where value is created and learning while doing. (Stevenson 2012).

Jidoka (orig. じどか) is one of the pillars of Lean, developed by Sakichi Toyoda tracing back to his work with automatic looms. Toyoda's looms had a device that detected when a thread broke and immediately stopped production for the operator to step in and investigate. This effectively prevents poor quality and waste of materials. This philosophy emphasizes that quality must be built from the start and that there must be a method to detect defects and halt production. (Liker 2004).

2.3 Technological trends

The growth and heterogeneity of data burdens most companies (Haas 2006) and places a higher demand for human specialist work in integrating information infrastructure (Jarke et al. 2014). There are numerous researches and theses done about integrating data collection into a machine, as mentioned in the introduction. In 2010, a survey (Brodie 2010) showed that about 40% of database-related work was spent on integration work with 68% requiring top management attention in 2007. This has no doubt grown since then, due to increasing amounts of data.

Gilchrist states that currently the required amount of standardization of protocols has not been achieved to create a full Industrial Internet of Things (IIoT) value chain. While system integration within an organization is possible, interconnecting with external partners will be potentially costly, inefficient and insecure (Gilchrist 2016). This subchapter introduces technological trends that drive the importance of information integration.

2.3.1 Industry 4.0 and Smart Manufacturing

Industry 4.0 and Smart Manufacturing are both industrial initiatives that aim to increase the efficiency of industrial businesses. Former being used mostly in Europe and latter in America. Similar national initiatives have been launched in China's Made in China 2025 (Notice of the State Council on Printing and Distributing "Made in China 2025"; Kenderdine 2017), France's New Industrial France (2015), Japan's New Robot Strategy (2015) and Korea's Manufacturing Innovation 3.0 (Korea Releases Middle to Long-Term Policy Tasks).

Industry 4.0 is a phase of industrial development, backed by the government of Germany. The name refers to the fourth industrial revolution, made possible by networking devices and services, which together form cyber-physical systems (CPS). These systems perform automatic interexchange of necessary information to optimize production processes. According to the theory about the fourth industrial revolution, the previous revolutions have been mechanical machines, electricity and electronics. (Kagermann et al. 2013).

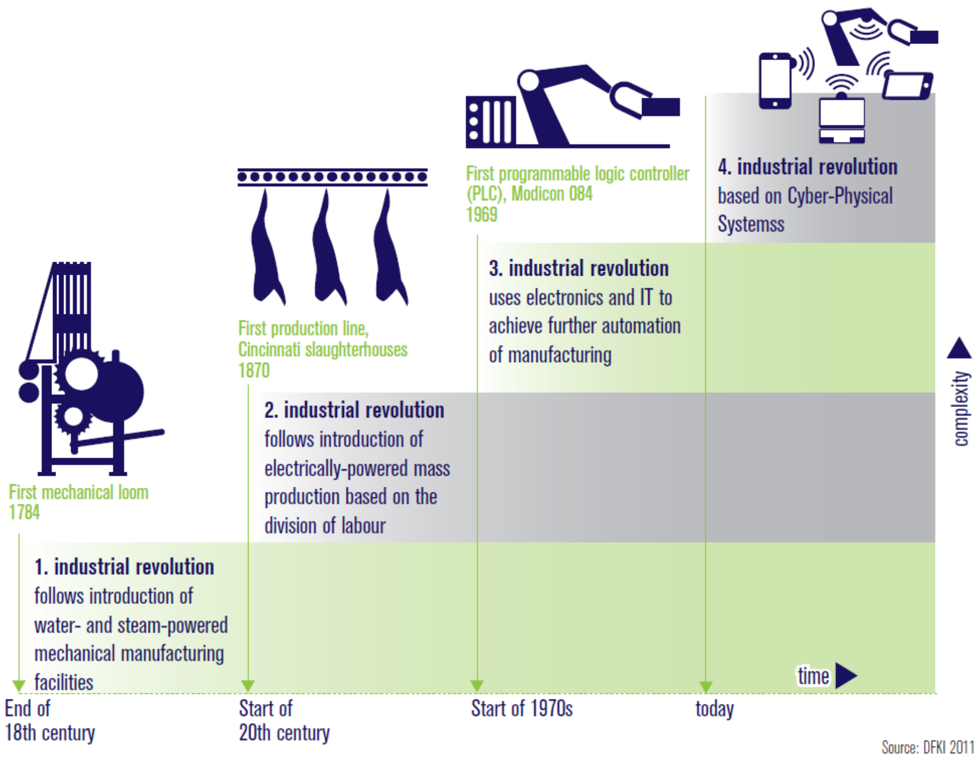


Figure 9. *Industrial revolutions (Kagermann et al. 2013).*

The eight priority areas of Industry 4.0 are (Kagermann et al. 2013):

- Standardization and open standards for a reference architecture
- Control of complex systems
- Wide-reaching broadband industrial infrastructure
- Safety and information security
- Organization and design of work
- Training and continuous professional development
- Regulatory framework
- Resource efficiency.

Smart manufacturing addresses similar issues in the following wording (Priorities, Infrastructure, and Collaboration for Implementation of Smart Manufacturing, 2012):

- Develop a standards-based reference architecture based on industry-driven collaboration with IT suppliers
- Establish an industry-shared SM Platform that includes an open architecture software development platform
- Implement R&D projects for joint investment and execution of SM systems
- Facilitate efforts to secure funding through public-private and private-private partnerships to address priorities
- Operate industry test beds for Smart Manufacturing System concepts and make them available to companies of all sizes
- Lower cost barriers for applying advanced data analysis, modelling and simulation in core manufacturing processes
- Build pre-competitive, open architecture infrastructure including network and information technology, interoperability and shared business data
- Integrate requirements of small, medium and large enterprises
- Create and provide broad access to next-generation sensors, including low-cost sensing and sensor fusion technologies
- Ensure multi-level cyber security and protection at a scalable level.

Sill (2016) shortly describes them as “Smart manufacturing seeks to extend traditional manufacturing methods to include autonomous, adaptive processes and to integrate these processes into modern information technologies.”. Sanders et al (2016) found that Industry 4.0 not only helps companies in lean manufacturing, but also makes the factories smart.

2.3.2 Industrial Internet

Industrial Internet is a term for the application of internet technologies in industry (Sauter & Wollschlaeger 2011). In this context, Sauter & Wollschlager (2011) defines internet technologies in three groups: transport and communication related technologies, technologies for information description and presentation and technologies for server- and client-side functions. They propose that Industrial Internet "seems to solve most major problems in the industrial automation domain" saying that "internet technologies seem to perfectly fit the requirements for vertical integration in automation". This means technologies such as easily installable, reusable browser and server software. Sauter & Wollschlager (2011) also mention its' downsides like heterogeneity technologies with very short life cycles. The World Economic Forum addressed the importance of Industrial Internet in a 2015 report (Industrial Internet of Things: Unleashing the Potential of Connected Products and Services, 2015) that "In the next 10 years, the Internet of Things revolution will dramatically alter manufacturing, energy, agriculture, transportation and other industrial sectors of the economy which, together, account for nearly two-thirds of the global gross domestic product (GDP).".

An alternative interpretation of Industrial Internet was made in 2012 by General Electric (GE) that it comprises of three elements (Evans & Annunziata 2012): interconnected intelligent machines, advanced analytics and connecting people at work. Industrial Internet starts with the integration of sensors, software, middleware, backend etc. to gain a better understanding of a company's operations (Evans & Annunziata 2012). According to Gilchrist (2016), industrial internet is the industrial-side of the Internet of Things (IoT) (Figure 10).

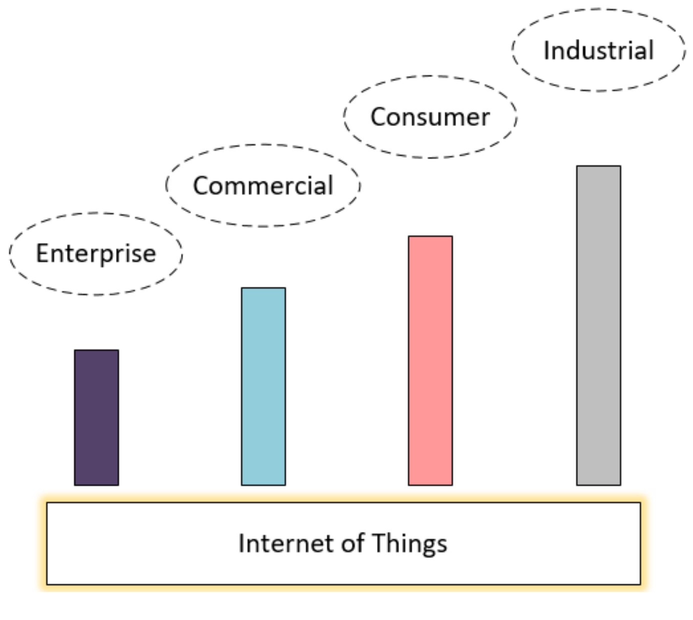


Figure 10. *Horizontal and vertical aspects of Internet of Things (Gilchrist 2016).*

When talking about Industrial Internet benefits, a term called the “Power of 1%” can be used (Evans & Annunziata 2012; Gilchrist 2016) to emphasize the amount of savings many industries need to make to realize benefits worth billions of dollars (Figure 11).

What if... Potential Performance Gains in Key Sectors

Industry	Segment	Type of Savings	Estimated Value Over 15 Years (Billion nominal US dollars)
Aviation	Commercial	1% Fuel Savings	$30B
Power	Gas-fired Generation	1% Fuel Savings	$66B
Healthcare	System-wide	1% Reduction in System Inefficiency	$63B
Rail	Freight	1% Reduction in System Inefficiency	$27B
Oil & Gas	Exploration & Development	1% Reduction in Capital Expenditures	$90B

Note: Illustrative examples based on potential one percent savings applied across specific global industry sectors.
Source: GE estimates

Figure 11. *Industrial Internet: The Power of 1 Percent (Evans & Annunziata 2012).*

Instead of only having precise sensors, components can be self-aware, self-predicting and self-comparing. This is possible because of the improvements in cost and size of sensor technologies in recent years. A key enabler of the Industrial Internet is Big Data and advanced analytics that can make sense of the vast amount of data being created (Lin et al. 2017). Combined with self-diagnosing components, they can provide, for example, preventive maintenance scheduling, mobile healthcare devices or health and safety inspection drones for the oil and mining industries. (Gilchrist 2016).

Why Industrial Internet is such an important topic now? Industrial processes are constantly becoming more complex and more difficult to optimize, outpacing the operators’ abilities. At the same time, increased capacity and reliability of data transfer as well as the overall technological maturity of related instruments combined with greatly decreased cost make it possible to have interconnected devices work together safely in an industrial environment. (Gilchrist 2016).

There are several key elements according to Evans and Annunziata (2012) that need to be invested in to allow Industrial Internet to thrive. Innovations in equipment, analytics, standardization and business processes. Digital infrastructure like data centres and cyber security management. Finally, these will not exist without proper talent management. Aside from technical know-how, cross-discipline skills will be ever more important.

According to World Economic Forum's report (Industrial Internet of Things: Unleashing the Potential of Connected Products and Services, 2015) the evolution of Industrial Internet is likely to go through four phases (Figure 12). The first phase emphasizes productivity and cost reductions, which are essentially part of Lean operations. The second phase is about finding new products and services based on the use of collected data and services that work pay-per-use. This is both agile manufacturing (2.2.1) and utilizing business intelligence (2.2.2). In the third phase, the focus is put on the outcome. This can mean products are manufactured according to users' customized preferences, essentially a pull system (2.2.4). In the final fourth phase that implements many of agile manufacturing strategies (2.2.1), products along with their manufacturing path, from raw material to finished product, are automatically adjusted according to markets' demands.

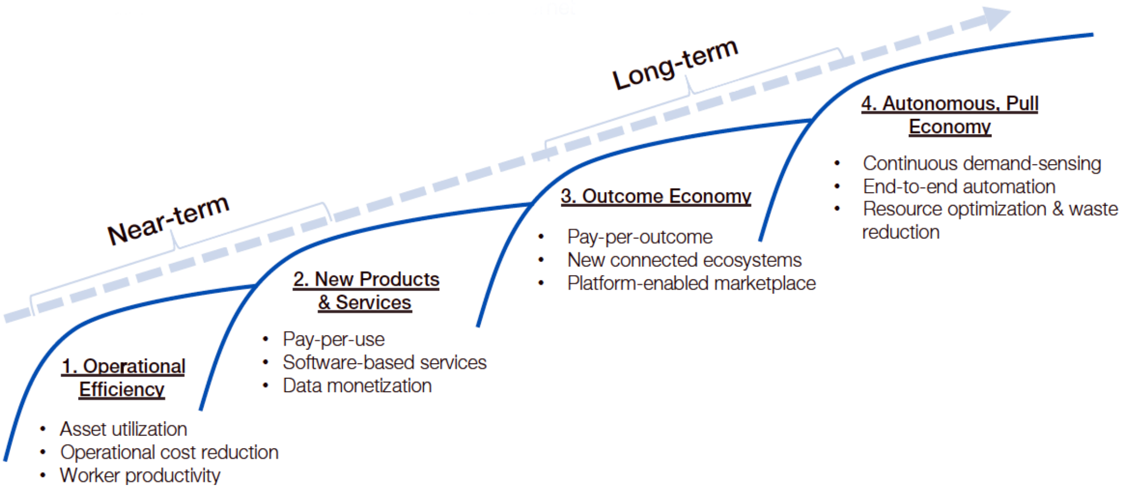


Figure 12. *The Adoption and impact path of the Industrial Internet (Industrial Internet of Things: Unleashing the Potential of Connected Products and Services, 2015).*

2.4 Information requirements regarding machine tools

What information do operations managers require from their machine tools? According to Pearce (1982), three economic goals guide every viable business organization: survival, growth and profitability. Above all else, profitability is the main goal (Pearce 1982) of any business organization as they need to cover their expenses, loan repayments and distribution of profits (Helin 2010). These goals have not changed over the years (Helin 2010). These create requirements for operations to strive for better efficiency and lower costs as explained in section 2.1. This can mean e.g. better resource economy, product quality and risk management. With machine tools, better resource economy can mean less scrap material and longer tool durability. Better product quality can mean less excess process heat and vibration. Better risk management can mean a lower chance of unexpected machine failure. This kind of data can be used to form information about productivity and reliability that are critical to current operations management trends (2.1) which aim to optimize production resources. (El-Hofy & Youssef 2008; Stevenson 2012).

P. Lade et al (2017) found five recurring categories where analytics are required in manufacturing, which support earlier findings. These are:

- Reducing test time and calibration
- Improving quality
- Reducing warranty cost
- Improving yield
- Performing predictive maintenance.

All the fore mentioned data is affordably collectible with current sensor technology and operations reporting systems. Sensors can collect thousands of process and quality measurements (P. Lade et al. 2017), including wear and tear data such as force, process temperature, vibration and acoustic emission, which can be further analysed to predict critical tool failure (CTF) (Downey et al. 2016). Furthermore, cloud information technology (IT) systems are increasingly used to manage the expanding tool inventory as single machines improve to do more and more varying tasks. These IT systems keep track of the tools' usage, collecting data per tool such as server feeding time and cutting time, to allow tool replacement prior to breakage. (Lin et al. 2015).

3. CONNECTIVITY AND DATA MODELLING TECHNOLOGIES

Connectivity and data modelling are essentially the core of this thesis as secure and flexible discovery protocols enable devices to find each other and depending on their secure certificate, control others. This chapter details technologies that are critical to this thesis, such as overviews of Universal Plug and Play, OPC UA and MTConnect. As hinted in the introduction, Universal Plug and Play is used as a successful example of simplifying device integration, to which OPC UA and MTConnect can be methodologically compared against. The operating environment of all of them can be explained with the Open Systems Interconnection (OSI) model's layers, originally standardized in 1984 (Table 2).

Table 2. *OSI basic model's layers (ISO/IEC 7498-1 1994).*

Layer	Protocol data unit	Function
7. Application	Data	High-level applications.
6. Presentation		Data translation, e.g. encoding, compression, encryption or decryption.
5. Session		Connection session management.
4. Transport	Segment (e.g. TCP) / Datagram (e.g. UDP)	Reliable data transmission.
3. Network	Packet	Addressing, routing and traffic control.
2. Data link	Frame	Transmission between two nodes.
1. Physical	Bit	Bit transfer through cables.

3.1 Universal Plug and Play

As briefly mentioned in the introduction, Universal Plug and Play (UPnP) is a high-level protocol led by the Open Connectivity Foundation that aims to expand the simplicity and autoconfiguration of device Plug and Play to entire networks of intelligent appliances, wireless devices and PCs of all sizes (ISO/IEC 29341 2000; Miller et al. 2001). UPnP was competed by Home API[1], which eventually merged its' efforts with the UPnP Forum (#TBT: A Brief History of Microsoft's Failed Attempts at Home Automation). For example, a universal serial bus (USB) printer can be connected to a PC through UPnP-compliant wired local area network (LAN) by simply connecting a cable. An example topology is depicted in Figure 13.

