2019-03-01

Complexity of Establishing Industrial Connectivity for Small and Medium Manufacturers with and Without Use of Industrial Innovation Platforms

Brian Dale Russell
Brigham Young University

Complexity of Establishing Industrial Connectivity for Small and Medium Manufacturers with and Without Use of Industrial Innovation Platforms

Brian Dale Russell

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Yuri Hovanski, Chair
Barry M. Lunt
Jason M. Weaver

School of Technology

Brigham Young University

ABSTRACT

Complexity of Establishing Industrial Connectivity for Small
and Medium Manufacturers with and Without
Use of Industrial Innovation Platforms

Brian Dale Russell
School of Technology, BYU
Master of Science

The manufacturing industry is continuously evolving as new practices and technology are adopted to improve productivity and remain competitive. There have been three well established manufacturing revolutions in recent history and some say that the fourth is occurring currently by the name of Smart Manufacturing, Indusrie 4.0, and others. This latest manufacturing revolution is highly dependent on industrial connectivity.

This research aims to gage the ability of Industrial Innovation Platforms (IIPs) to reduce complexity of implementing base-line industrial connectivity for small and medium-sized enterprises (SMEs). The results of this study would be especially relevant to decision makers in industrial SMEs who are considering implementing industrial connectivity as well as providing insights into approaches for establishing base-line industrial connectivity.

The research methodology consists of three main steps: 1) creation of IIP and non-IIP connectivity solutions that enable connectivity of the vast amount of industrial equipment, 2) Gathering measures from solutions in accordance with metrics identified for complexity evaluation, 3) discussion and interpretation of data

To have a more complete analysis, quantitative and qualitative data was used and evaluated to address the varying elements of the broad task of establishing industrial connectivity.

The research showed that IIPs can reduce complexity for select industrial equipment. Some industrial equipment have robust and streamlined connectivity solutions provided by the IIP. In these cases, the IIP almost certainly will reduce the complexity of establishing connectivity. Other industrial equipment have a solution provided by the IIP which requires piecing together and some component modifications. In these cases, the IIPs reduce complexity of establishing connectivity dependent on circumstances. Lastly, when no form of solution is available through the IIP for the industrial equipment, the IIP's has no ability to reduce complexity other than hosting the server used in connectivity.

These findings open additional avenues of research which could improve the understanding of benefits IIPs may provide to SMEs.

Keywords: industrial connectivity, industrial innovation platform, industrial internet of things, smart manufacturing, industrie 4.0, ad-hoc connectivity

ACKNOWLEDGEMENTS

First, I would like to thank God for all his guidance and the opportunities and capabilities needed for this work. I owe a special thanks to Professor Yuri Hovanski my thesis advisor, for his tremendous willingness to work with, learn with, and guide me through this endeavor. Additionally, it was his lab funds that generously financed many of my studies. I thank my other graduate committee members, Professor Jason Weaver and Professor Barry Lunt for their guidance and sacrifice and Ruth Ann Lowe for all her help and kind encouragement. My fellow lab colleagues have also been of great help as they have selflessly shared their talents and knowledge whenever asked. I strongly believe that we, students of Brigham Young University, stand on the shoulders of giants. I want to thank all those who have gone before me who have made BYU what it is today; that includes the founders, administration, staff, fellow students, donors, and many others.

My friends and family have been a tremendous support in this endeavor, particularly my parents and grandparents. My friend Dallin Faulkner as a technology professional also provided great advice.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 Progress and Improvement in Manufacturing

The current trend in manufacturing is to incorporate technology that enhances the performance of a facility. Smart Manufacturing or Industrie 4.0 has often been referred to as the latest form of industrial revolution. Smart manufacturing is an enhanced process of production with the objective of optimizing concept generation, product transaction, and manufacturing. Smart manufacturing is made possible by merging operation technology and information technology into an integrated real-time process. Smart manufacturing usually consists of interoperability, capturing data, modeling, simulation, real-time data analysis, big data analysis, controlling and planning, and cloud computing. (TMR, 2016)

In order to establish industrial connectivity (interoperability and data capture), offerings called Industrial Innovation Platforms (IIP) have been developed. IIPs support implementation, management, and expansion of smart manufacturing in organizations

Smart manufacturing systems help to optimize processes, prevent undesirable events, increase safety, decrease product time to market, enhance supply chain interaction, and inform facility and corporate personnel with appropriate metrics and analysis.

The question of this paper is: to what extent do IIPs reduce complexity for small and medium-sized enterprises (SME) when establishing connectivity for smart manufacturing? There are only a limited number of Smart Manufacturing and Industry 4.0 roadmaps, maturity models, frameworks and readiness assessments that are available today that reflect the specific requirements and challenges of SMEs. In fact, most models have a disconnect from the practical needs of and realities of the SMEs, assuming that the companies already have access to necessary resources such as advanced and connected machines, IT integrated systems and more (Mittal, Khan, 2018). Currently there is little indicating whether SMEs would be able to afford the cost of implementing a smart factory solution. Again there is a limited amount of research focused on SMEs in the field of smart manufacturing, requiring new research to begin filling this gap (Radziwon, Bilberg, 2014).

The intent of this paper is to determine whether or not IIPs provide significant value in enabling SMEs to implement smart factories. In order to address this, an overview of smart manufacturing is given with information highlighting the difficulties faced by SMEs when implementing a smart factory. Next, the methodology of this research will be explained in detail. From there, methods of connectivity with and without the aid of an IIP will be discussed and finding about difficulty and complexity will be shared. The paper will close with a discussion of the conclusions reached, the impact of this research, and suggestions for further research.

2 REVIEW OF THE LITERATURE

2.1 Introduction to the Literature Review

In order to understand the importance of smart manufacturing for industrial SMEs, it is necessary to have an understanding of a few key elements. First, a brief overview of the progression of manufacturing will be provided. Next, a basic overview of smart manufacturing including some history and relevant terms will be introduced. Lastly, difficulties faced by SMEs in implementing smart manufacturing will be shared.

2.2 History of Manufacturing

The first industrial revolution began in the late eighteenth century with production mechanization using water and steam power. The second industrial revolution began in the early twentieth century with the onset of mass production, the assembly line and electricity. The third industrial revolution began in the early nineteen seventies with IT supported operations leading to automation. The fourth and current phase in manufacturing progress is smart manufacturing led by advances in cyber-physical systems. (Lydon, 2016)

During these phases, other frameworks and technics for manufacturing were developed and continue in use. Examples of these paradigms include lean, flexible, agile, sustainable, digital, and cloud manufacturing (Helu, Libes, 2016).

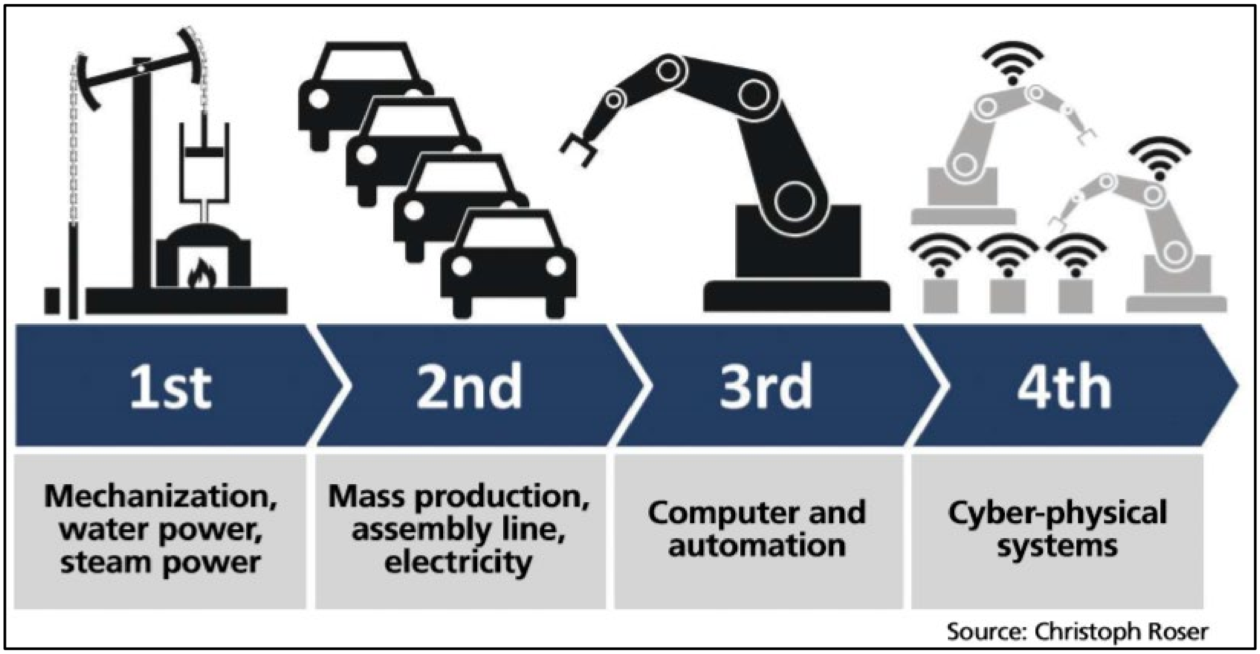

Figure 1: Industrial Revolutions

2.3 Smart Manufacturing

2.3.1 Future of Manufacturing

Smart manufacturing is the latest technological evolution in manufacturing. Through IIPs, both old and new technologies are becoming more accessible and relevant to small and medium-sized manufacturers (Wank, Adolph, 2016). IIPs have made this possible by continuing to simplify the processes of creating asset interoperability. The first step for manufacturers to implement smart manufacturing is to overcome the connectivity challenge (Mui, 2016). This has led to IIPs gaining importance.

Top analyst firms, consulting firms, and manufacturing technology companies are forecasting that Smart Manufacturing will grow tremendously. Mckinsey Global Institute estimates that smart manufacturing will, ”have an annual economic impact of $1.2 to $3.7 trillion annually by 2025” (Patel, 2017). General Electric expects investment in IIoT to top $60 trillion in the next fifteen years (GE, 2016). Gartner expects IoT to have more than 21 billion connected things by 2020 and due mostly to the impact of IIoTs (Alaybeyi, 2016). The future of smart manufacturing is looking like the future for manufacturing.

By connecting BYU’s manufacturing equipment and providing a framework for future growth, the foundation for greater understanding and research of smart manufacturing at BYU will be made possible. Methods to establish connectivity can be divided into two categories, custom and pre-built applications.

2.3.2 Industrial Internet of Things

Industrial Internet of Things (IIoT) is the foundational platform facilitating smart manufacturing. Forbes, Frost and Sulivan, McKinsey, and techtarget.com agree that IIoT is a network of assets integrated into a system enhanced with information services (Janarthanan, 2015, Manyika, 2015, Mui, 2016, Rouse, 2018). IIoT is the network of integrated assets consists of a merger of information technology and operational technology that can be managed from an IIoT platform. Information services include but are not limited to big data analytics, cloud computing, user interface, and access availability.