[1] A standardization working group founded by Compaq, Honeywell, Intel, Microsoft, Mitsubishi and Philips.

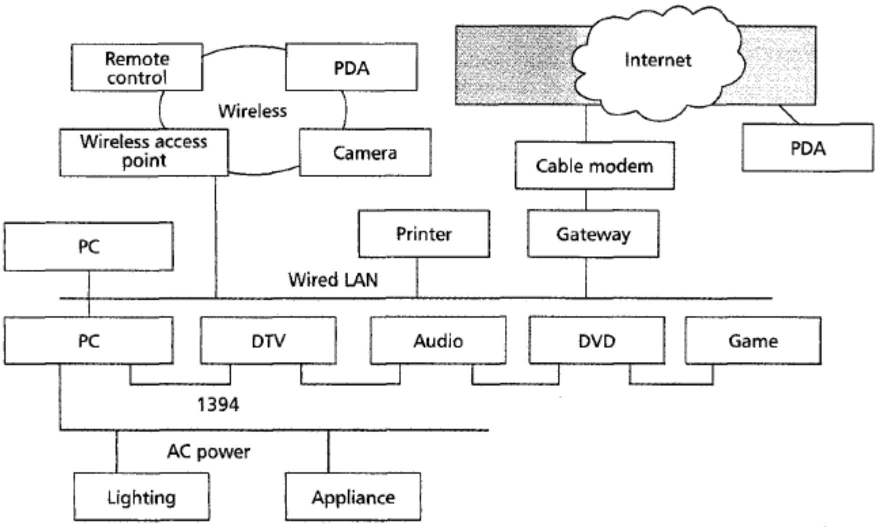


Figure 13. *An example UPnP topology (Miller et al. 2001).*

Using common internet components such as standard internet protocol (IP), hypertext transfer protocol (HTTP) and XML, TCP and UDP, it allows a device to join a network, obtain an IP address and converse with other devices about its' and their capabilities without any need for device drivers (ISO/IEC 29341 2000; Miller et al. 2001).

UPnP is independent of the underlying physical media due to its' pure IP nature and it can be implemented using any programming language and any operating system. UPnP network nodes are classified into four categories (figure 3):

- Control point, intelligent active devices like PCs
- Controlled device, less intelligent passive devices able to perform an action, e.g. DVD player
- UPnP Bridge, intelligent multiprotocol, multitechnology UPnP device
- Legacy devices, not UPnP compliant and/or unsuitable for TCP and HTTP.

Mature protocols like IP, UDP and TCP are used to ensure interoperability between devices. Figure 14 depicts the protocol stack used by UPnP. (Miller et al. 2001).

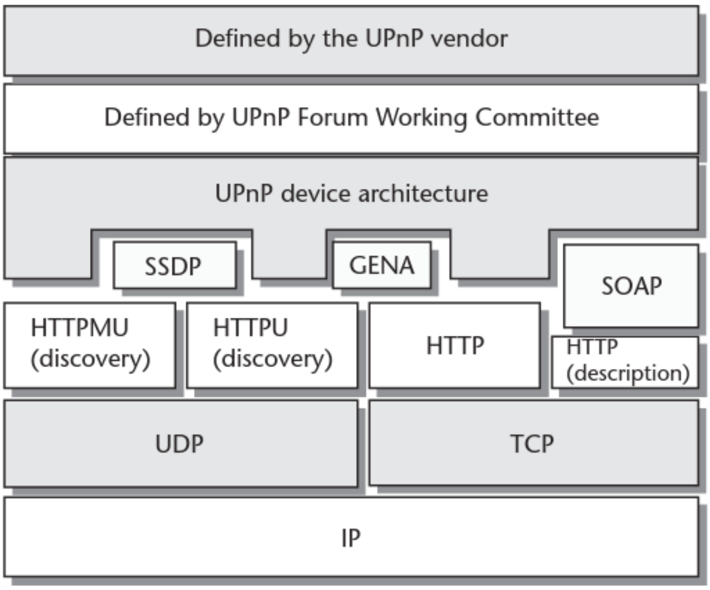


Figure 14. *UPnP protocol stack (Miller et al. 2001).*

According to the UPnP standard (ISO/IEC 29341 2000), the foundation of UPnP is IP addressing. When an UPnP device is connected to a network, AutoIP[2] is used to generate an IP address in the range of 169.254.xxx.yyy and the Address Resolution Protocol (ARP) is used to check if another device is using the same address. If the address is in use, the device chooses another address and tries again until it finds an empty address. After an IP address is chosen, the device will try and seek an active DHCP server. After the DHCP connection, the device will advertise itself and its' services to the network controller using the UPnP discovery protocol, which is based on Simple Service Discovery Protocol (SSDP). Control points can also use it to search for devices. These discovery messages are done using multicasts and the network traffic is minimized with advertisement life-time, time to live (TTL).

According to Häber (2007) connections are traditionally made in singular channels or so called unicast transmissions, where node A sends a packet to node B. In broadcast, one node sends a packet to all other nodes in the network. A smaller version of broadcast is the multicast where the broadcast is limited to a multicast-group; this is generally used in PC-to-Router connections. The fourth option is Anycast, which is essentially multicast, but it only transmits to the closest member of the group to balance the network load (Figure 15). In unicast transmission, each node in the network must have a unique address and these can be provided in three separate ways: manually, dynamically or randomly.

[2] A module that enables server-less, dynamic IPv4 address assignment to a device.

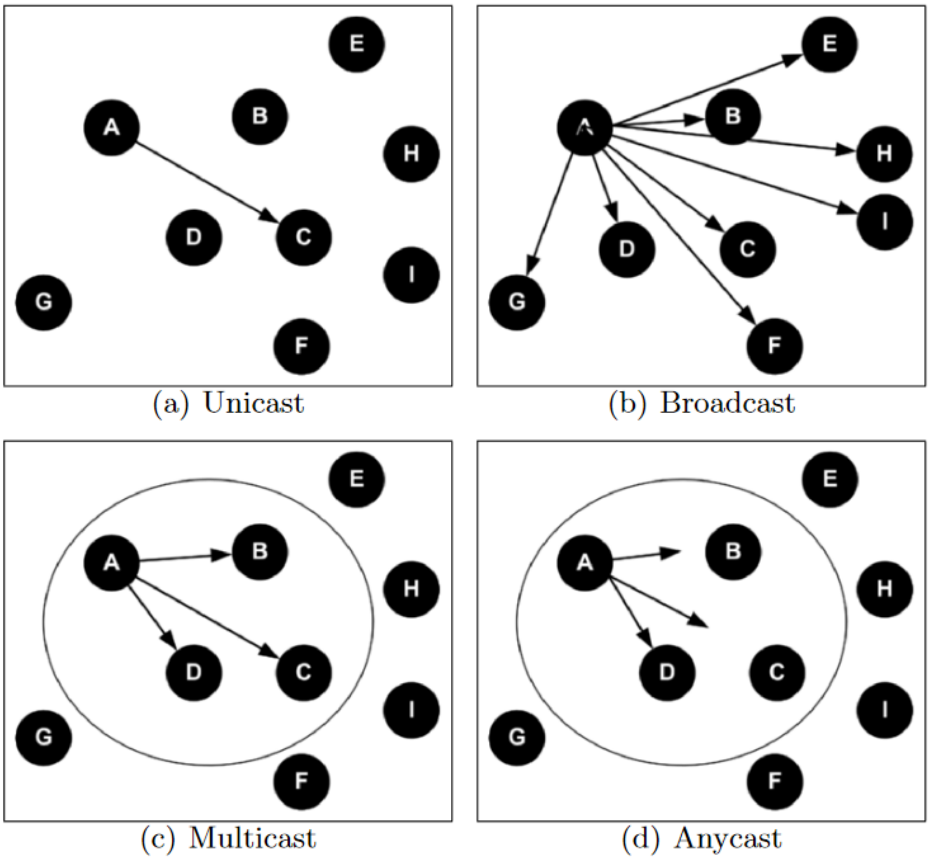


Figure 15. *Different routing schemes (Häber 2007).*

Responding to the UPnP discovery message, a control point will request a device description and one or more service descriptions from the URL contained by the discovery message. The device description contains vendor-specific information and the service description contains descriptions of the methods the service responds to. After discovery, transmitted messaged can be of two types: control or event messages. (Miller et al. 2001; Grimmett & O'Neill 2012).

3.2 OPC UA

OPC UA is an effort of the OPC Foundation in defining how information is described and transferred. Collaborative organizations like the MTConnect Institute is one of the many that define what data needs to be described and transferred (Mahnke et al. 2009). It is based off the earlier Object Linking and Embedding for Process Control (OPC), which is now more commonly referred to as OPC Classic to differentiate it from OPC UA (What is OPC?).

According to Mahnke et al (2009), OPC Classic is based on Microsoft's Component Object Model (COM) and Distributed Component Object Model (DCOM) technology and has three major specifications for data integration between process level and the management level. These specifications are Data Access (DA), Alarm and Events (A&E) and

Historical Data Access (HDA). These are mainly used in Human-Machine Interfaces (HMI) and Supervisory Control and Data Acquisition (SCADA) systems. OPC Foundation (What is OPC?; Unified Architecture) states that OPC Classic does not meet today's industry requirements because of the Windows platform dependent COM and the networking issues of DCOM, which effectively prevent internet communication. OPC Classic also has no information security or scalability. OPC XML Data Access (DA) tried to make OPC platform-independent by replacing COM and DCOM with HTTP, Simple Object Access Protocol (SOAP) and Web Service technologies, but eventually failed due to inferior performance.

OPC UA aims to replace all existing COM-based specifications without losing performance (Mahnke et al. 2009). It is an expandable, platform-independent Service Oriented Architecture (SOA) framework that evolves the main OPC Classic specifications. To ensure broad compatibility, OPC UA is based on common technologies such as TCP/IP, HTTP, Ethernet and XML (IEC 62541-1 2016). In 2004, Wollschlaeger & Bangemann deemed XML as the most promising technology for being the integration core for automation and control systems due to its' flexibility.

3.2.1 Connection management

OPC UA requires multiple layers of communication channels to establish a secure data communication route (Figure 16). The application session and the secure channels are handled by the UA Stacks. These are both implemented within the UA Stacks and are used via the Stack API.

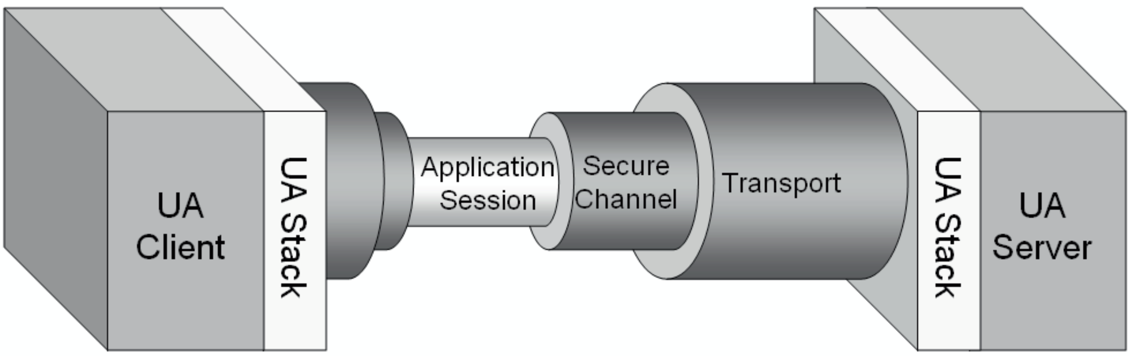


Figure 16. *Different levels of communication channels (Mahnke et al. 2009).*

OPC UA uses two transport protocols, UA TCP and SOAP/HTTP (Mahnke et al. 2009). UA TCP is a small protocol built on top of TCP, because buffer size negotiation was required at the application level and different endpoints of the OPC UA servers should have the possibility to share one IP address and port. Finally, a robust error recovery was needed at the transport layer. UA TCP defines a general message chunk structure consisting of a message header and a message body (Figure 17). Cavalieri & Chiaccio (2013) found that UA TCP is faster in establishing communication than SOAP, also resulting in shorter round-trip times. These benefits diminish however as remote certification authorities are added to secure connections.

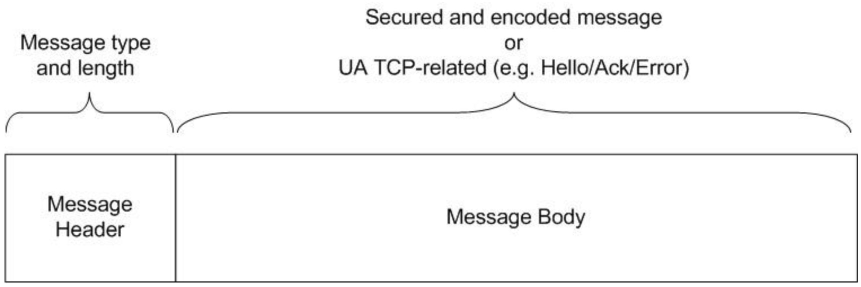


Figure 17. *Message chunk according to UA TCP (Mahnke et al. 2009).*

SOAP/HTTP stands for SOAP over HTTP and is used especially in Web Services over the Internet. It is used for calling remote procedures and relies on other standards like XML for data representation and HTTP for transport. (Butterfield & Ngondi 2016). OPC UA uses SOAP/HTTP for data transfer after the connection has been established, moving data in XML format (Mahnke et al. 2009).

For a client (Figure 18) to establish connection with a server (Figure 20), it first sends a hello message, requesting buffer sizes for sending and receiving data as well as maximum chunk and total message lengths. The server sends this information in an acknowledge message. This agreement on message sizes is critical in resisting malicious attacks like Denial of Service (DOS) attacks.

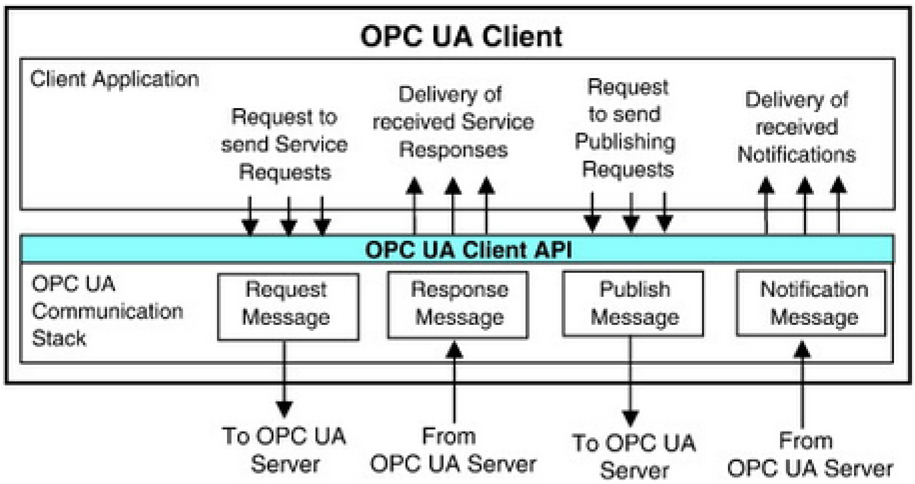


Figure 18. *OPC UA Client (Cavalieri & Chiacchio 2013).*

Any connection malfunction will automatically engage reauthentication of the connection. (Mahnke et al. 2009). This exchange sequence is visualized in Figure 19.

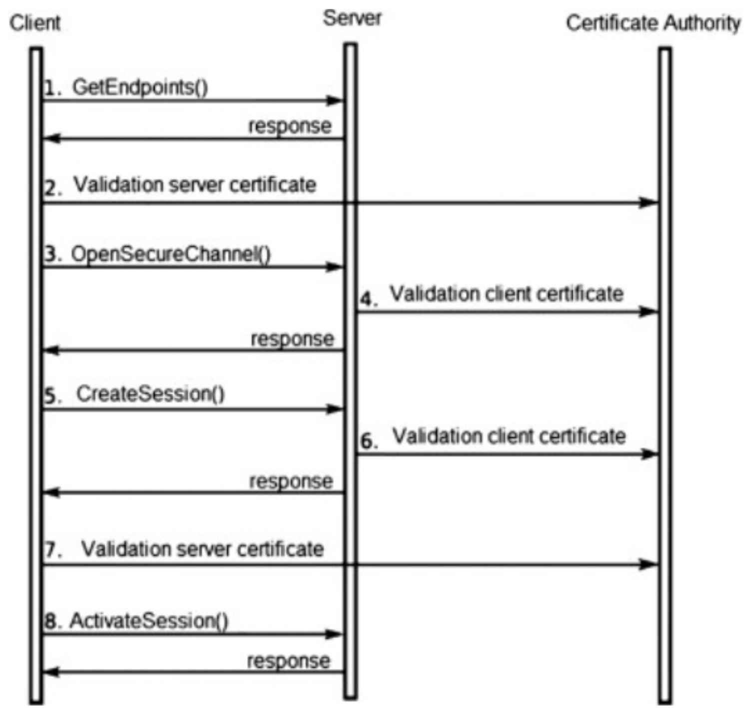


Figure 19. *Operations for the establishment of a communication context (Cavalieri & Chiacchio 2013).*

Figure 20 illustrates the server side of the communication where it handles the information handovers such as subscriptions and the OPC UA AddressSpace.