2.3.3 PLCs, Other Computers, WSN

PLCs (programable logic controllers) are essential to connectivity. They are the most common type of industrial control systems and industrial control systems are the systems that have historically been designed to interface with physical industrial equipment. Industrial control systems are commonly used in industrial facilities to connect, monitor, and control equipment. (Galloway, Hancke, 2013)

Computers are machines capable of processing data and performing instructions. While PLCs are a subsegment of computers, as are other forms of industrial control systems, some computers that were not designed specifically for industrial equipment interface yet are used in industrial settings. (Zhao, Jegatheesan, 2015) Raspberry Pis are one such type of computer that can be used in industrial settings.

WSNs (Wireless Sensor Networks) are used to exchange information between an application, a platform and one or more sensor nodes. This capability lends itself to assisting in data transferring for any type of sensor or actuator retrofitted on to industrial equipment. (Cecílio and Furtado, 2014)

2.3.4 Industrial Innovation Platform and Ad Hoc Solutions

IIPs are a subset of IIoTs with a focus on supporting industrial connectivity implementation (McAvoy, 2018). Like IIoTs they also support management and expansion of smart manufacturing applications.

When looking to build industrial connectivity without purchased solutions like IIPs, SMEs turn to internal development. Although there are a variety of approaches, researchers have classified most methods of building connectivity into two categories RESTful Web services and Big Web Service (Jabbar, Khan, 2018). RESTful messaging is one of the most common and practical means for individuals to apply and build connectivity solutions.

2.3.5 Information Technology and Operation Technology, an IIoT Perspective

Gartner has defined information Technology (IT) as, "the entire spectrum of technologies for information processing, including software, hardware, communications technologies, and related services. In general, IT does not include embedded technologies that do not generate data for enterprise use." Gartner defines operation technology (OT) as, "hardware and software that detects or causes a change through the direct monitoring and/or control of physical devices, processes, and events in the enterprise." (Gartner, 2019).

IIoT aims at using the latest IT to centralize the monitoring and control in a single platform. This is accomplished by enabling IT resources to interface with OT resources and culminate operational data of physical systems. With the centralization of technology in IIoT settings new opportunities arise to improve areas like big data analytics, machine optimization, and more accessible data.

2.4 The Challenges of SMEs Adopting Smart Manufacturing

Despite the drivers for investing in smart manufacturing, a significant amount of hesitation exists especially among small and medium-sized enterprise (SME). Challenges such as the cost of industrial connectivity, the risk of legacy system integration, security threats, skill shortage, and more prevent manufacturing facilities from investing in smart manufacturing (Sullivan, 2016). Connectivity, the foundational step in establishing smart manufacturing, is commonly the barrier preventing adoption. With the development of IIPs, industrial connectivity can be simplified. SMEs now have more approaches to establish industrial connectivity and to begin obtaining benefit from smart manufacturing.

2.4.1 SMEs

SMEs may be defined as the enterprises which employ less than 250 employees (Mittal, Khan, 2018). Generally SME's data collection consists of manual tracking like reports, journals, tickets, or any other protocol papers (Wank, Adolph, 2016). Additionally SME's have relatively low financial resources, IT developers, processes development, and research driven decision making (Mittal, Khan, 2018).

2.4.2 Diversity of Equipment Age & Manufacturer

One of the largest challenges to achieving connectivity in industrial settings is integrating widely differentiated industrial equipment. Many of the differences stem from manufacturers sourcing their equipment from different companies and different time periods. Through the advancement in time, data sharing technology has changed along with perspectives on the importance of data sharing features. Additionally, manufacturers have different capabilities, product strategies, and equipment interface preferences. One studied indicated that in 2014 the average age of industrial equipment in the US was beyond 10 years (Hagerty, 2014). The heterogeneity of legacy devices and machines created a multitude of communication structures (Jirkovský, Obitko, 2017). For instance, more than 25 industrial ethernet protocols are specified and offered to the market based on vendor-specific implementations. (Givehchi, Landsdorf, 2017). Given that industrial ethernet is just one of many communication protocols used in industry, protocols that span the physical and digital components of information communication, significant complexity is introduced. The resulting landscape has become complex and difficult for manufactures to implement connectivity.

2.4.3 Diversity of Technical Methods Used in Equipment Connectivity

The task of connecting industrial equipment to a central hub is complicated by the use of a vast amount of different means to establishing connectivity. As discussed above, differing industrial equipment support differing communication structures for data transfer. To complicate matters further, each approach to connectivity consists of several interdependent components.

At the most basic level, a piece of equipment must have a physical interface. Some devices were never designed with an interface and require external sensors such as RFID tags, voltage monitor, thermocouple, and cameras to name a few. Other devices were designed to communicate but use one of any number of physical forms of interaction. USB, Ethernet, RS232, RS422, DH-485, ATM, FDDI, B8zS, V.35, RJ45, Wifi, Radio, Microwave, and Bluetooth are some but by no means all of the physical means that devices employ to establish communication.

Once a physical connection is made, an assortment of protocols are employed to recognize, handle and deliver data. Today, the connectivity is mostly done using legacy protocols and technologies designed for shop floor communications (Givehchi, Landsdorf, 2017). ASCII, AS-I, BSAP, CIP, ControlNet, DeviceNet, DirectNet, EFINS, HART, HTTP, Interbus, JPEG, Modbus, NetBios, OPC, OSGP, REST, WebSocket, Zigbee, and Z-wave are a few of the protocols used in industrial connectivity to support the data models and interfaces of different manufacturers. Additionally, protocols have differing functionality addressing different needs of network connectivity and some are more encompassing than others.