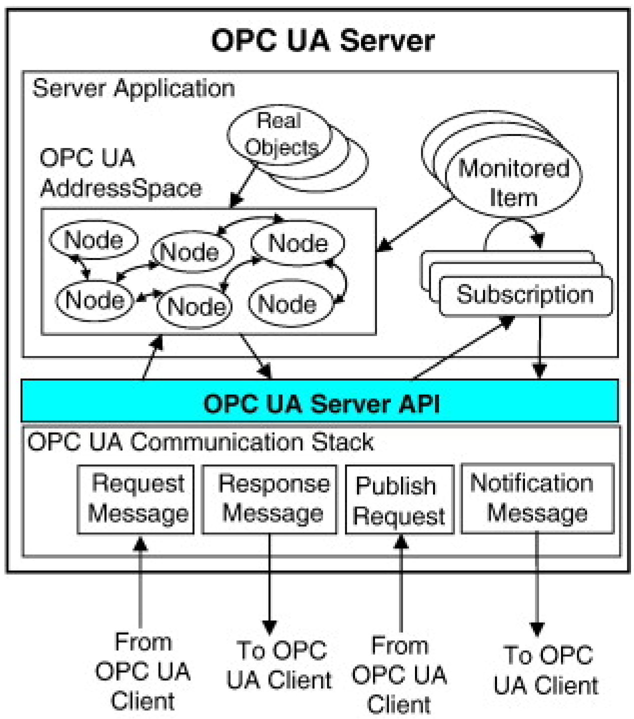


Figure 20. *OPC UA Server (Cavalieri & Chiacchio 2013).*

Client-Server data transmission provides reliable delivery using buffering, acknowledgements and retransmissions. Each data is transmitted separately via separate sessions to each client (Figure 21).

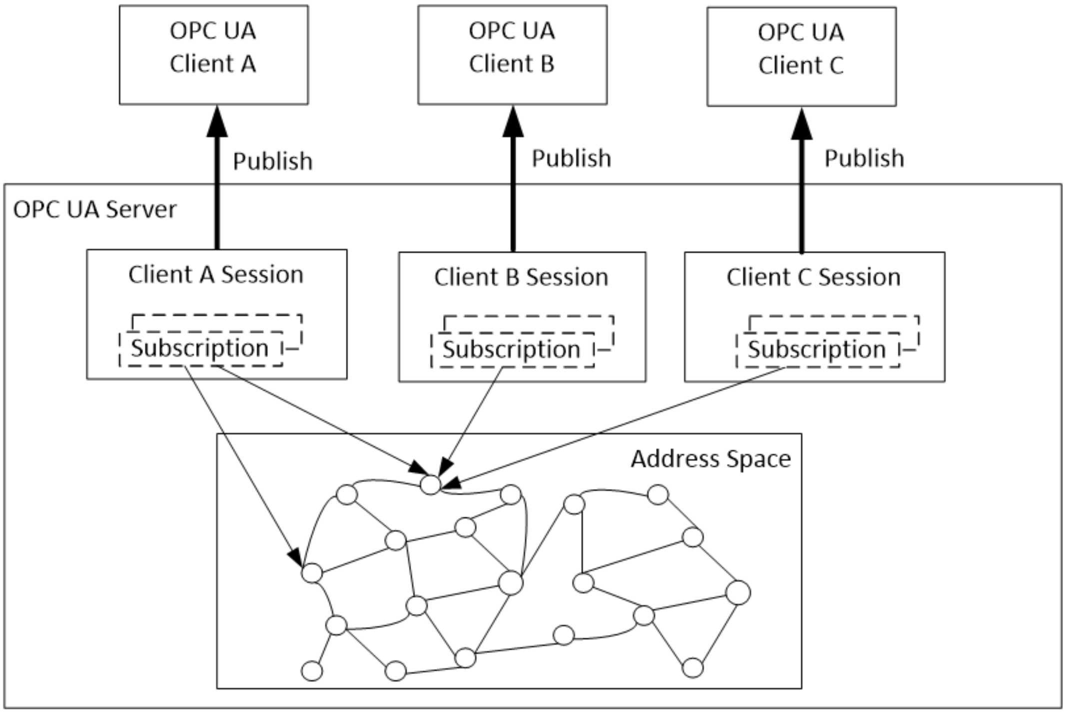


Figure 21. *OPC UA Client-Server data transmission (OPC UA Specification Part 14: PubSub, 2017).*

Aside from client-server data transmission, the OPC Foundation have introduced a cloud-like data transport specification for OPC UA called the Publish/Subscribe (OPC UA Specification Part 14: PubSub, 2017). Hoefling et al (2015) state that the Publish/Subscribe model's major advantage is "Simplified and incremental integration of new applications.", which is why this thesis considers it. The basic principle of the model is that any device with a valid security credentials can publish data under a certain topic and vice versa, any valid device can subscribe data of a certain topic (Figure 22).

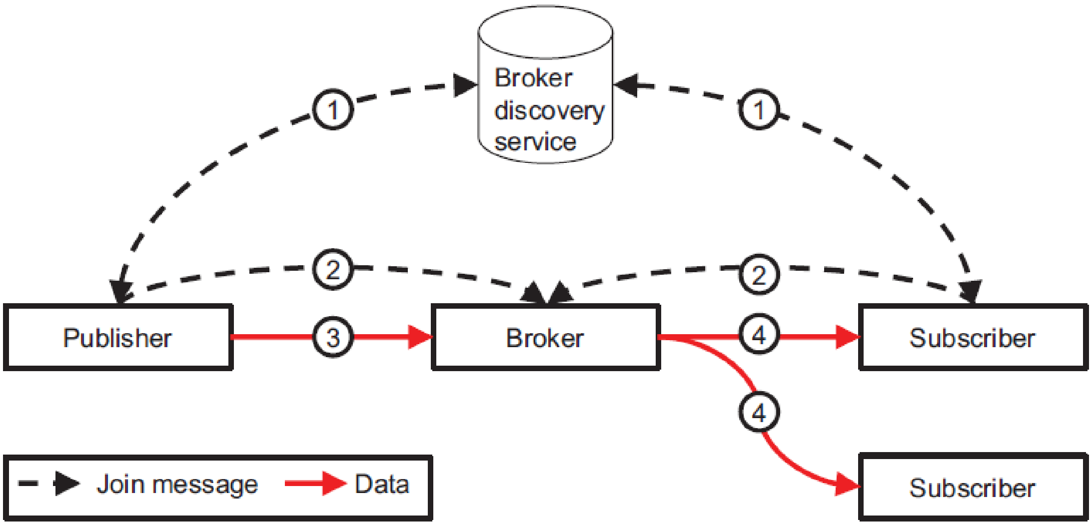


Figure 22. *Basic PubSub communication. After successful broker discovery (step 1) and client join (step 2), publishers send data to brokers (step 3) which forward data to subscribers (step 4) (Hoefling et al. 2015).*

According to the PubSub release candidate (OPC UA Specification Part 14: PubSub, 2017), in PubSub data transmission the data is published into a Message Oriented Middleware (MOM) from where subscribers can choose what data they wish to collect. The middleware provides anonymity to both publishers and subscribers. Middleware can be implemented in two different ways, a broker-less form where data is transmitted through a network that can route datagram-based messages like UDP multicast, or a broker-based form, where application layer protocols like Advanced Message Queuing Protocol (AMQP) or Message Queue Telemetry Transport (MQTT) are used to communicate with the broker. The broker can arrange messages into certain topics and it can act as a translator between differing protocols of the publisher and the subscriber. The PubSub minimizes the resource requirements for the publisher as it only communicates with the middleware.

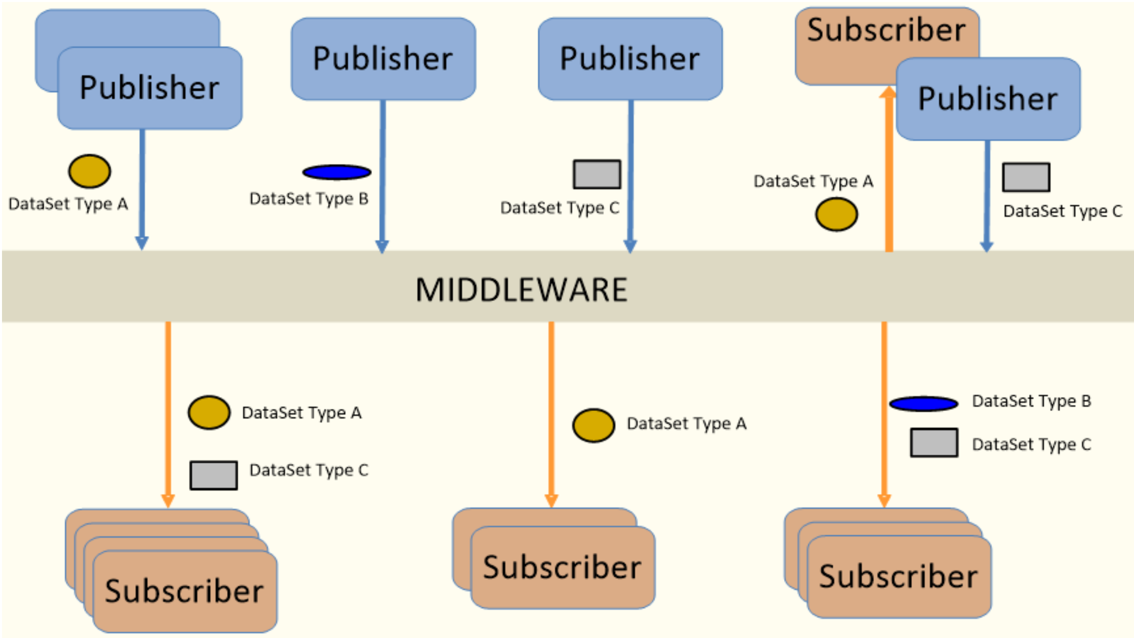


Figure 23. *OPC UA PubSub data transmission (OPC UA Specification Part 14: PubSub, 2017).*

These two data transmission models can combine as the levels of complexity increases, as both servers and clients can publish and subscribe to anonymous data. (OPC UA Specification Part 14: PubSub, 2017). Figure 24 illustrates the publisher information flow.

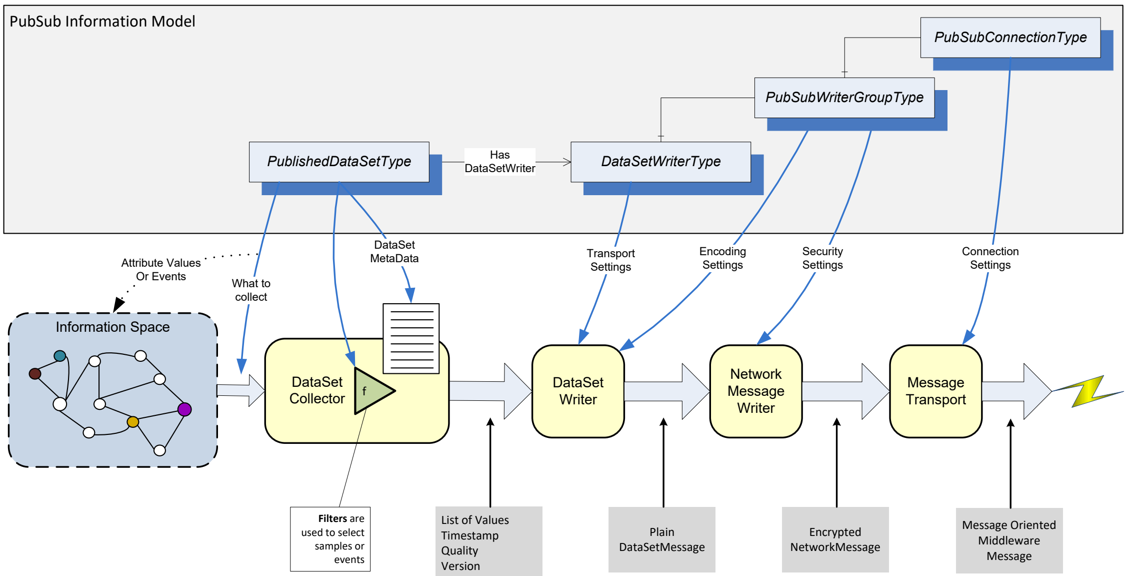


Figure 24. *OPC UA PubSub - Publisher information flow (OPC UA Specification Part 14: PubSub, 2017).*

PubSub can be used to configure publishers and subscribers with configuration methods and parameters described in the PubSub standard. The configurable parameters include various statuses and access levels.

3.2.2 Information modelling

Mahnke et al (2009) state that "The fundamentals of OPC UA are data transport and information modeling.". They further elaborate that the basis of OPC UA information modeling is based on nodes and references. Individual nodes can be grouped into NodeClasses according to their purpose for grouped configuration. Attributes are used to describe nodes according to their NodeClass. There are several attributes that are common to all nodes (Table 3).

Table 3. *Common attributes (IEC 62541-1 2016).*

Attribute	DataType	Description
NodeId	NodeId	Uniquely identifies a Node in an OPC UA server and is used to address the Node in the OPC UA Services
NodeClass	NodeClass	An enumeration identifying the NodeClass of a Node such as Object or Method
BrowseName	QualifiedName	Identifies the Node when browsing the OPC UA server. Not localized.
DisplayName	LocalizedText	Contains the Name of the Node that should be used to display the name in a user interface. Localized.
Description	LocalizedText	This optional Attribute contains a localized textual description of the Node.
WriteMask	UInt32	Is optional and specifies which Attributes of the Node are writable, i.e., can be modified by an OPC UA client.
UserWriteMask	UInt32	Is optional and specifies which Attributes of the Node can be modified by the user currently connected to the server.

Each Node is uniquely identified by their NodeId, which is used by the server to transport data and by the clients to address service calls. A node can have several different NodeId depending on how they are referenced. A reference is a connection between two nodes, the source node and the target node and can be uniquely identified by the two nodes, the semantic and the direction of the reference. (Mahnke et al. 2009).

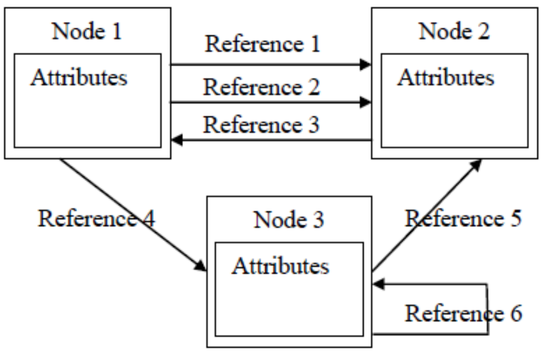


Figure 25. *Nodes and references between nodes (Mahnke et al. 2009).*

One key difference between OPC Classic and OPC UA information modeling is that OPC UA can provide type information for objects and variables whereas OPC Classic could only provide data types such as Int32 or String. Overall, this data modelling approach provides an object-oriented information model framework that defines rules on how information should be represented with the base types in the address space, depicted in Figure 26 as the "Information Access (Data Model and Services)". The services part provides essential services like connect, address space, event and subscription services. (Mahnke et al. 2009).

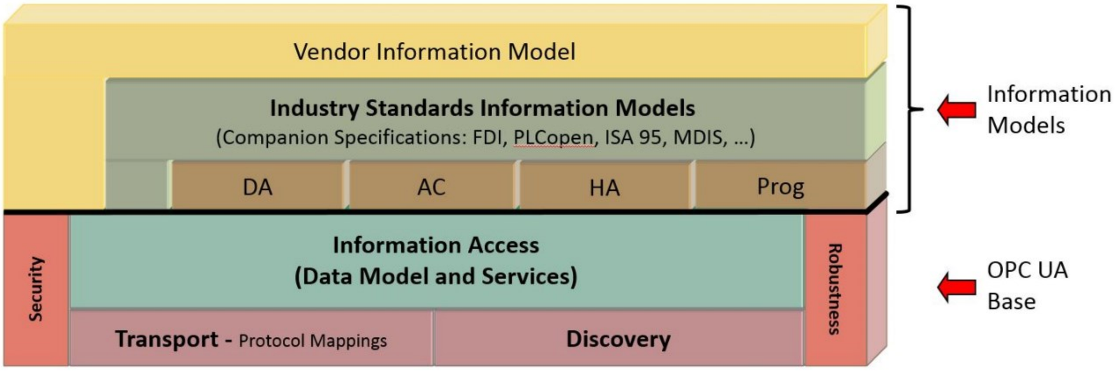


Figure 26. *OPC UA architecture (Unified Architecture).*

3.2.3 Discovery

The base OPC UA has no discovery features and the connection handshake must be initiated manually (IEC 62541 2010). The OPC UA Specification Part 12: Discovery was introduced in 2015, aiming to address this missing ability. The specification introduces a Local Discovery Server (LDS) extension, which can be used to have servers running on a same host register themselves with each other. These are specified in detail in OPC UA specification part 12 (2015). Dedicated systems that only have one server on host do not need to use the LDS extension (Figure 27).

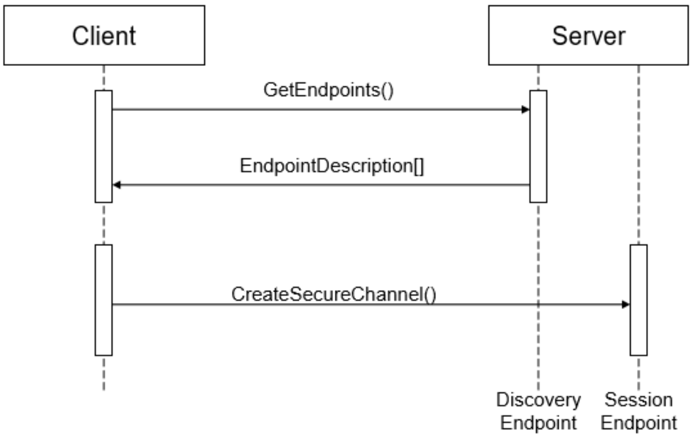


Figure 27. *OPC UA simple discovery process* (OPC UA Specification Part 12: Discovery, 2015).

As the network increases in complexity, registered servers can be found by a client by polling the LDS with a FindServers method, which the LDS answers to with an application description of the registered servers. This polling, the discovery process is illustrated in Figure 28.

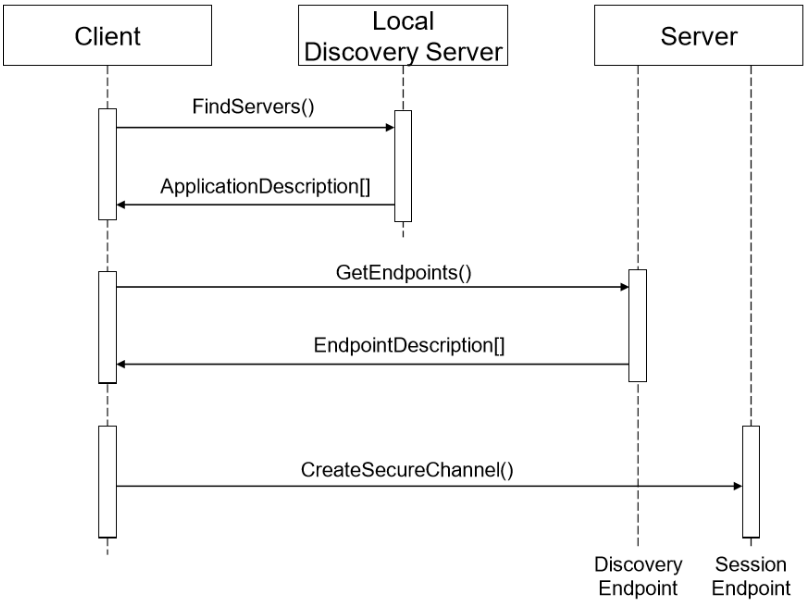


Figure 28. *OPC UA local discovery process* (OPC UA Specification Part 12: Discovery, 2015).

In larger networks, clients need to look for a LDS with the Multicast Extension (LDS-ME) and request a list of DiscoveryUrls for Servers and DiscoveryServers (Figure 29).

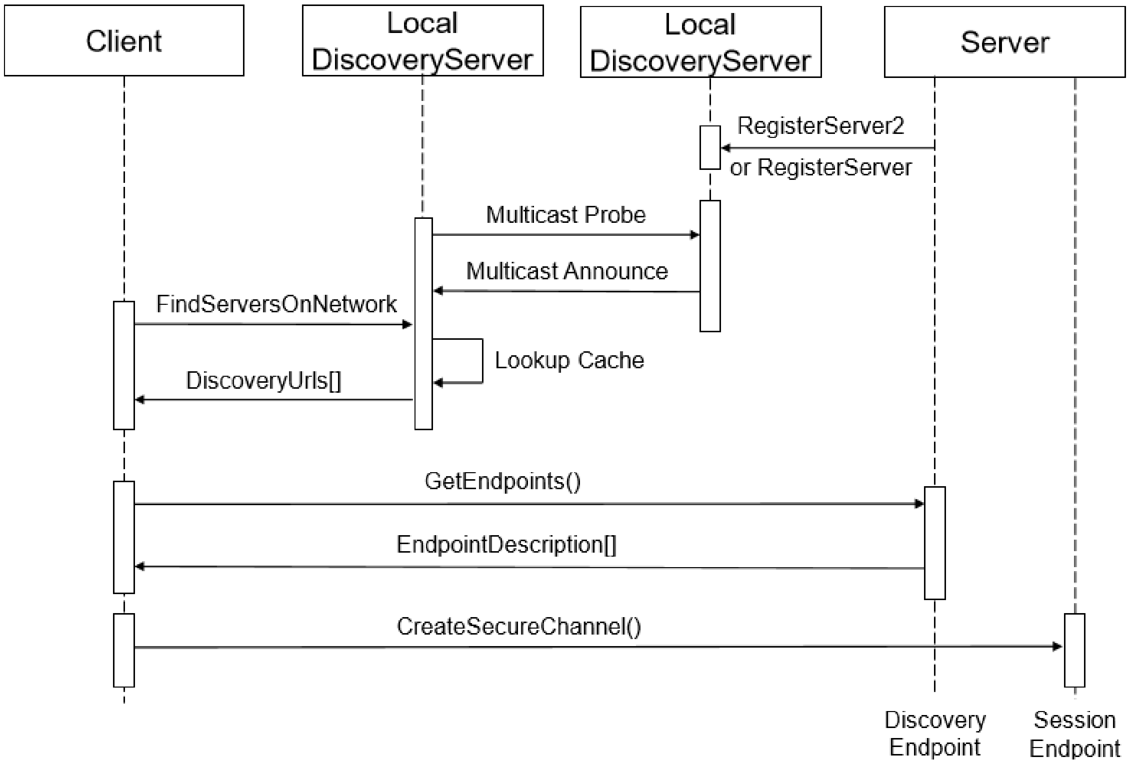


Figure 29. *OPC UA MulticastSubnet discovery process* (OPC UA Specification Part 12: Discovery, 2015).

Profanter et al (2017) found recently that the LDS-ME extension allows easy integration of devices without pre-configuration. These devices however, do need to implement a basic UA stack (Figure 14).

3.3 MTConnect

MTConnect is a data standard for numerically controlled machine tools meant to decrease the required number of adapters, driven by the Association of Manufacturing Technology, the University of California and Georgia Institute of Technology since 2006 (MTConnect® Standard, 2014).

An MTConnect system is comprised of five components:

- Device
- Adapter
- Agent
- Network
- Application

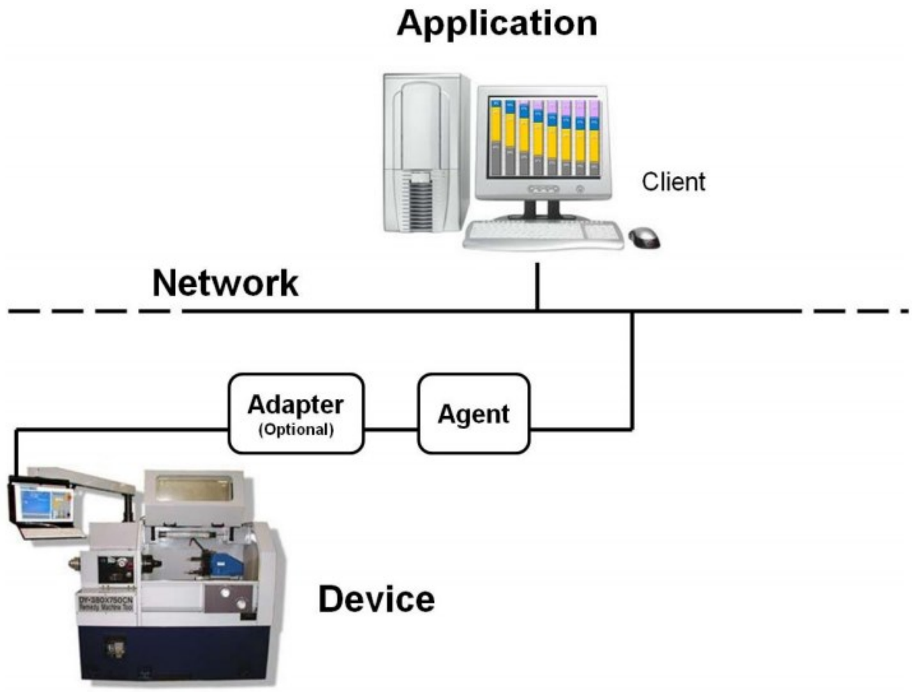


Figure 30. *MTConnect system (MTConnect® Standard, 2014).*

Sobel (2011) writes that a device is generally a machining tool, but it can be any other machine as well. The adapter is a part of a program or device that translates MTConnect into different forms. The Agent collects, organizes and processes data while handling the data transmission requests. It can only store a fixed amount of data. When new data is acquired by the Agent, the oldest data drops out (Figure 31). The Network is a data transmission tool, generally an Ethernet network. Application the data transfer customer that usually visualises the data. The MTConnect standard does not define how the physical implementation is done.

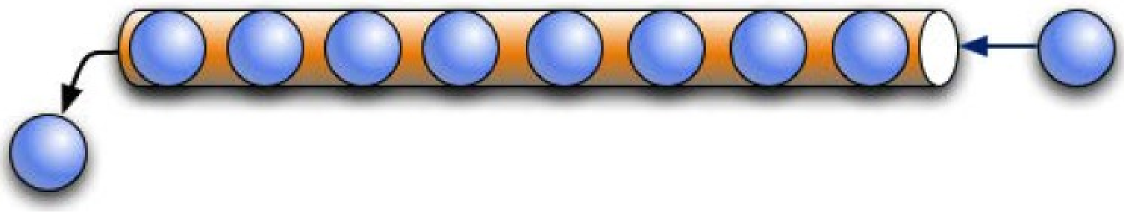


Figure 31. *MTConnect Agent Data Storage (MTConnect® Standard, 2014).*

3.3.1 Data model

A typical MTConnect device is arranged as depicted in Figure 32 and it is comprised of four upper level types (Sobel 2015):

- MTConnectDevices, contains metadata
- MTConnectStreams, contains data and timestamps
- MTConnectAssets, contains part and maintenance data as well as database keys
- MTConnectError, contains protocol errors.

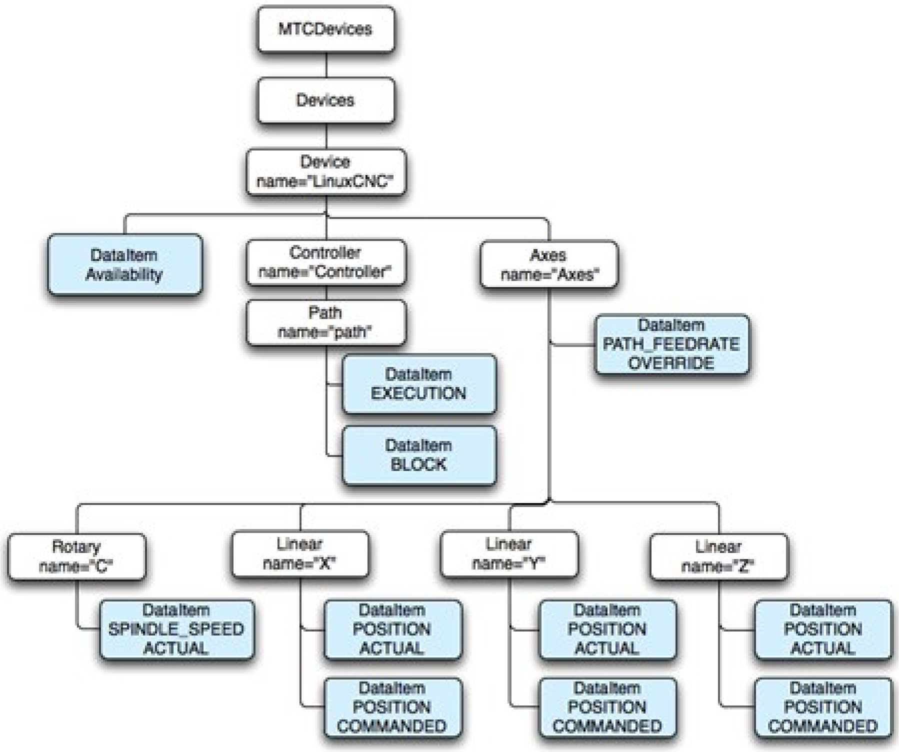


Figure 32. *Sample device organization (MTConnect User Portal).*

3.3.2 Connection management

MTConnect communication (MTConnect User Portal) works in four steps: agent initialization, probe, current and sample. The entire communication is implemented with HTTP with some Representational State Transfer (REST) interfaces to ensure efficient networking (Fielding 2000). With MTConnect this means that the agent does not know anything of the client that makes the connection requests, therefore all context must be passed in the URL and in the agent response. This simplicity makes it possible for the application to have multiple simultaneous conversations referencing different data with the MTConnect Agent. First, the MTConnect agent establishes connection with the device by first authenticating with the name server, registering its' URI. Afterwards it connects to the device using the device's API or other specialized process. Finally, the device starts sending data to the agent (Figure 33).

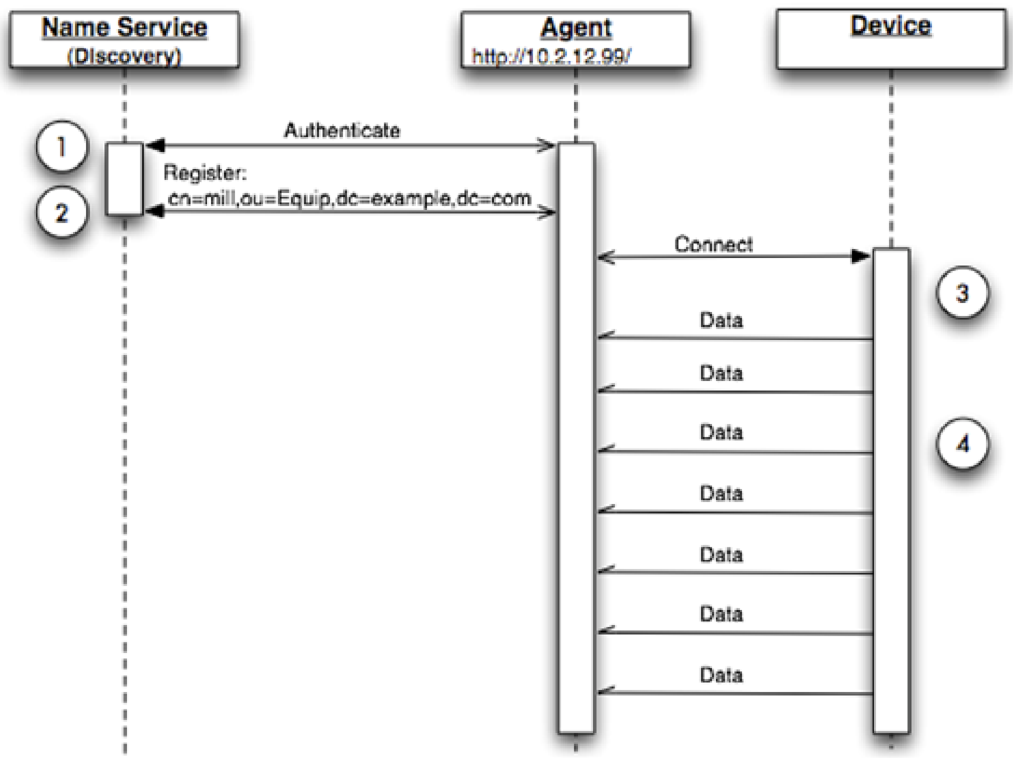


Figure 33. *MTConnect Agent initialization (MTConnect User Portal).*

When the application is connecting to the agent, it first sends out a probe request that returns metadata about what devices the agent represents. This metadata includes component composition of the devices and all available data items. (MTConnect® Standard, 2014). Figure 34 illustrates the probe request in relation to the device, agent and application.

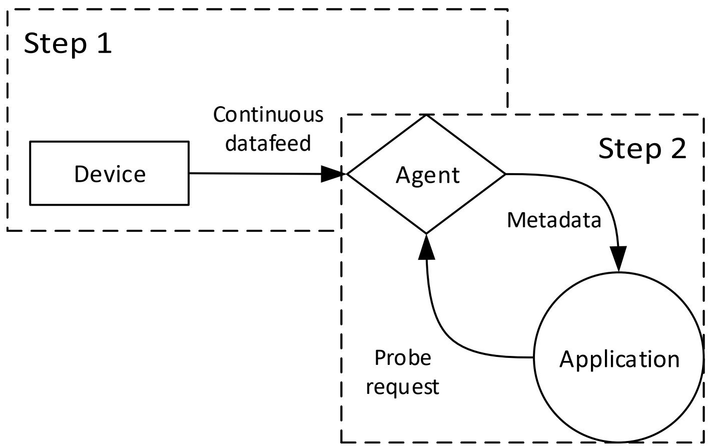


Figure 34. *MTConnect communication workflow – steps 1 and 2.*

After receiving the probe, the application can request the current state of the devices, which returns the latest snapshot of the entire device stream and all the component streams. When the current state of the device is known to the application, the Agent can be sampled at a rate determined by the application. (MTConnect® Standard, 2014). This is further demonstrated in Figure 35.