Industrial Equipment that transfers data via proprietary protocols often provide greater constraints on protocols used to establish industrial connectivity. Fortunately, often these proprietary protocols are used in a closed environment that provides access to the information through a Graphical User Interface (GUI) application. Advantages of the proprietary protocol systems are that many of them are designed specifically for their intended use. For example, National Instruments Proprietary Wireless Sensor Network Protocol offers features such as device coordination, reliability through mesh networking topologies, and the functionality to create user defined profiles that allow for customization and flexibility within the protocol (Instruments, 2017).

To illustrate that protocols have different functionality and some encompass greater functionality than others, two different combinations of protocols are explained:

1) Ethernet is a protocol that expands the physical or first layer and data link or second layer. Often internet protocol or IP is used in combination with ethernet at the network layer or third layer. TCP, which services the transport or fourth layer, is then often used in combination with IP. TCP can then communicate with NetBIOS, a common session layer or fifth layer protocol. NetBIOS can then translate the data to HTTP for a presentation or sixth layer protocol and an application or seventh layer protocol (Rouse, 2018).
2) ControlNet has developed a protocol solution that spans layers one through four and employs common industrial protocol (CIP) which covers layers 5 through 7. (ControlNet, 2016)

These two combinations show the difference in complexity from one connectivity solution to another. This difference is illustrated in Figure 2 below. Because these solutions are so often dependent on legacy equipment and preexisting connectivity implementations, the overarching task of industrial connectivity can prove difficult.

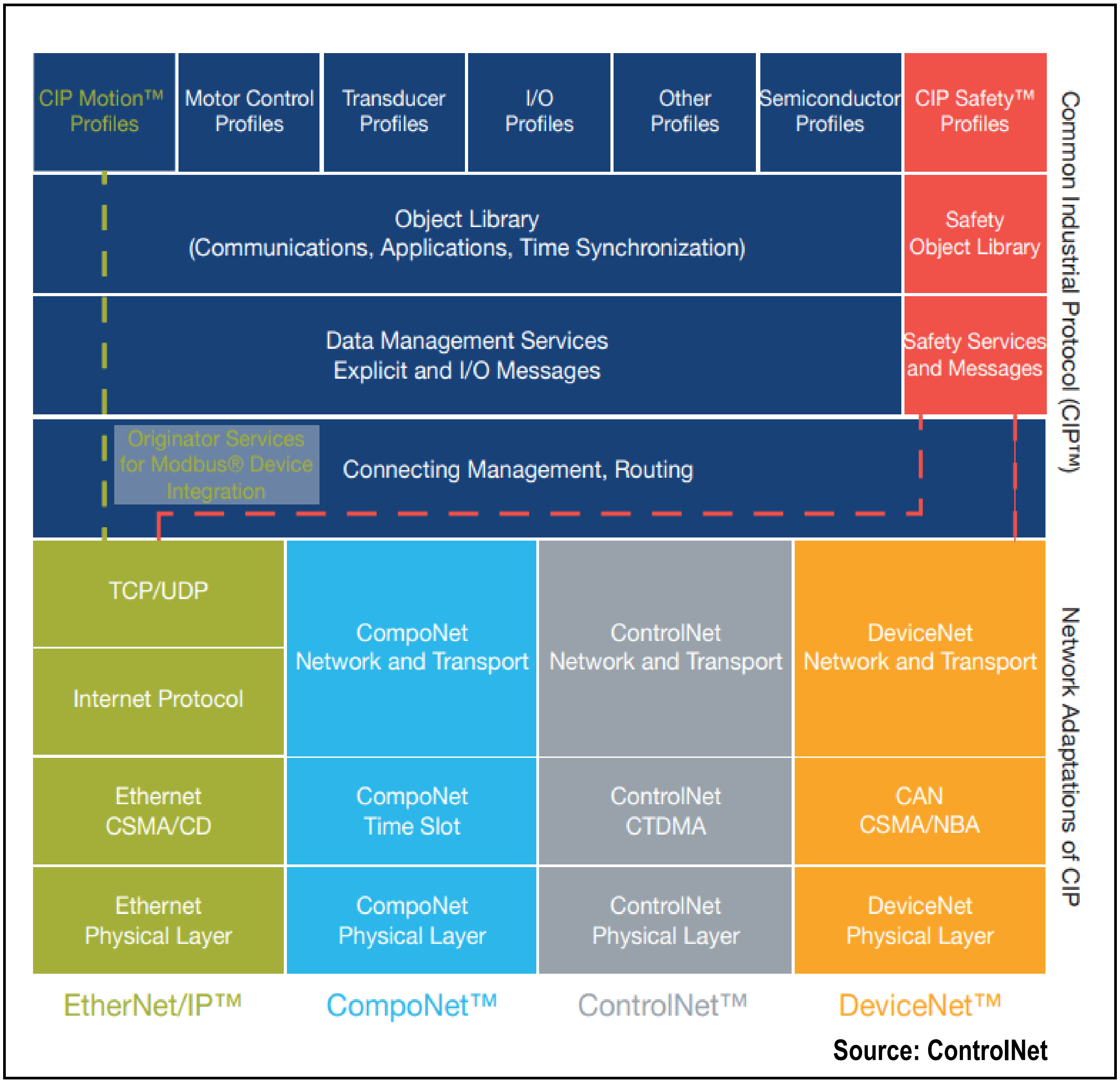

Figure 2: Protocols and Scope of Functionality

2.4.4 Diversity of Equipment Management Software

Beyond establishing a single entity for collecting data from industrial equipment lies a challenge to integrate higher level shop floor applications (Givehchi, Landsdorf, 2017). For instance, Manufacturing Execution Systems (MES), Enterprise resource planning (ERP), and supervisory control and data acquisition (SCADA) software are three of a number of different applications that depend on receiving live data from the shop floor. A significant loss of functionality and interactions with compounding elements of the plant occur when the data is not complete and up to date (Panetto and Molina, 2008). These systems often deal with essential tasks such as process control, quality assurance, and material requisitioning, so their relationship to industrial connectivity is considerable and should be considered in designing underlying connectivity solutions.

IIPs and other solutions exist that that aim to simplify the diffusion of data across applications in addition to simplifying the task of establishing connectivity.

2.4.5 Software Applications for Connectivity

<u>API</u>

Application Programming Interface (API) is a term used to describe a set of subroutine definitions, protocols, and tools for building applications. APIs can be thought of as serving a similar purpose as electrical wall sockets; that is, they are standardized with predictable patterns of access and service conforming to certain specifications (Berlind, 2015). All connectivity applications use one or more APIs. REST API is an especially common form of communication for custom applications.

<u>Applications Using REST API</u>

Representational state transfer (REST) API is a method of establishing communication between computer systems by means of the internet (Booth, 2004). REST API is known as a simple and common form of communication that leverages commonly accepted W3c/IETF standards. One such standard is HTTP which is used by all major operating systems and programming languages (Pautasso, Zimmermann, 2008). This leads to Rest API being versatile in connecting devices and consequentially reducing the barrier to establishing interoperability. A large drawback of REST API is its need to constantly re-establish a connection. The re-establishment of a connection between a device and server can have a significant negative impact on communication with high session establishment frequency.

REST API is relatively simple to implement for IT professionals. They are best suited for device connectivity where pre-built applications are not available and low implementation costs are preferred to greater performance optimization.

Custom Applications

A custom application can be built using different APIs to connect devices to IIoT platforms. This is often done with the aid of software development kits (SDK). SDKs are development tools created for a purpose like creating applications for IIoT platforms. Often, they provide access to other APIs like web sockets which can help optimize communication. (GS, 2015, PubNub, 2015). When implementing connectivity, custom applications typically require more labor hours than alternatives. The key benefit of this approach is the ability to use different APIs including REST API to meet desired requirements.

Pre-Built Applications

Pre-Built applications often come from the IIP developer or communications connectivity companies like Kepware. Often these are solutions specific to device and platform like an IIP and are prepared so that minimal configuration is needed to establish connectivity. Many of these applications, like custom applications, use other APIs for increased performance.

2.4.6 Hardware Solution to Smart Manufacturing (Rip and Replace)

One approach to creating industrial connectivity, is to purchase new equipment that has been designed for seamless industrial connectivity. This approach is referred to as rip and replace. Modern equipment is largely designed for greater connectivity and data sharing but comes with a large price tag.

The general opinion of industry experts is to not replace equipment on the basis of establishing industrial connectivity. The alternative to rip-and-replace is referred to as surround-and-extend. These other solutions, such as those considered in this paper, do not involve replacing industrial equipment and allow for connectivity at a much lower price tag. (Horth, 2017, Paul, 2018)

3 METHODOLOGY

3.1 Analysis of Alternatives Means to Establishing Industrial Connectivity

The health of a manufacturing business depends on its ability to push the boundaries of productivity in an increasingly competitive environment. The next leap in process improvement is found in Smart Manufacturing which enables industrial facilities to employ the most recent technology. While extensive Smart Manufacturing research has been published, little has been done to understand how it applies to SMEs. More specifically, studies have not focused on the technical complexity of alternative means of establishing industrial connectivity for SMEs. A study focusing on the technical complexity of connectivity approaches, contrasting solutions with and without the aid of an IIP, can provide essential information to industrial SME decision makers to begin implementing industrial connectivity. This study will provide much needed guidance for those looking to understand how to go about improving connectivity in industrial facilities.

The type of devices selected for connectivity in this study were designed to collectively incorporate a means for establishing connectivity for almost any piece of industrial equipment. Three distinct devices that each represent one of the three device types are used as a basis for analyzing connectivity solutions. Six connections, two connections for each device, were developed and used to compare IIP and non-IIP implementations. One form of connectivity employed the aid of an IIP for connecting to an IIP hosted server. The other used non-IIP means for connecting to a private server.

The results of this research relied upon the assumption that the author's ability to establish connectivity by different means would be an acceptable representation of the general ability of industrial SME IT professionals. This study focuses on comparing connectivity solutions that meet minimum data receiving and storing functionality or base-line connectivity. Because functionality beyond base-line connectivity is outside the scope of this project, the performance and security benefits of each of the connections will not be addressed.

In this study, the non-IIP connections are ad hoc solutions designed for base-line connections. These solutions require a greater degree of in-house maintenance and developer expertise to expand functionality beyond basic data transmission to other smart manufacturing applications. IIPs have been designed with many applications in mind and often support a simpler transition to these more advanced smart manufacturing applications. The analysis of the methods of connectivity provides compelling data to suggest the complexity tradeoffs of establishing base-line connectivity with and without IIPs.

3.2 Selecting Representative Devices for Connectivity Study

The three devices chosen to achieve broad connectivity are an Allen Bradley MicroLogix 1400 PLC, a Raspberry Pi 1 Model B+, and a LORD MicroStrain WSDA -Base-101 LXRS wireless sensor network.

The Allen Bradley PLC, a common industrial control system, is representative of the industrial control system device category. As discussed in the review of literature, PLCs are a common and perhaps the most prevalent form of industrial control system. Industry analysts like

Frost and Sullivan have called PLCs essential to factory automation. An Allen Bradley PLC was selected due to availability and prevalence in the PLC market with 22% market share (Statista, 2017).