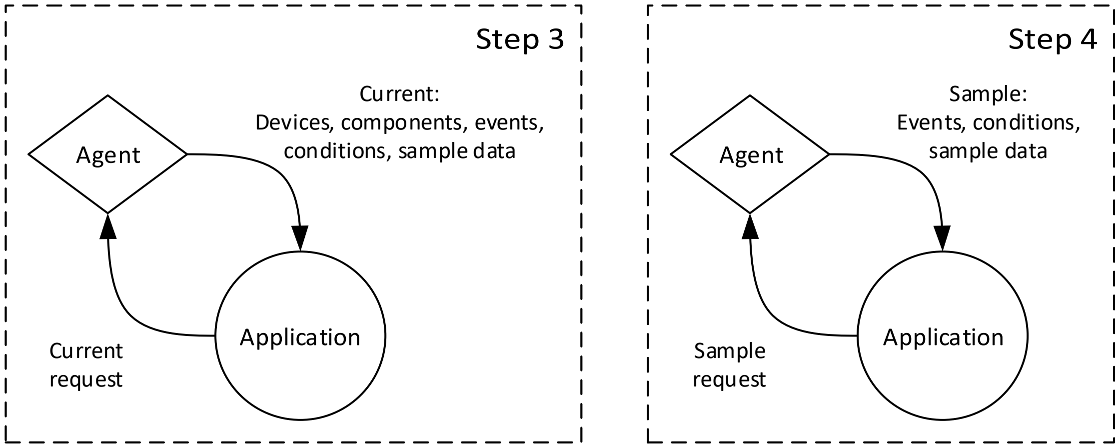


Figure 35. *MTConnect communication workflow – steps 3 and 4.*

3.4 MTConnect-OPC UA Companion Specification

Officially starting from September 2010, the OPC Foundation and the MTConnect Institute have been working together towards complete manufacturing technology interoperability. The outcome of this agreement is the MTConnect-OPC UA Companion specification (MTConnect – OPC UA Companion Specification). The goals of this companion specification are as follows:

- Incremental adoption, technical barrier between MTConnect and OPC UA must be greatly reduced through the companion specification, source code and binaries available
- Evolution, incremental improvement without endangering backwards compatibility
- Customizability, extensibility without jeopardizing compatibility with other equipment or software
- Non-proprietary, built on open standards backed by hundreds of companies, individuals, government organizations and non-profits with the goal of increased productivity.

The companion specification discusses both the backend and client side of MTConnect-OPC UA integration. Several key use cases are illustrated in the companion specification (MTConnect – OPC UA Companion Specification). Figure 36 illustrates a device manufacturer use case where OPC UA is used as a generic base, allowing easier interoperability to various equipment.

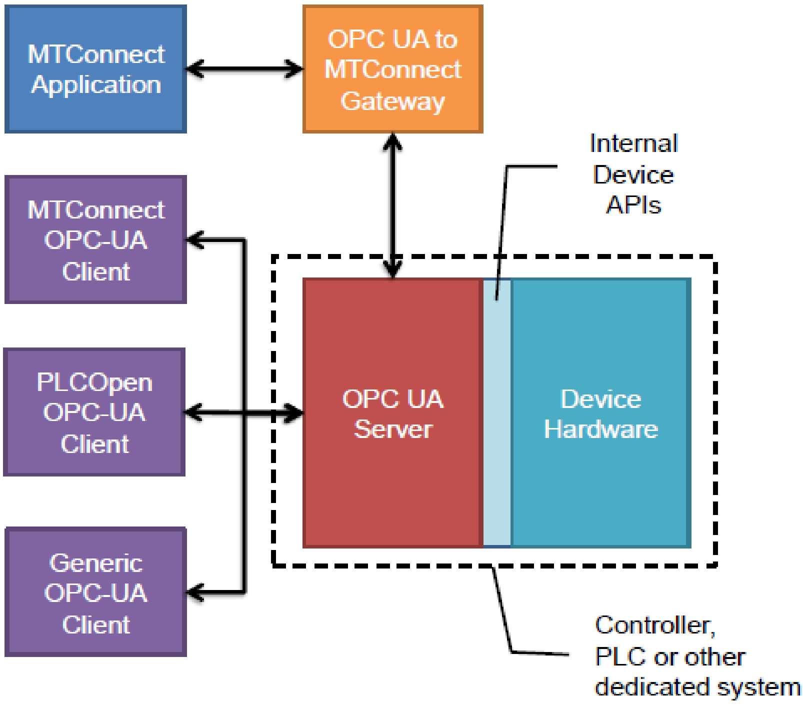


Figure 36. *Device manufacturer use case – OPC UA Server (MTConnect – OPC UA Companion Specification).*

The following figure illustrates, according to a small system integrator company, the more common manner of MTConnect implementation.

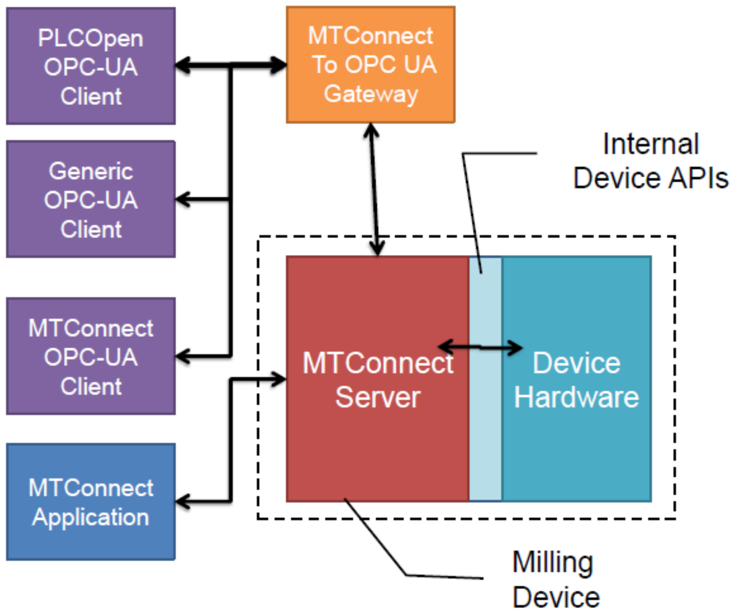


Figure 37. *Device manufacturer native MTConnect (MTConnect – OPC UA Companion Specification).*

Figure 38 illustrates a use case where an independent software vendor provides a solution where data is extracted from MTConnect and OPC UA sources and then made available to both MT Connect and OPC UA clients.

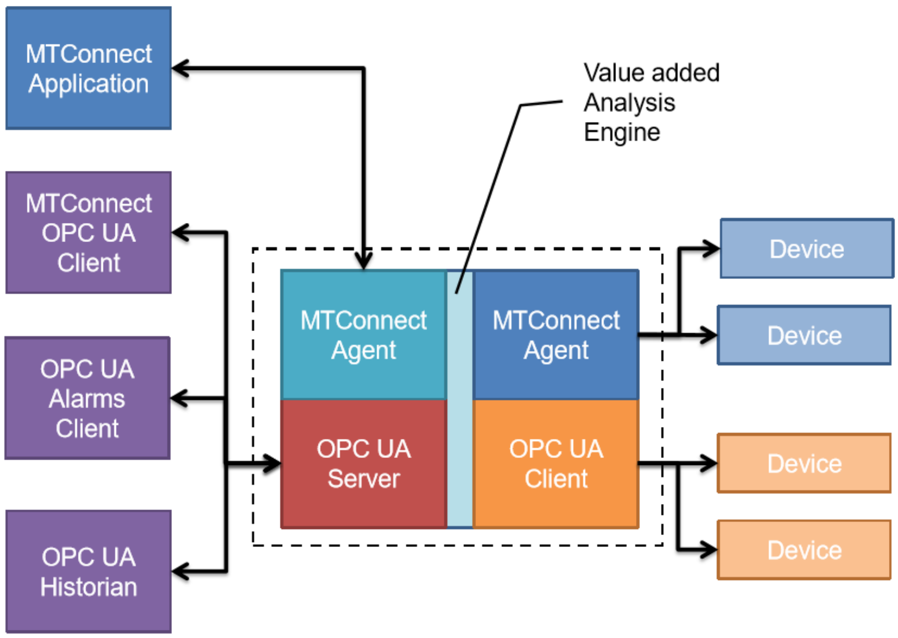


Figure 38. *Independent software vendor (ISV) use case (MTConnect – OPC UA Companion Specification).*

The companion specification also defines how MTConnect functions are mapped into the OPC UA nodes. Table 4 introduces several of the common MTConnect-OPC UA cross mappings. The MTConnect Probe request is mapped to an OPC UA browse since both return the connect devices and their structure in the address space. MTConnect Sample request is mapped to either a OPC UA read or subscription service. The read service returns the latest value of the target. In some cases, this may require historical data access if an older value is requested. The OPC UA subscription service is used with the MTConnect Streaming command, which obtains a steady stream of values from the target device. The OPC UA subscription service supports filtering based on data changes or aggregates, also returning any associated OPC events. (MTConnect – OPC UA Companion Specification).

Table 4. *Common MTConnect-OPC UA mappings.*

MTConnect	OPC UA
Probe request	Browse
Sample request	Read OR Subscription
Streaming	Subscription
Asset	Device
Error	Error where appropriate
Description	Description
Name	BrowseName
NativeName	DisplayName
UUID	NodeID

Starting from the top, the topmost data layer of an MTConnect Agent is the MTDevices element, and object of the MTDevicesType and a subtype of FolderType, which is the topmost data layer of OPC UA. Figure 39 depicts the MTDeviceType, which describes an MTConnect device under MTDevices, including its' DataItems, Components and Conditions. (MTConnect – OPC UA Companion Specification).

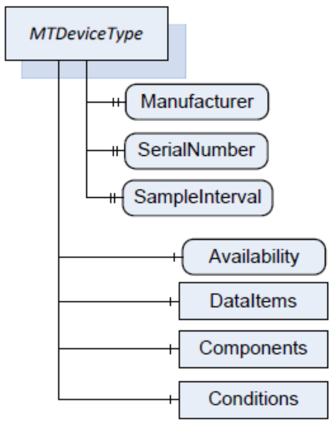


Figure 39. *MTDeviceType (MTConnect – OPC UA Companion Specification).*

Most of the companion specification document is lists of MTConnect data types and their presentation as OPC UA nodes. Figure 40 illustrates an example from the companion specification on how the presentation is done for each type of data.

Attribute	**Value3**				
BrowseName	MTDeviceType				
IsAbstract	False				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Is a Subtype of UA: BaseObjectType					
HasComponent	Variable	Availability	String	DataItemType	Mandatory
HasProperty	Variable	Manufacturer	String	PropertyType	Mandatory
HasProperty	Variable	SerialNumber	String	PropertyType	Mandatory
HasProperty	Variable	Configuration	String	PropertyType	Optional
HasProperty	Variable	SampleInterval	Duration	PropertyType	Optional
HasComponent	Object	DataItems		FolderType	Mandatory
HasComponent	Object	Components		FolderType	Mandatory
HasComponent	Object	Conditions		FolderType	Optional

Figure 40. *MTDeviceType as an OPC UA node.*

4. PROTOTYPE APPLICATION

In order to have solid business relevance and in adherence to design science principles, a part of this thesis is creating a prototype application to aid in confirming or disproving the topic of the thesis: if implementing UPnP level of success (Schiele et al. 2010; Lofgren 2015) in integration ease is possible in the industrial sector with OPC UA and MTConnect. This chapter details the project framework for the prototype, such as the requirement specification, preliminary design and planned iterative steps to complete the prototype.

The prototype seeks to prove research questions 2 and 3, OPC UA's capability of mapping MTConnect data and if their combination eases manufacturing information integration. Universal Plug and Play is used as a comparison for verification of simple interconnection.

The requirement specification is made based on the what is required by the methodology to facilitate this research. Furthermore, there is a developing business need for integrating machine tools that utilize MTConnect into manufacturing data analysis software. Due to OPC UA's information modeling and wide array of companion specifications with various industries' data models, using it has the possibility of saving time in further applications.

4.1 Design

The following requirements have been identified and set for the prototyping: The primary requirement for the prototype application is to connect to a MTConnect Agent and monitor data from it. The secondary requirement is using an OPC UA Server as an intermediate between the MTConnect Agents and the data collection application. This would provide a future-proof solution as OPC UA claims that it can transfer other data models, e.g. MTConnect, within its' information model. Tertiary requirement is to have the server auto-discover MTConnect devices that are added, to make the system integration simpler. This will be verified by comparing the ease of connecting to that of connecting a printer to a PC.

Figure 41 depicts the proposed concept use case for the prototype. This is alike to the ISV use case depicted in the OPC UA – MTConnect Companion Specification (Figure 38). For the first requirement, the data monitoring solution will consist of a MTConnect client. In subsequent requirements, the prototype will be iterated upon.

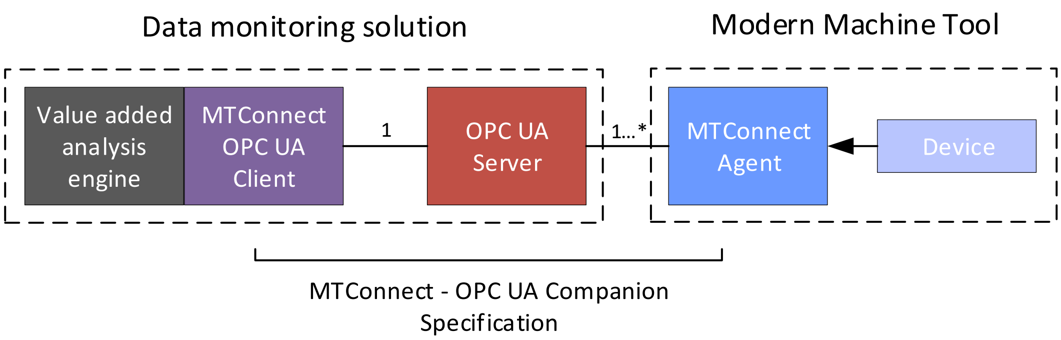


Figure 41. *Prototype application use case.*

To create a robust, simple solution, this thesis will make leverage of existing commercial applications as far as possible. Klocke et al (2017) have defined a robust technology chain design methodology (Figure 42) and the feature oriented point of view of it is utilized. A thorough documentation of the process would be enough for a thesis of its' own, therefore this thesis will only lightly touch on the subject. Financial Times have published an article (Outsourcing: Robust, proven technology is the key 2010) about how innovative use of existing, proven technology is important in challenging financial circumstances.

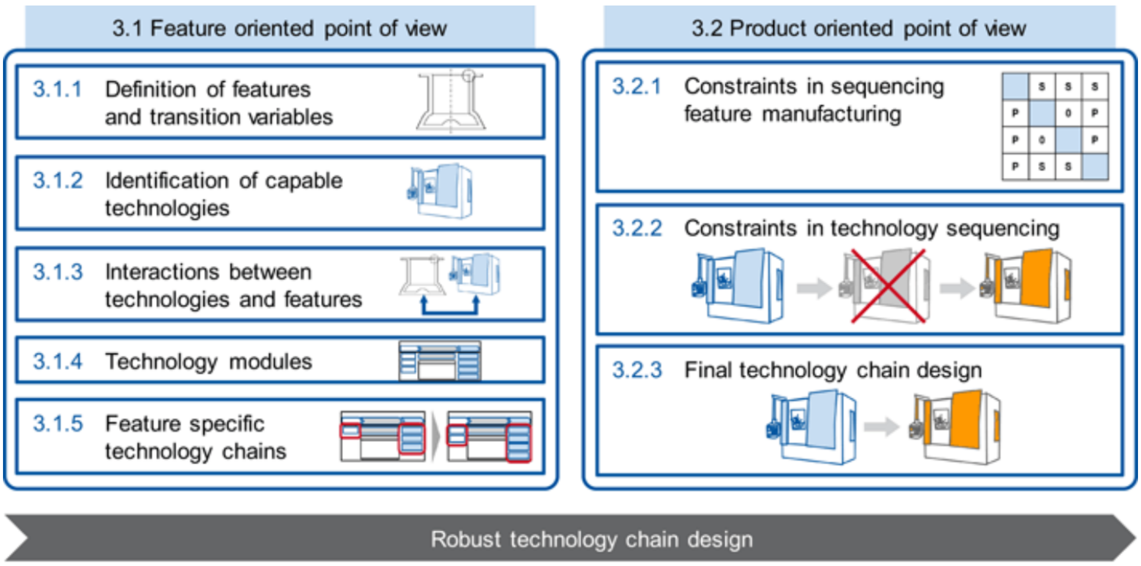


Figure 42. *Main steps of the methodology for robust technology chain design (Klocke et al. 2017).*

Based on features needed, and provided by KEPServerEX Connectivity Platform (KEPServerEX V6 Manual 2017), KEPServerEX will be used as the OPC UA server due to its' OPC Certified MTConnect driver (MTConnect Driver Manual 2017). OPC Foundation's OPC UA sample client (Unified Architecture .NET Reference Implementations) will be used as base for the OPC UA client implementation due to its' ease of availability.