The Raspberry Pi is a small inexpensive computer. In this study, it is a representative of computers that directly connect to industrial equipment for data acquisition purposes. Raspberry Pis are designed for practical projects and although not generally considered industrial grade in reliability, have little variation from other computers for the purposes of this study.

The WSN is a wireless network designed with end devices or nodes that connect directly with sensors. WSNs are used as a prevalent solution for gathering data from equipment beyond what the original equipment design allows for. This is done by using the WSN to transmit data from sensors that are added to equipment. Wireless solutions like WSN are also valuable means of transmitting data without modifying IT wiring infrastructure like running additional Ethernet.

As all protocols were designed to communicate with some form of computer, equipment that is designed to transmit data, no matter the protocol used, can communicate with either an industrial control system or another computer. For equipment that was not designed to transfer data, WSNs provide a solution for gathering data via augmented sensory devices. Together these three device types offer the ability to establish industrial connectivity for industrial SMEs.

3.3 Selecting Approach for Connectivity Solutions

Designing a means of connectivity between a device and a server can be complicated as there are many potential protocols, applications and other tools that can be used. Further, developing a connectivity solution requires a significant amount of effort. For these reasons, only six connectivity solutions were developed for IIP and non-IIP comparison. Two additional connections were developed for analysis of the progress of ThingWorx 7 to ThingWorx 8.2.

The connection solutions developed were intended to be the most practical and most likely approach that would be taken by an SME in establishing base-line connectivity. This allows for the complexity comparisons to compare solutions that best represent the realities preceeding the decision to use or not use an IIP for base-line connectivity.

3.3.1 Selection of Approach for IIP Connections

This study uses Parametric Technology Corporation's (PTC) ThingWorx 8.2, one of the leading IIPs, as a proxy for other IIPs in understanding general IIP value. ThingWorx was ranked the top IIP by Gartner for its ability to execute on market requirements and its vision of current and future market needs (Goodness, Friedman, 2018). Because of PTC's strong accomplishments, it is a good representative for IIPs and their ability to reduce complexity in establishing connectivity.

The approach used in communicating with the IIP server varies based on the resources and recommendations provided by ThingWorx. Some of the approaches are more complicated than necessary for base-line connectivity; however, since they are the recommended approaches by the IIP, SMEs would likely to implement the suggested approach. This means that some of the IIP solutions could be simpler, but the selected approach is the most likely to be implemented by SMEs.

IIPs likely select their solutions using additional criteria such as potential for a variety of smart manufacturing applications. These criteria go beyond the scope of this study and will not be addressed. When the IIP did not provide a recommended solution, RESTful messaging was employed to connect to the IIP's server.

3.3.2 Selection of Approach for Non-IIP Connections

Non-IIP connections are ad hoc solutions developed specifically for the device being connected. The three non-IIP connections use RESTful messaging as the final step in transmitting data to a private server. RESTful messaging was selected because of the straight-forward nature of the technology. Additionally, RESTful messaging is a very common solution employed by developers and consequentially has a strong community with various resources for facilitating RESTful messaging. RESTful messaging has various shortcoming but does accomplish base-line connectivity as needed for this study.

Python was the language selected to create applications that would gather data from sources dependent on the equipment and run the REST protocol for communication with a private server. Python is a common language used in IOT connectivity and has many libraries that allow it to interface with distinct protocols. Additionally, like RESTful messaging, python is very common and has a strong community to provide guidance in its uses.

One of the largest differences between IIP and non-IIP connectivity is the availability of a server. In addition to the process of sending information to a server, non-IIP connections require that a server be created and running. For this study, the server consisted of a PC located on the same local network as the devices that ran a REST API that received RESTful messages. The server was created for basic data reception and storage functiality.

3.3.3 Approach for Analyzing IPP's Change Over Time

In order to evaluate the progress of an IIP over time, two IIP connectivity solutions were made for both the PLC and the Raspberry Pi. One of the solutions for each the PLC and the Raspberry Pi used ThingWorx 7 which was the active version in 2017. The second set of solutions were developed using ThingWorx 8.2 which was the most current version at the time of this study.

The information gathered from this comparison provides insight on both the potential for IIP improvement over time and the potential for IIP solutions to be less effective.

3.4 Data Analysis Method

For this analysis on the ability of IIPs to reduce the complexity of establishing industrial connectivity, metrics were selected as objective measures of complexity. Because the task of establishing connectivity has so many possibilities for implementation, the metrics help identify which key elements have the greatest impact in task complexity as well as provide a means of measuring overall complexity. This simplification, while not a perfect representation of reality, allows for objective evaluations to be made on the complexity of implementing industrial connectivity with and without the aid of IIPs.

3.4.1 Number of Steps

This metric is primarily designed to measure the length of a connectivity solution. The number of steps include only those steps necessary to implement the connection and do not account for any of the work leading up to development the solution. Each step was broken down into minimal directives requiring only common knowledge to understand. Some of the simplest steps consist of selecting the appropriate button on a graphical user interface. Some of the more complicated steps included inserting appropriate text into a software file or running commands on a command line interface.

3.4.2 Number of Software Products

In each connectivity solution, at least one software product was needed. In this context, a software product refers to a separately obtained file with software code intended for use as part of or as an application. Each additional software product is another component of a greater connectivity solution that needs to be integrated in a specific manner to achieve functionality. Additionally, each software product introduces dependence on the products themselves. Software products are subject to change by the developing entity which in turn can change the compatibility of the product with interfacing components. For these reasons, each additional software product introduces an increased chance for error and complexity.

3.4.3 Use of Command Line Interface

Command line interface, the command line for a Windows machine or the terminal for a Linux machine, is a means of controlling a computer. These interfaces are especially useful when a Graphical User Interface (GUI) has not been developed for a desired interaction. Many of the forms of connectivity require use of the command line which can add an additional level of difficulty for implementing connectivity especially for those not experienced with the commands or style of the command line interface.