The OPC UA client will need an interface for third-party applications, so that a separate value-added analysis engine can utilize it robustly. According to Koskimies & Mikkonen

(2005), a good component based architecture is scalable, efficient, robust, clear and intelligible. A Local Discovery Server extension is not required in the prototype application as the network (Figure 41) only has one OPC UA server, Figure 27 illustrates the prototype use case in the discovery layer.

Having metadata along with differently grouped data is possible with the XML data structure, which is inherently supported by OPC UA and MTConnect (3.2, 3.3). MTConnect fits directly in the "Industry Standards Information Model" slot of the OPC UA stack (Figure 26) thanks to the OPC UA-MTConnect Companion Specification (MTConnect – OPC UA Companion Specification), which provides information how the MTConnect information model should be mapped to the OPC UA data model.

Implementation can be split into several steps based on available foresight:

1. Create a client that can connect to an MTConnect device
2. Have the client monitor data from the MTConnect device
3. Setup an OPC UA server and connect the MTConnect device to it
4. Have a client monitor the MTConnect data through the OPC UA Server
5. Create interface to the OPC UA client to feed select data outward
6. Have the OPC UA Server able to auto-discover connected MTConnect devices

4.2 Implementation

The MTConnect client was straightforward to create with the use of .NET MTConnect library, installed from the NuGet package manager. As section 3.3 describes, MTConnect broadcasts its' data in XML and it can be retrieved via simple HTTP GET commands. The client can connect to a single MTConnect agent and read the component structure of the devices (Figure 43, point 1), monitor the overall status of said devices (Figure 43, point 2) and upon double clicking a device in the component view, all other sample values can be viewed (Figure 43, point 3). As the MTConnect agent, we used the test agent at agent.mtconnect.org, provided by the MTConnect Institute.

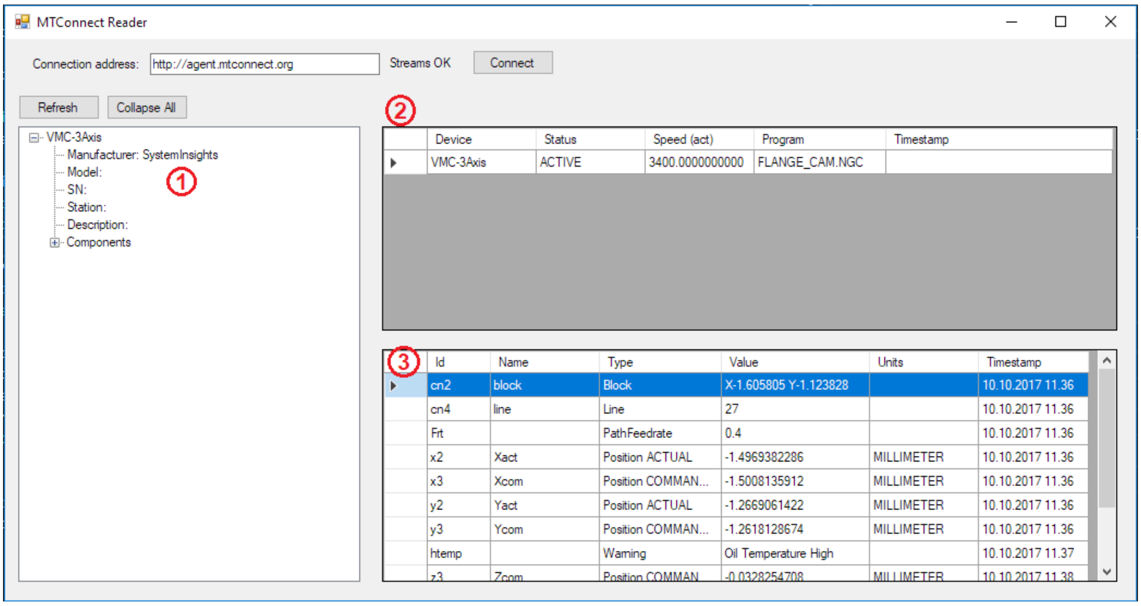


Figure 43. *MTConnect client prototype.*

A Program main class instantiates the graphical user interface (GUI) and the MTCReader, latter of which is used to create the data link to the MTConnect agent. The ClientForm realizes the GUI and uses the MTCReader to connect and retrieve data (Figure 44).

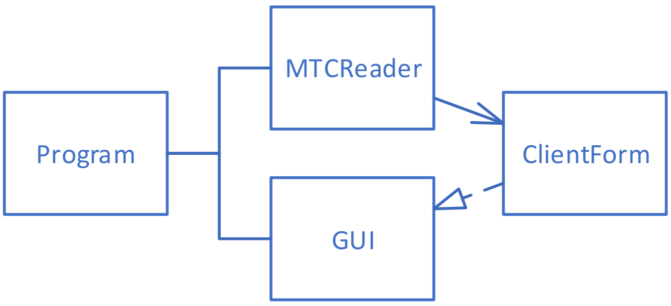


Figure 44. *MTConnect client prototype architecture.*

The OPC UA Server was implemented as planned with KEPServerEx because of its' existing MTConnect Driver. The driver simply maps the MTConnect information model to the OPC UA data model in accordance to the companion specification (MTConnect – OPC UA Companion Specification), thus saving a lot of time in implementation (MTConnect Driver Manual 2017). As the OPC UA Client, OPC Foundation's OPC UA Sample Client was used since it already contains all the basic functionalities of a client (Mahnke et al. 2009). MTConnect Institute's own test agent from "http://agent.mtconnect.org" address was used to simulate the MTConnect device. The prototype setup is depicted in Figure 45, which closely represents the designed use case (Figure 41).

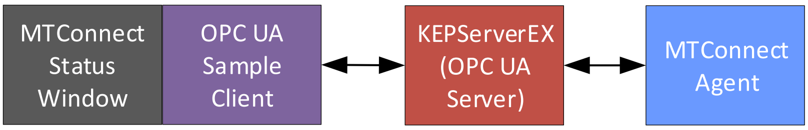


Figure 45. *Implemented prototype setup.*

It was straightforward to first add a MTConnect channel to the KEPServerEx OPC UA Server and then add the MTConnect devices to that (Figure 46). This could be all done via the right click menu and following the device wizard that pops up.

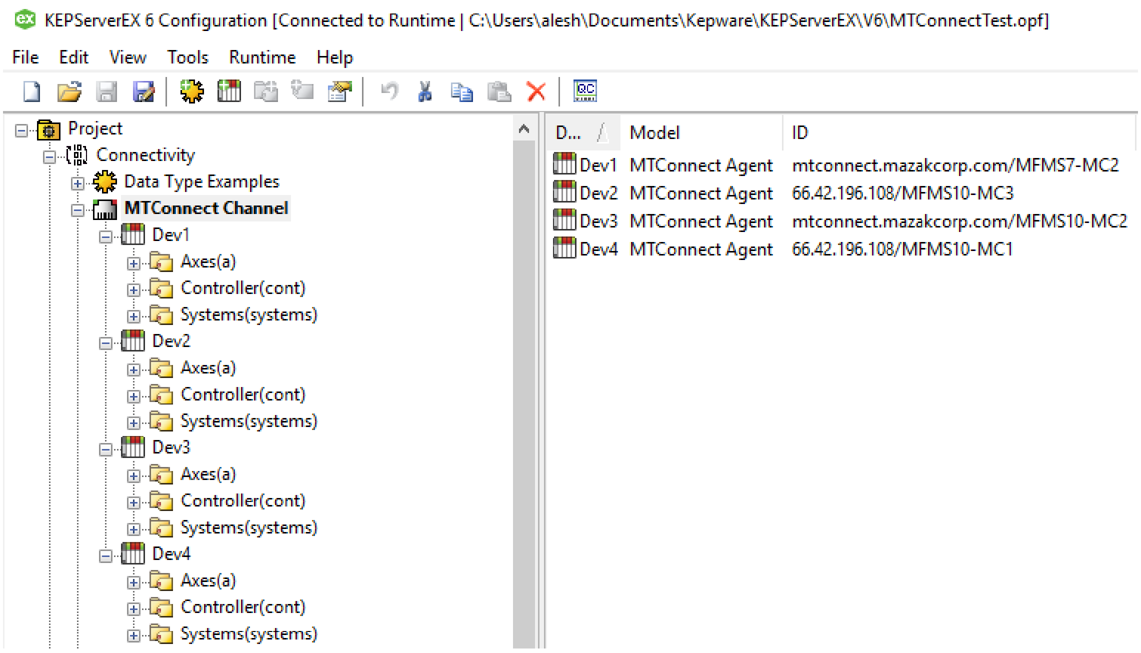


Figure 46. *MTConnect devices in KEPServerEx 6.*

Automatic discovery of MTConnect devices seems to be possible from the server side, but this thesis was unable to confirm if it works with MTConnect devices due to lack of available physical MTConnect devices. The Device Discovery (Figure 47) is based on scanning through a range of IP addresses, but it didn't work through the internet with the MTConnect devices shown in Figure 46.

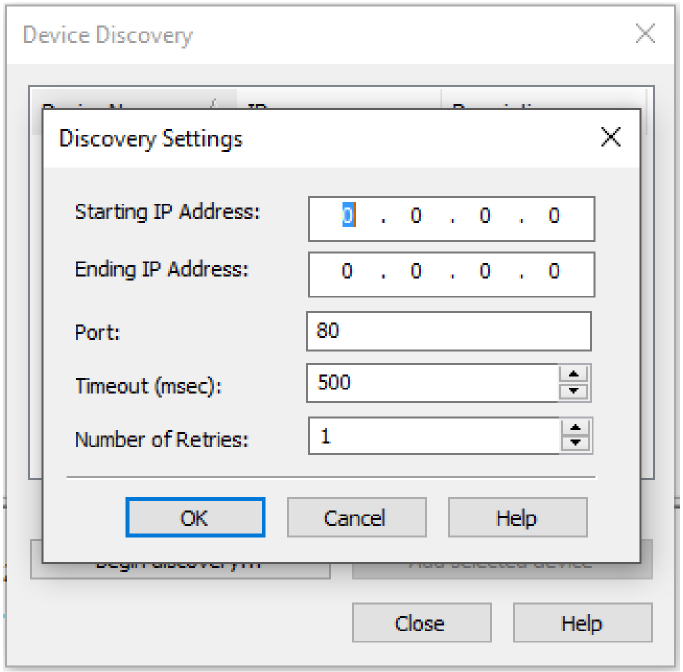


Figure 47. *KEPServerEX 6 MTConnect Device Discovery window.*

Without automatic device discovery, devices need to be added manually with their exact address and name (Figure 48).

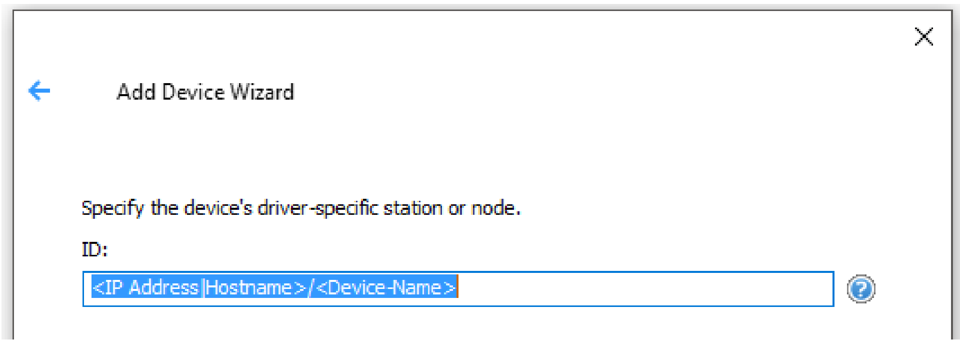


Figure 48. *KEPServerEX 6 Add Device Wizard.*

In order to connect to the server and view the MTConnect data, a OPC UA sample client (Figure 49) provided by OPC Foundation was used (OPC UA .NET). The client can browse the address space of an OPC UA Server, read nodes, write to nodes and subscribe to nodes.

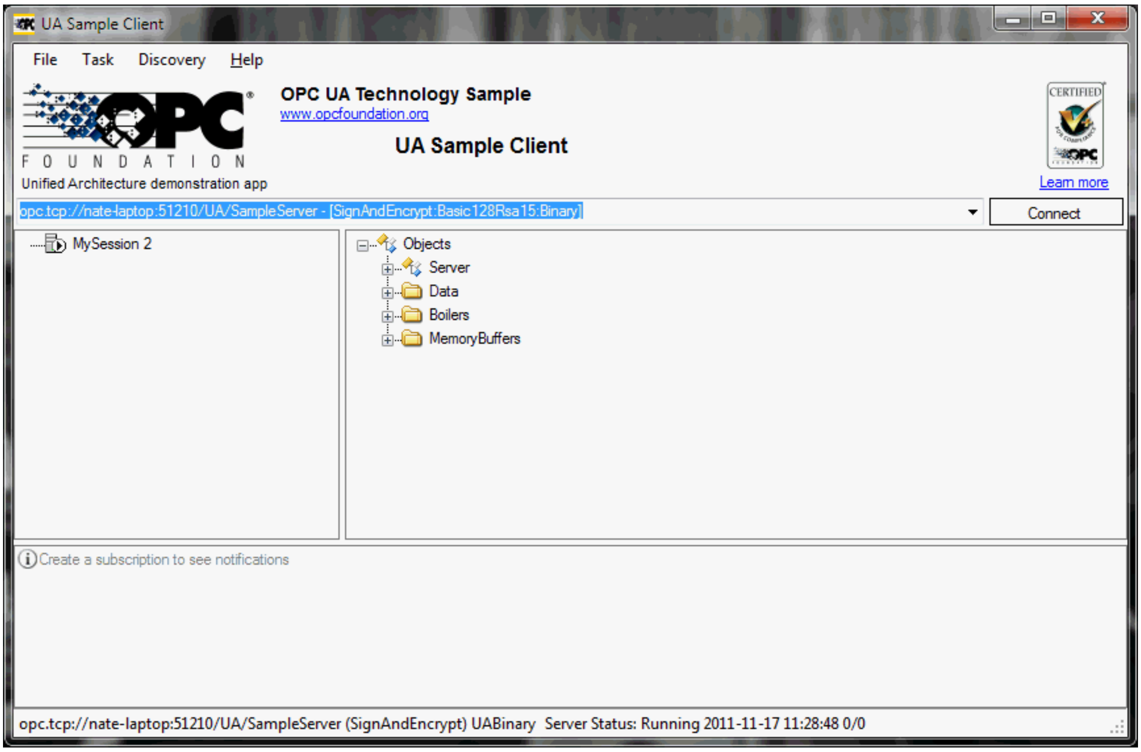
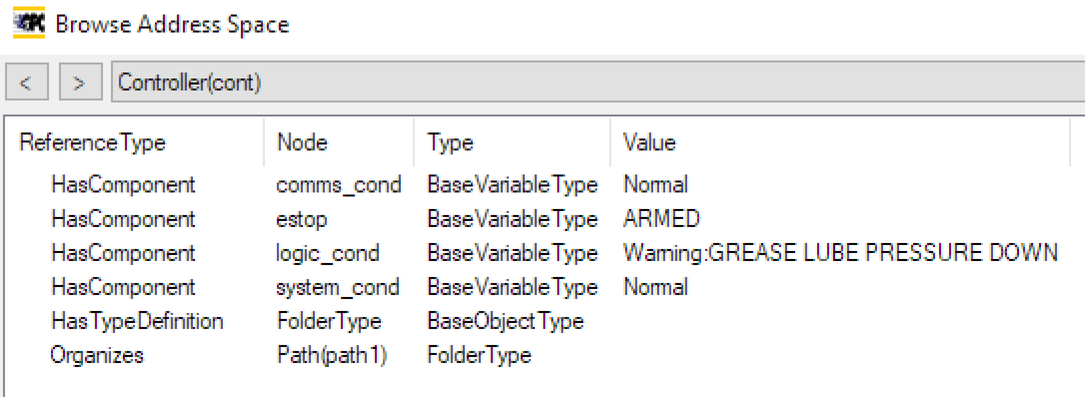


Figure 49. OPC UA Sample Client (OPC UA .NET).

The sample client does not provide an easy under-the-hood method for forwarding subscribed nodes to an external interface, but it can subscribe to MTConnect data nodes in MTConnect devices that were manually connected to the KEPServerEX server. Figure 50 illustrates how the MTConnect data is visible from the OPC UA Sample Client. To see specific data items, the user needs to right click the desired node and select browse.



ReferenceType	Node	Type	Value
HasComponent	comms_cond	BaseVariableType	Normal
HasComponent	estop	BaseVariableType	ARMED
HasComponent	logic_cond	BaseVariableType	Warning:GREASE LUBE PRESSURE DOWN
HasComponent	system_cond	BaseVariableType	Normal
HasTypeDefinition	FolderType	BaseObjectType	
Organizes	Path(path1)	FolderType	

Figure 50. MTConnect data in OPC UA Sample Client.