Additionally, when an application run through command line interface does not function as expected, the debugging process is often harder than with GUI applications. Identifying the source of the problem and understanding the problem become more difficult without command line interface experience.

3.4.4 Number of Code Files Created

Writing code is comparable to speaking a language. It is simpler for those who are experienced and unwieldy for newcomers. For an experienced developer, the hardest part would be becoming familiar with the libraries needed to interface with the protocols of the connectivity solution. Because developing a file from beginning to end is distinctively complex, the number of code files to be created will be measured.

3.4.5 Number of Code Files Modified

Making modifications to code can be an intimidating task. In order to customize code, a text editor is used to access code and then the code is adjusted accordingly with the appropriate syntax to be compatible with the coding language interpreter. Further, components of code are organized in a specific way and a non-experienced individual can easily introduce errors without knowing. For example, in Python, line indentation is very particular. When tabbed white space is replaced by white space created by single spaces the code is broken and vice versa. It is possible that code deemed “need to be created” could be found online and modified, however, for this study the means used to acquire the code whether modified or created will determine the category.

3.4.6 Number of Application Configurations

Software applications often need to be configured to support the desired use in a connectivity solution. Because software configurations depend on the intended use of the software, understanding how to go about making the desired changes can be a challenge. Configuration can be as simple as enabling a setting in the control panel. Conversely, it can include instructing an API service on how to transfer data between two different servers requiring a succession of specific instructions.

3.4.7 Lack of Community Support for Device

An additional metric that will not be considered on its own but only be used as a factor in overall difficulty of implementation is a figure representing community support. A device community is a significant resource in establishing industrial connectivity. The community is helpful when designing the connectivity solution, when encountering unforeseen errors, or otherwise implementing a connectivity solution. The IIP community will be considered for IIP solutions. In assigning a value to the community support figure, the points range spans from no points for a strong community and up to 15 points for a weak community.

3.4.8 Estimated Difficulty

This measure will be discussed in the qualitative section and not mentioned in the quantitative section. Each of the quantitative measures described above help account for the difficulty of a project. To calculate the overall difficulty measure, the complexity measures will be weighted, summed up, and added to the community support figure. Points are weighted as follows: steps are worth one point each, each software product is worth two points, each application configuration is worth three points, use of command line interface is worth ten points, each file of code created is worth fifty points, and each file of code modified is worth ten points.

A step is the most granular and least complex aspect measured per unit. For this reason, steps are worth the lowest whole value, one point. Software products introduce complexity with additional need for component integration into the solution and familiarity with district interfaces. In the experience of the author, the burden of managing an additional software product was small but more significant than performing one additional step. As a result, software products are each worth two point or twice as much as a single step. As mentioned above, command line interface is not as intuitive as GUI alternatives. To account for unfamiliarity, each connectivity solution that requires command line interface was given ten points, the equivalent of ten steps, to account for the interface adjustment and other potential issues. Application configurations requires becoming familiar with the application and making the necessary modification for the application to function properly. Navigating through an application is usually easier than learning to use command line interface, for this reason, application configurations are assigned three points or approximately a third of the assigned points for use of command line interface. Creation and modification of code involve additional skills and reasoning. The average number of lines of code per file created in this study is ninety-three. To account for the application of coding skills and accompanying reasoning of structuring code, each code file created is assigned fifty point or the equivalent of fifty steps. The average number of code modifications is five. To account for the application of coding skills and familiarization with existing code, each code file modified is given ten points or the equivalent of ten steps. One additional note is that these points stack. For example, there are steps to obtain software products, execute command line interface tasks, and incorporate code and both the steps and the other metrics are assigned points in accordance to the structure outlined.

When the measure is 75 or under it is considered easy difficulty. Easy difficulty indicates that this task relatively low amount of work and expertise for industrial connectivity implementations. Easy tasks are more manageable especially for those lacking key IT skill sets. Measures between 75 and 120 are considered medium difficulty. These tasks generally require greater customization than the easy tasks and use of specialized skills, which in turn creates a significant gap in the efficiency between skilled and non-skilled implementors. The solutions that have 120 or more are considered hard difficulty. Tasks with hard difficulty generally are lengthy and require even more customizations than medium tasks. At this point significant additional complexity is attributed to the sheer number of customized and non-customized components needing to be integrated into a synchronized solution. Hard tasks are likely to be overwhelming and impractical for implementors lacking key IT skill sets.

3.4.9 Estimated Labor Hours

This measure will be discussed in the qualitative section and not mentioned in the quantitative section. This estimation represents the time needed to undertake the tasks involved in implanting the solution including creating. The score used to determine difficulty will also be used to determine estimated labor hours for each connectivity solution. Generally, expected labor time for easy tasks are three hours, for medium tasks six hours, and for hard tasks upwards of nine hours.

The time taken to establish the connectivity solutions of this study are in conformity with the time estimations and difficulty ordering of tasks (see appendix B). These estimations are further corroborated by the estimations of ThingWorx, the IIP provider. The IIP provider recently began providing time estimations of connectivity solutions they have designed. The estimations follow the same ordering from shortest to longest and match within an order of magnitude (PTC, 2019).

These estimations are primarily provided for the convenience of manufacturing managers with less technical experience and can help communicate insight into the resources expected for each type of connectivity solution.