The user can also subscribe to specific data items and group subscriptions. Figure 51 shows what the subscription window looks like.

Item	Variable	Value	Status	Source Time	Server Time
[4]	execution	STOPPED	Good	21.57.15.258	21.57.15.258
[5]	mode	MANUAL	Good	21.57.15.258	21.57.15.258
[6]	program	M239-2	Good	21.57.15.258	21.57.15.258

Figure 51. *OPC UA Sample Client subscription window.*

Figure 52 shows an alternative visualization of OPC UA nodes from the KEPServerEX's OPC Quick Client.

Item ID	Data ...	Value	Timestamp	Qu...	Update Count
MTConnect Channel.Dev4.Axes(a).Linear(x).Xfrt	Double	1,46167	20:43:43.013	Good	35
MTConnect Channel.Dev4.Axes(a).Linear(x).Xload	Double	42	20:43:43.013	Good	26
MTConnect Channel.Dev4.Axes(a).Linear(x).Xtravel	String	Normal	20:33:19.474	Good	1
MTConnect Channel.Dev4.Axes(a).Linear(y).Yfrt	Double	0	20:43:43.013	Good	23
MTConnect Channel.Dev4.Axes(a).Linear(y).Yload	Double	2	20:34:03.416	Good	2
MTConnect Channel.Dev4.Axes(a).Linear(y).Ytravel	String	Normal	20:33:19.474	Good	1
MTConnect Channel.Dev4.Axes(a).Linear(z).Zabs	Double	-4,98545	20:43:43.013	Good	42
MTConnect Channel.Dev4.Axes(a).Linear(z).Zfrt	Double	0	20:43:43.013	Good	51
MTConnect Channel.Dev4.Axes(a).Linear(z).Zload	Double	1	20:43:20.504	Good	25
MTConnect Channel.Dev4.Axes(a).Linear(z).Ztravel	String	Normal	20:33:19.474	Good	1
MTConnect Channel.Dev4.Axes(a).Rotary(br).Bdeg	Double	90	20:33:19.473	Good	1
MTConnect Channel.Dev4.Axes(a).Rotary(br).Bfrt	Double	0	20:33:19.473	Good	1
MTConnect Channel.Dev4.Axes(a).Rotary(br).Bload	Double	0	20:33:19.473	Good	1
MTConnect Channel.Dev4.Axes(a).Rotary(br).Brfunc	String	Unknown	20:33:19.473	Bad	1

Figure 52. *MTConnect data in KEPServerEx, viewed with KEPServerEx OPC Quick Client.*

During the thesis, the MTConnect Agent at http://agent.mtconnect.org was moved to https://smstestbed.nist.gov/vds address that required Transport Layer Security (TLS) 1.2, which in turn required .NET Framework 4.6. This effectively stopped the OPC UA Server's connection to the MTConnect Agent as the current version of KEPServerEX is implemented with .NET Framework 4.5. Due to bad availability, the reference MTConnect Agent was changed to ones provided by MazakUSA in at http:// mtconnect.mazakcorp.com.

As illustrated in Figure 41, the remaining part of the test setup is the analysis engine. For this thesis, access was given to a web-based analysis engine that has a JavaScript Object Notation (JSON) over HTTP interface for data input. On the sender side, the data is first input into a series of data rows that contain a device header, status and event information. It is then serialized into a JSON object (Figure 53) that is sent to the data analysis engine through a HTTP web request.

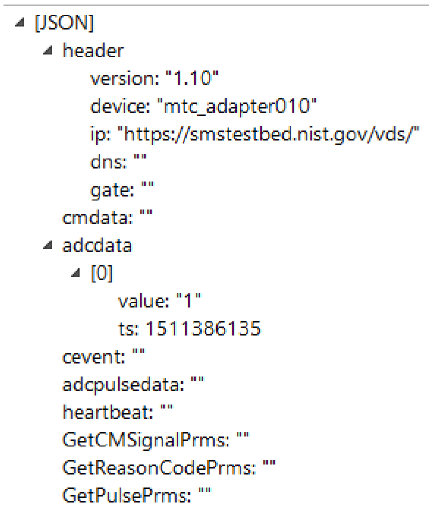


Figure 53. *JSON object for the data analysis engine.*

The analysis engine accepts 4 bytes as status values and each bit's meaning can be configured on the analysis engine side through a web page. However, this means that mapping different statuses of a MTConnect device into bytes must be done on the MTConnect or OPC UA client side.

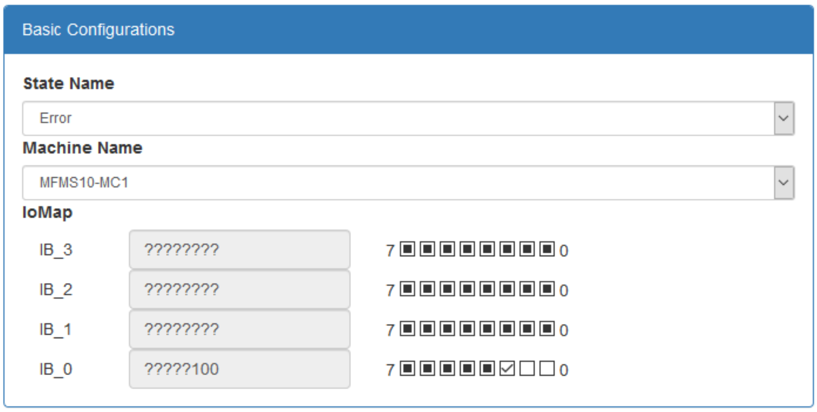


Figure 54. *Mapping of the status bytes in the analysis engine.*

Due to the large amount of overhead complexity in the OPC UA sample client, this mapping was done straight to the MTConnect client prototype (Figure 43) to provide faster results. It simply reads the execution state and the operation mode to deduce whether machine is running in automatic or manual mode, if it is in error mode etc. Depending on the mode, it will send a value ranging from 1 bit to 4 bytes along with the device ID and timestamp to the data analysis engine. Over a period of testing, the analysis engine's web interface shows the collected data as illustrated in Figure 55.

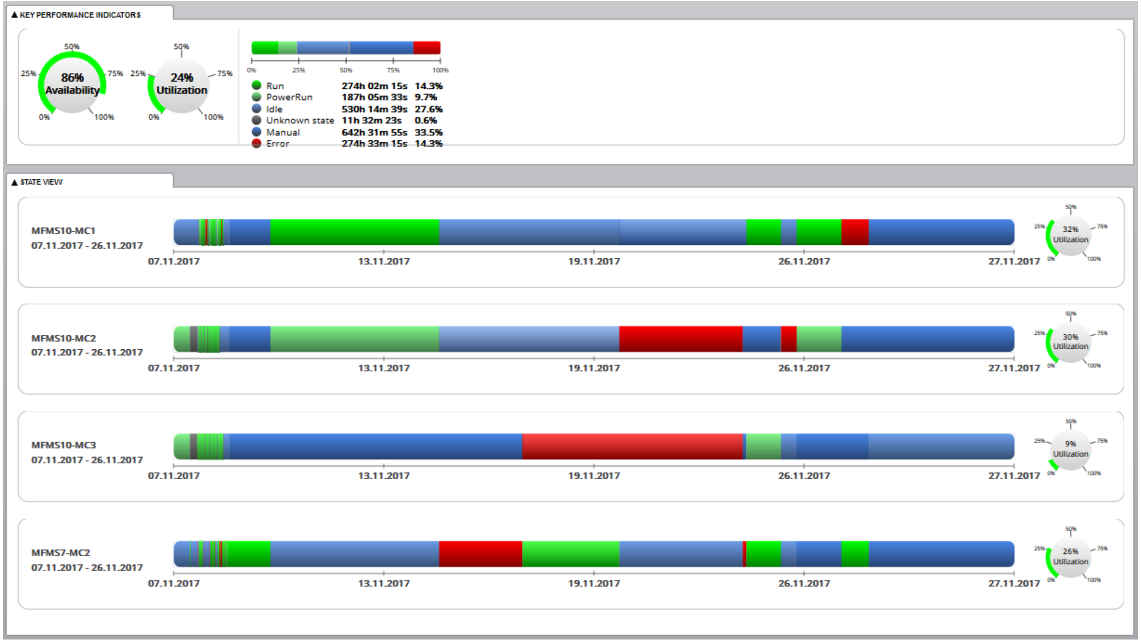


Figure 55. *Data analysis engine showing sent data from the MTConnect client.*

5. RESULTS AND DISCUSSION

Like UPnP, OPC UA is built to use many older, mature technologies such as TCP/IP and HTTP precisely because of their success in the consumer electronics industry. The manners in which they transmit data is very similar because of this. Both are standardized, UPnP as ISO/IEC 29341 in 2008 and OPC UA as IEC 62541 in 2010. As mentioned in the UPnP standard (ISO/IEC 29341 2000), it was developed for home, office and public spaces whereas OPC UA is purely focused in connecting industrial devices. UPnP's success is based on wide standardization of device discovery and auto-configuration is based on multicasts which have been impossible with OPC UA until 2015, when the discovery specification was released. This effectively brings OPC UA closer to Industrial Plug and Play, also known as Plug and Produce (Profanter et al. 2017). Nevertheless, the level of success that UPnP has enjoyed cannot be mimed with technology alone and as Profanter et al (2017) recently noted, "this common interface is only possible if all the manufacturers agree on the same standard and way of implementation".

It is debatable whether the development towards multicasting is tasteful or not, since it inherently requires more network resources, but concurrently provides opportunities for more interconnected applications. According to Thrybom & Prytz (2010), there had been historical aversion of multicast in industrial networks based on the industrial routers' inability to handle multicast, effectively turning it into broadcast that strains low-end devices. However, interconnectivity can be more useful when it communicates to a broader number of devices, especially in later phases of Industrial Internet development (Figure 12). Van Mieghem et al (2001) did a paper comparing the efficiency of unicast and multicast in reaching an arbitrary number of peers and found that the reachable number of peers grows exponentially with each connection with an effective degree of approximately 3,2. When this is considered along with the ever-increasing network capabilities, multicast can be a good way to go. This can mean that OPC UA is bound to compete with UPnP Forum's UPnP+, which is an effort to broaden UPnP scope to the heavy industries.

Chapter 2 explored what kind of data the manufacturing industry needs to collect from the machine tools. As mentioned in section 2.4, survival, growth and profitability drive most business demands with profitability as the highest priority (Pearce 1982). Recent management trends like Lean and technological trends like Industry 4.0, along with their many variants, create a growing need for data collection applications in an age where information is viewed as the key for success. In manufacturing, P. Lade et al (2017) found the following five recurring topics in manufacturing where analytics are sorely needed:

- Reducing test time and calibration
- Improving quality
- Reducing warranty cost

- Improving yield
- Performing predictive maintenance.

These require that each action, component and phase in the manufacturing process needs to be identified uniquely to each machine and program with a timestamp (Lin et al. 2015; Downey et al. 2016). It was found in section 3.2.2 that the OPC UA's information modelling allows data to be organized in object-oriented manner. In effect, this enables hierarchal data grouping to, e.g. devices and their components, with each node capable of being embedded with metadata that describes how the data should be interpreted.

The prototype application showed that MTConnect devices can be linked to an OPC UA server with the help of the OPC UA and MTConnect companion specification, which provides clear lists of how different datatypes link together between the standards. However, implementing the OPC UA-MTConnect integration was more time-consuming in an environment where MTConnect is the only non-OPC UA data model. A MTConnect client integration into a data collection application, with little over 500 lines of code plus MTConnect .NET and JSON libraries, was faster and less complex to implement than the proposed prototype setup (Figure 45), in which the OPC UA sample client alone is over 12000 lines of code, not including stack libraries. There are several commercial SDKs available that abstract much of it. Admittedly, OPC UA provides a robust security layer that is plainly non-existent in MTConnect.

However, in a case where a manufacturer may have many different devices that use multiple different data model standards, a middleware OPC UA server could potentially decrease engineering costs in the long-term. If a companion specification exists, the driver component only needs to be coded once, and can be used in other devices utilizing the same data model, with only a fraction of the integration time used. Should the standardization effort of OPC Foundation continue, perhaps different manufacturers will eventually agree on a common way of implementation resulting in true plug and produce. This would not only make the integration job easier, but also provide production flexibility with less investments, resulting in major cost savings which is largely in line with the aspirations of lean operations. These factors all follow the phases of the evolution of the Industrial Internet (Figure 12). As the integration effort gets easier and cheaper, it might ease the threshold on whether companies should embrace new technology.

Sending the data further to the data analysis application is essentially a separate part that can be implemented on top of either the MTConnect client or the OPC UA client. Another option, more robust option would be to integrate the client directly to the configuration tool of the data analysis application. When implemented properly, the user could browse data nodes and select e.g. "If this has value ACTIVE, switch machine state to running.". In the prototype this kind of mapping is done under the hood in the code. When compared to UPnP, the ease is not yet at the same level. The strength here is UPnP’s auto-configuration, which communicates the critical interfaces between the devices. This can be seen in Windows operating systems as the automatic driver installations.

6. SUMMARY

This thesis provided a view into the interconnectivity needs in the manufacturing industry and how management and technological trends reinforce those needs. The investigation for connectivity and data modelling possibilities were focused to OPC UA and MTConnect due to business needs of a small industrial informatics engineering company. Universal Plug and Play was used as an ideal example of easy information integration to be compared against.

Connectivity prototyping was done with KEPServerEx as the OPC UA Server, OPC Foundation's OPC UA Sample Client as the OPC UA Client and a self-coded MTConnect client. Freely available MTConnect agents provided by MTConnect Institute, NIST and MazakUSA were used to gain a proper test data feed in MTConnect format. Results showed that both MTConnect only and MTConnect through OPC UA routes are possible, albeit the OPC UA route requires more initial coding work, but possibly reducing work in the long term depending on the need of other industries' data models. MTConnect-OPC UA companion specification provides clear lists of how to map different types of MTConnect data into OPC UA nodes. This has enabled companies like Kepware to prepare MTConnect drivers for their OPC UA servers.

If the manufacturing environment only has MTConnect devices, it is a lot simpler not to use OPC UA if other means are used to ensure the data security. OPC UA provides built-in security and many other extra features. When compared to UPnP, the ease of connectivity is not there yet. This thesis was unable to confirm how easy to use KEPServerEx's MTConnect auto-discovery feature is, but it is still another click compared to just plugging UPnP devices in. Furthermore, a lot of manual configuration is needed to forward the data ahead from either the OPC UA server or the MTConnect client. Once this configuration is done though, adding new devices gets easier, but not as easy as connecting UPnP devices where the UPnP's auto-configuration has devices share their operable interfaces without user interaction.

REFERENCES

Booth, R. (1995). In the market, Manufacturing Engineer, Vol. 74, Iss. 5, pp. 236-239. Available: http://ieeexplore.ieee.org.libproxy.tut.fi/document/489912/.

Brodie, M.L. (2010). Data Integration at Scale: From Relational Data Integration to Information Ecosystems, Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, WA, Australia, Apr 20-23, 2010, pp. 2-3.

Butterfield, A. & Ngondi, G.E. (2016). A Dictionary of Computer Science, 7th ed. Oxford University Press, 627 p.

Cavalieri, S. & Chiacchio, F. (2013). Analysis of OPC UA performances, Computer Standards & Interfaces, Vol. 36, Iss. 1, pp. 165-177. Available: http://www.sciencedirect.com.libproxy.tut.fi/science/article/pii/S0920548913000640.

Che, T. & Liu, R.L. (2013). Networked Connection of NC Machine Tools Based on MTConnect, Applied Mechanics and Materials, Vol. 418, pp. 229.

Conlin, R.S. (2003). Driving corporate change with technology and pay for performance, Workspan, Vol. 46, Iss. 8, pp. 42-45.

Corrêa, H.L. (2001). Agile Manufacturing as the 21st Century Strategy for Improving Manufacturing Competitiveness, in: Gunasekaran, A. (ed.), Agile Manufacturing: The 21st Century Competitive Strategy, Elsevier Science Ltd, Oxford, pp. 3-23.

Downey, J., O'Sullivan, D., Nejmen, M., Bombinski, S., O'Leary, P., Raghavendra, R. & Jemielniak, K. (2016). Real time monitoring of the CNC process in a production environment- the data collection and analysis phase, Procedia CIRP 41, Ischia, Italy, Jun 24-26, 2015, ScienceDirect, pp. 920-926.

El-Hofy, H. & Youssef, H.A. (2008). Machining Technology, CRC Press, Boca Raton, 672 p.

Evans, P.C. & Annunziata, M. (2012). Industrial Internet: Pushing the Boundaries of Minds and Machines, pp. 37. Available: http://www.ge.com/docs/chapters/Industrial_Internet.pdf.

Fielding, R.T. (2000). Architectural styles and the design of network -based software architectures, ProQuest Dissertations Publishing, 24 p.

Microsoft introduces new open-source cross-platform OPC UA support for the industrial Internet of Things, Microsoft, website. Available (accessed on 23.3.2017): https://blogs.microsoft.com/iot/2016/06/23/microsoft-introduces-new-open-source-cross-platform-opc-ua-support-for-the-industrial-internet-of-things/

Gilad, T. & Gilad, B. (1986). Business Intelligence -- The Quiet Revolution, Sloan management review, Vol. 27, Iss. 4, pp. 53-61. Available: https://search-proquest-com.libproxy.tut.fi/docview/224963827?accountid=27303.

Gilchrist, A. (2016). Industry 4.0: The Industrial Internet of Things, Apress, 258 p.

Goldman, S.L., Nagel, R.N. & Preiss, K. (1995). Agile competitors and virtual organizations, Van Nostrand Reinhold, New York, USA, 414 p.

Grimmett, J. & O'Neill, E. (2012). UPnP: Breaking out of the LAN, Proceedings of the 2012 IEEE Wireless Communications and Networking Conference Workshops, Paris, France, Apr 1, 2012, IEEE, pp. 170-174.

Gross, J.M. & McInnis, K.R. (2003). Kanban Made Simple: Demystifying and Applying Toyota's Legendary Manufacturing Process, AMACOM, New York, 259 p.

Gunasekaran, A., McGaughey, R. & Wolstencroft, V. (2001). Agile Manufacturing: Concepts and Framework, in: Gunasekaran, A. (ed.), Agile Manufacturing: The 21st Century Competitive Strategy, Elsevier Science Ltd, Oxford, pp. 25-49.

Haas, L. (2006). Beauty and the Beast: The Theory and Practice of Information Integration, in: Anonymous (ed.), Database Theory – ICDT 2007, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 28-43.

Hannelius, T., Salmenperä, M. & Kuikka, S. (2008). Roadmap to adopting OPC UA, Proceedings of the 2008 6th IEEE International Conference on Industrial Informatics, Daejeon, South Korea, Jul 13-16, 2008, IEEE, pp. 756-761.

Helin, P. (2010). The profitability change of few major car manufacturers during the years 1998-2008, Tampere University of Technology, 126 p.

Hevner, A.R., March, S.T., Park, J. & Ram, S. (2004). Design Science in Information Systems Research, MIS Quarterly, Vol. 28, Iss. 1, pp. 75-105. Available: http://www.jstor.org/stable/25148625.

Hoefling, M., Heimgaertner, F., Fuchs, D., Menth, M., Romano, P., Tesfay, T., Paolone, M., Adolph, J. & Gronas, V. (2015). Integration of IEEE C37.118 and publish/subscribe communication, Proceedings of the 2015 IEEE International Conference on Communications, London, UK, Jun 8-12, 2015, IEEE, pp. 764-769.

Hoffmann, M., Thomas, P., Schutz, D., Vogel-Heuser, B., Meisen, T. & Jeschke, S. (2016). Semantic integration of multi-agent systems using an OPC UA information modeling approach, Proceedings of the 2016 IEEE 14th International Conference on Industrial Informatics, Poitiers, France, Jul 19-21, 2016, IEEE, pp. 744-747.

Häber, A. (2007). Service Discovery, in: Fitzek, F.H.P. & Reichert, F. (ed.), Mobile Phone Programming: Application to Wireless Networking, Springer Netherlands, Dordrecht, pp. 239-255.

Iatrou, C.P. & Urbas, L. (2016). Efficient OPC UA binary encoding considerations for embedded devices, Proceedings of the 2016 IEEE 14th International Conference on Industrial Informatics, Poitiers, France, Jul 19-21, 2016, IEEE, pp. 1148-1153.

Industrial Internet of Things: Unleashing the Potential of Connected Products and Services, (2015). World Economic Forum, 40 p. Available: http://www3.weforum.org/docs/WEFUSA_IndustrialInternet_Report2015.pdf.

Industrie du Futur: Réunir la Nouvelle France Industrielle, (2015). The Government of the French Republic, 64 p. Available: https://www.economie.gouv.fr/files/files/PDF/industrie-du-futur_dp.pdf.

Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model ISO/IEC 7498-1. 1994.

Inman, R.A., Sale, R.S., Green, K.W. & Whitten, D. (2011). Agile manufacturing: Relation to JIT, operational performance and firm performance, Journal of Operations Management, Vol. 29, Iss. 4, pp. 343-355. Available: http://www.sciencedirect.com.libproxy.tut.fi/science/article/pii/S0272696310000458.

#TBT: A Brief History of Microsoft's Failed Attempts at Home Automation, Electronic House Publishing, website. Available (accessed on 7.9.2017): http://www.cepro.com/article/tbt_a_brief_history_of_microsofts_failed_attempts_at_home_automation.

Jarke, M., Jeusfeld, M. & Quix, C. (2014). Data-centric intelligent information integration—from concepts to automation, Journal of Intelligent Information Systems, Vol. 43, Iss. 3, pp. 437-462. Available: http://search.proquest.com/docview/1628175859.

Kagermann, H., Helbig, J., Hellinger, A. & Wahlster, W. (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0, Forschungsunion, Berlin, Germany, 82 p.

Kalibataitė, G. (2014). Integration of Activity Data: Problems, Peculiarities and the Importance of Metadata, Socialnės Technologijos, Vol. 4, Iss. 1, pp. 93-117. Available: http://www.ceeol.com/search/article-detail?id=117513.

Kenderdine, T. (2017). China's Industrial Policy, Strategic Emerging Industries and Space Law, Asia & the Pacific Policy Studies, Vol. 4, Iss. 2, pp. 325-342. Available: http://onlinelibrary.wiley.com/doi/10.1002/app5.177/abstract.

KEPServerEX V6 Manual, PTC, 2017, pp. 266.

Keskinen, J. (2017). Enablers for Agile Business Intelligence – Case SAP, Tampere University of Technology, 98 p. Available: http://dspace.cc.tut.fi/dpub/handle/123456789/24531.

Kief, H.B. (2011). CNC Handbook, McGraw Hill, New York, 480 p.

Kiran, S. (2017). Effectiveness of Real-time Business Intelligence on Enterprise Performance Management: a Systematic Literature Review, Tampere University of Technology, 76 p. Available: http://dspace.cc.tut.fi/dpub/handle/123456789/24813.

Klocke, F., Mattfeld, P., Stauder, J., Müller, J. & Grünebaum, T. (2017). Robust technology chain design: considering undesired interactions within the technology chain, Production Engineering, Vol. 11, Iss. 4, pp. 575-585. Available: https://doi.org/10.1007/s11740-017-0756-1.

Korea Releases Middle to Long-Term Policy Tasks, South Korea's Ministry of Strategy and Finance, website. Available (accessed on 6.9.2017): http://www.korea.net/Government/Briefing-Room/Press-Releases/view?articleId=1896.

Koskimies, K. & Mikkonen, T. (2005). Ohjelmistoarkkitehtuurit, Talentum, Helsinki, 250 p.

Kozar, S. & Kadera, P. (2016). Integration of IEC 61499 with OPC UA, Proceedings of the 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation, Berlin, Germany, Sep 6-9, 2016, IEEE, pp. 1-7.

Laihonen, H., Hannula, M., Helander, N., Ilvonen, I., Jussila, J., Kukko, M., Kärkkäinen, H., Lönnqvist, A., Myllärniemi, J., Pekkola, S., Virtanen, P., Vuori, V. & Yliniemi, T. (2013). Tietojohtaminen, Tampereen teknillinen yliopisto, Tietojohtamisen tutkimuskeskus Novi, 86 p.

Lawton, G. (2006). Making Business Intelligence More Useful, Computer, Vol. 39, Iss. 9, pp. 14-16. Available: http://ieeexplore.ieee.org.libproxy.tut.fi/document/1703302/.

Liker, J.K. (2004). Toyota Way: 14 Management Principles from the World's Greatest Manufacturer, McGraw-Hill Education, 330 p.

Lin, Y., Ieromonachou, P. & Sun, W. (2017). Smart manufacturing and supply chain management, Sydney, Australia, Jul 24-27, 2016, IEEE, pp. 1-5.

Lin, Y., Lin, C. & Chiu, H. (2015). Developing a cloud virtual maintenance system for machine tools management, Proceedings of the 11th EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), Taipei, Taiwan, Aug 19-20, 2015, pp. 358-364.

Lindström, O.M. (2015). Standardized Collection of Production Data in Factory Environment, Tampere University of Technology, Available: http://dspace.cc.tut.fi/dpub/handle/123456789/22802.

Lofgren, S. (2015). UPnP: The Discovery and Service Layer For The Internet of Things, Available: http://upnp.org/resources/whitepapers/UPnP_Internet_of_Things_Whitepaper_2015.pdf.

Mahnke, W., Leitner, S. & Damm, M. (2009). OPC Unified Architecture, Springer Berlin Heidelberg, 351 p.

Miller, B.A., Nixon, T., Tai, C. & Wood, M.D. (2001). Home networking with Universal Plug and Play, IEEE Communications Magazine, Vol. 39, Iss. 12, pp. 104-109. Available: http://ieeexplore.ieee.org.libproxy.tut.fi/document/968819/.

MTConnect Driver Manual, PTC, 2017, pp. 27.

MTConnect User Portal, MTConnect Institute, website. Available (accessed on 10.10.2017): http://mtcup.org/

MTConnect – OPC UA Companion Specification, MTConnect Institute, website. Available: http://www.mtconnect.org/opc-ua-companion-specification/.

MTConnect® Standard, (2014). MTConnect Institute, 69 p. Available: http://www.mtconnect.org/standard-documents.

New Robot Strategy, (2015). Japan's Ministry of Economy, Trade and Industry, 91 p. Available: http://www.meti.go.jp/english/press/2015/pdf/0123_01b.pdf.

Notice of the State Council on Printing and Distributing "Made in China 2025"
, The State Council of the People's Republic of China, website. Available (accessed on 6.9.2017): http://www.gov.cn/zhengce/content/2015-05/19/content_9784.htm

OPC UA .NET, OPC Foundation, website. Available (accessed on 27.7.2017): http://opcfoundation.github.io/UA-.NET/help/.

OPC UA Specification Part 12: Discovery, (2015). OPC Foundation, 69 p. Available: https://opcfoundation.org/developer-tools/specifications-unified-architecture.

OPC UA Specification Part 14: PubSub, (2017). OPC Foundation, 118 p. Available: https://opcfoundation.org/developer-tools/specifications-unified-architecture.

OPC Unified Architecture IEC 62541. 2010. Available: https://opcfoundation.org/developer-tools/specifications-unified-architecture.

P. Lade, R. Ghosh & S. Srinivasan (2017). Manufacturing Analytics and Industrial Internet of Things, IEEE Intelligent Systems, Vol. 32, Iss. 3, pp. 74-79.

Palm, F., Gruner, S., Pfrommer, J., Graube, M. & Urbas, L. (2015). Open source as enabler for OPC UA in industrial automation, 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation, Luxembourg, Luxembourg, Sep 8-11, 2015, IEEE, pp. 1-6.

Pearce, J.A., II (1982). The Company Mission as a Strategic Tool, Sloan Management Review (pre-1986), Vol. 23, Iss. 3, pp. 15. Available: https://search-proquest-com.libproxy.tut.fi/docview/206766548?accountid=27303.

Ponomarjovs, A. (2013). Business Value of Business Intelligence, Tampere University of Technology, 99 p. Available: http://dspace.cc.tut.fi/dpub/handle/123456789/21689.

Priorities, Infrastructure, and Collaboration for Implementation of Smart Manufacturing , (2012). Smart Manufacturing Leadership Coalition (SMLC), 27 p. Available: https://smartmanufacturingcoalition.org/sites/default/files/smlc_forum_report_vf_0.pdf.

Outsourcing: Robust, proven technology is the key. (2010). Financial Times, Available: https://www.ft.com/content/a3df3b40-01b9-11e0-9c3e-00144feab49a.

Profanter, S., Dorofeev, K., Zoitl, A. & Knoll, A. (2017). OPC UA for Plug and Produce: Automatic Device Discovery using LDS-ME, Proceedings of the 2017 22nd IEEE International Conference on Emerging Technologies And Factory Automation, Limassol, Cyprus, Sep 12-15, 2017, IEEE, pp. 8.

Ramakrishnan, T., Jones, M.C. & Sidorova, A. (2012). Factors influencing business intelligence (BI) data collection strategies, in: Marsden, J.R. (ed.), Decision Support Systems, pp. 486-496.

Rentschler, M., Trsek, H. & Durkop, L. (2016). OPC UA extension for IP auto-configuration in cyber-physical systems, Proceedings of the 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), Bochum, Germany, Jul 29-31, 2013, IEEE, pp. 26-31.

Rouhiainen, O. (2015). Integration of PLC based data collection to a cloud service, Tampere University of Technology, 49 p.

Sanders, A., Elangeswaran, C., Wulfsberg,J. (2016). Industry 4.0 implies lean manufacturing: Research activities in industry 4.0 function as enablers for lean manufacturing, Journal of Industrial Engineering and Management, Vol. 9, Iss. 3, Available: http://www.jiem.org/index.php/jiem/article/view/1940.

Sauter, T. & Wollschlaeger, M. (2011). Industrial Internet, in: Wilamowski, B.M. & Irwin, J.D. (ed.), Industrial Communication Systems, CRC Press, pp. 793-806.

Schiele, G., Handte, M. & Becker, C. (2010). Pervasive Computing Middleware, in: Nakashima, H., Aghajan, H. & Augusto, J.C. (ed.), Handbook of Ambient Intelligence and Smart Environments, Springer US, Boston, MA, pp. 201-227.

Seilonen, I., Tuovinen, T., Elovaara, J., Tuomi, I. & Oksanen, T. (2016). Aggregating OPC UA servers for monitoring manufacturing systems and mobile work machines, Proceedings of the 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation, Berlin, Germany, Sep 6-9, 2016, IEEE, pp. 1-4.

Shettar, M., Hiremath, P., Rangaswamy, N. & Chauhan, V.R. (2015). KAIZEN – A case study, International Journal of Engineering Research and Applications, Vol. 5, Iss. 5, pp. 101-103. Available: https://doaj.org/article/e27f4f31f9734f23ac0aef6bc42ed499.

Sill, A. (2016). Cloud, Data, and Business Process Standards for Manufacturing, IEEE Cloud Computing, Vol. 3, Iss. 4, pp. 74-80.

Sobel, W. (2011). Getting Started with MTConnect - Connectivity Guide, MTConnect Institute, 37 p. Available: http://www.mtconnect.org/s/Getting-Started-with-MTConnect-FINAL.pdf.

Sobel, W. (2015). MTConnect Technical Workshop, MTConnect Institute, 121 p. Available: http://www.mtconnect.org/s/MTConnectTechnicalWorkshopAtNAMRC43-x6n1.pdf.

Stevenson, W.J. (2012). Operations Management, 11th ed. McGraw-Hill/Irwin, Rochester Institute of Technology, 945 p.

Stjerna, A.A. (2017). Deployment of Cloud Based Platforms for Process Data Gathering and Visualization in Production Automation, Tampere University of Technology,

Suh, S., Chung, D., Kang, S.K. & Stroud, I. (2008). Theory and Design of CNC Systems, 1. Aufl. ed. Springer Verlag London Limited, London, 466 p.

Suárez-Barraza, M.F., Dahlgaard-Park, S.M., Rodríguez-González, F.G. & Durán-Arechiga, C. (2016). In search of "Muda" through the TKJ diagram, International Journal of Quality and Service Sciences, Vol. 8, Iss. 3, pp. 377-394.

Think Tank Suggests Agile Manufacturing for U.S. Success, (1992). Machine Design, Vol. 64, Iss. 4, pp. 32. Available: https://search-proquest-com.libproxy.tut.fi/docview/217129865?accountid=27303.

Thrybom, L. & Prytz, G. (2010). Multicast filtering in industrial Ethernet networks, 2010 IEEE International Workshop on Factory Communication Systems Proceedings, pp. 185-188.

Tyrväinen, T. (2014). Business Intelligence Trends in Finland in 2013, Tampere University of Technology, 125 p. Available: http://dspace.cc.tut.fi/dpub/handle/123456789/22097.

Unified Architecture, OPC Foundation, website. Available (accessed on 9.3.2017): https://opcfoundation.org/about/opc-technologies/opc-ua/.

Unified Architecture .NET Reference Implementations, Available (accessed on 12.7.2017): https://github.com/OPCFoundation/UA-.NET.

Universal Plug and Play Device Architecture ISO/IEC 29341. UPnP Forum2000. Available: http://upnp.org/specs/arch/UPnPDA10_20000613.pdf.

Van Mieghem, P., Hooghiemstra, G. & van der Hofstad, R. (2001). On the efficiency of multicast, IEEE/ACM Transactions on Networking, Vol. 9, Iss. 6, pp. 719-732.

Vuori, V. (2005). Information Needs of Decision Makers in Seasonal Management, Tampere University of Technology, 98 p.

What is OPC? OPC Foundation, website. Available (accessed on 9.3.2017): https://opcfoundation.org/about/what-is-opc/.

Ziegler, P. & Dittrich, K.R. (2004). Three Decades of Data Integration — All Problems Solved? IFIP Advances in Information and Communication Technology, Vol. 156, pp. 3-12.