3.4.10 Guiding Questions for Qualitative Findings

The guiding questions formed the structure that allowed for consistency in gathering hard to measure information on the connectivity solutions. Two of the questions are positioned to understand how the solutions were developed. These questions are “how difficult was it to design the solution?” and “How much complexity was introduced due to the IIP’s selection of the type of connectivity solution?”. The former question applies mostly to the non-IIP solution and the latter applies to IIP solutions exclusively. The last six were simply to understand the complexity of some of the aspects measured but not completely explained by the number of occurrences. These questions are: “Did the management of the number of software products introduce additional complexities?”, “Were the software products easy to integrate into the solution?”, “To what extent were command line interfaces used?”, “How complex was the code files created?”, “How complicated was the modification of the given code files?”, “How complex was the configuration of custom applications?”.

4 DISSCUSSION AND RESULTS

4.1 Introduction to Findings

The guiding question of this research is, "to what extent do IIPs reduce complexity of establishing industrial connectivity for SMEs?" In order to analyze the level of complexity of the options available for SMEs, the findings will be discussed in three contexts. First, a quantitative approach will be used to compare IIP and non-IIP complexity. Second, a qualitative perspective will be used to evaluate the complexity of the connectivity solutions. Lastly, the final observations will focus on how the IIP solutions for the PLC and Raspberry Pi have changed over time and the implications for future expectations.

For the quantitative approach, the metrics mentioned in methodology will be the most constant and objective measure of complexity in this study. First the study will evaluate trends that apply to IIPs and non-IIP connections. Although the sample size of the data is small, figures like averages and ranges will be indicative. Next, a device level analysis is done comparing the IIP and non-IIP connection of each device. Since the IIP solutions differed significantly from one device to the next, this separation led to insight on which types of solutions reduce complexity the most.

Qualitative information was driven by a few guiding questions that allow for consistency in gathering hard to measure information. At the end of each summary a recommendation on the level of practicality is given to indicate the complexity for establishing baseline connectivity.

The last observations will be made to understand the improvement of IIPs over time. This will be done by comparing equivalent solutions from ThingWorx 7 and ThingWorx 8.2. These observations can provide a basis for expectations of future improvements and capabilities of IIPs to reduce the complexity of implementing industrial connectivity.

4.2 Quantitative Findings

4.2.1 Non-IIP Server and Impact on Results

To begin this analysis, it is important to note that IIP solutions are connecting to an IIP server which is provided by the IIP. Non-IIP solutions require the implementation of their own server. While separate to the process of establishing connectivity, creating a server is a reality that SMEs face and will be required to do if establishing individual connectivity without a server provider. For this reason, the complexity of establishing the server will be shown with the same metrics used for measuring complexity of the individual connections, however, this additional complexity will not be factored into connection level analysis between IIP and non-IIP solutions. This complexity will be considered in the summary as a one-time task that would be required for a non-IIP industrial connectivity approach.

The process of creating a RESTful server for this study has 35 steps, uses 3 software products, requires command line interface, requires creating a custom Python script, and requires a custom firewall configuration. According to the estimation process described in the methodology section, this process has a medium difficulty rating and an estimated six hour implementation time.

Table 1: RESTful Server Implementation Metric

RESTful Server	
Number of Steps	35
Number of Software Products	3
Command Line Interface	1
Number of Code Files Created	1
Number of Code Files Modified	0
Number of Application Configurations	1
Lack of Community Prominence of Device	5
Complexity Measure	109
Estimated Difficulty*	M
Estimated Labor Hours*	6 hours

*For an individual with beginning networking skills

4.2.2 Complexity Analysis from Comparing IIP and Non-IIP Data

The mean number of steps for IIP solutions is 61 while the mean number of steps for non-IIP solutions is 51.3. The range in number of steps for IIP solutions is 24 with a minimum of 48 and a maximum of 72. The range in number of steps for non-IIP solutions is 28 with a minimum of 33 and a maximum of 61. There is little difference in the statistics of the number of steps between IIP and non-IIP solutions. This measure does not show that IIPs reduce complexity of establishing connectivity.

Although the values above seem fairly consistent across IIP and non-IIP solutions, the size of the range relative to the average indicates a large degree of deviation in the individual device solutions.

The mean number of software products is 4.3 for IIP connections and 5.6 for non-IIP connections. For all connectivity solutions, the number of software products needed were between five and six with the notable exception of the IIP PLC connectivity solution. This indicates that most connections, including both with or without the aid of an IIP, are dependent on a variety of independent software products. This highlights the heterogenous nature of these solutions and difficulty of connectivity solution design even when using an IIP.

The use of command line interface is consistent across IIP and non-IIP solutions. All the connections employed some use of the command line interface, again, with the exception being the IIP PLC connection. This indicates that implementing connectivity for industrial settings will most likely require using command line interface. The extent of command line interface use is discussed in greater detail in the qualitative findings. Without a device level perspective, the IIP and non-IIP solutions are comparable on this measure of complexity meaning there is little indication for differentiated complexity.

Each of the non-IIP solutions requires creating a custom code file while only one of the IIP solutions required creating a custom code file. All the IIP solutions that had a recommended connectivity solution did not require this level of code development. The trend indicates that when a connectivity recommendation is offered by an IIP provider, the level of resources available will not require ground-up code development.

Although ground-up code development was not needed in IIP connectivity recommendations, code modifications were. The only connectivity solution that required code modification was the Raspberry Pi IIP connection. This connectivity solution required modifying three code files. This significant increase from the number of code files used in the other solutions were dependent on the connectivity approach chosen by the IIP.

The mean number of custom application configurations were 1.3 for both IIP and non-IIP solutions, with the range being one to two for both IIP and non-IIP solutions. There was no difference in this measure between IIP and non-IIP solutions. This measure suggests that the complexity of IIP and non-IIP are equal.

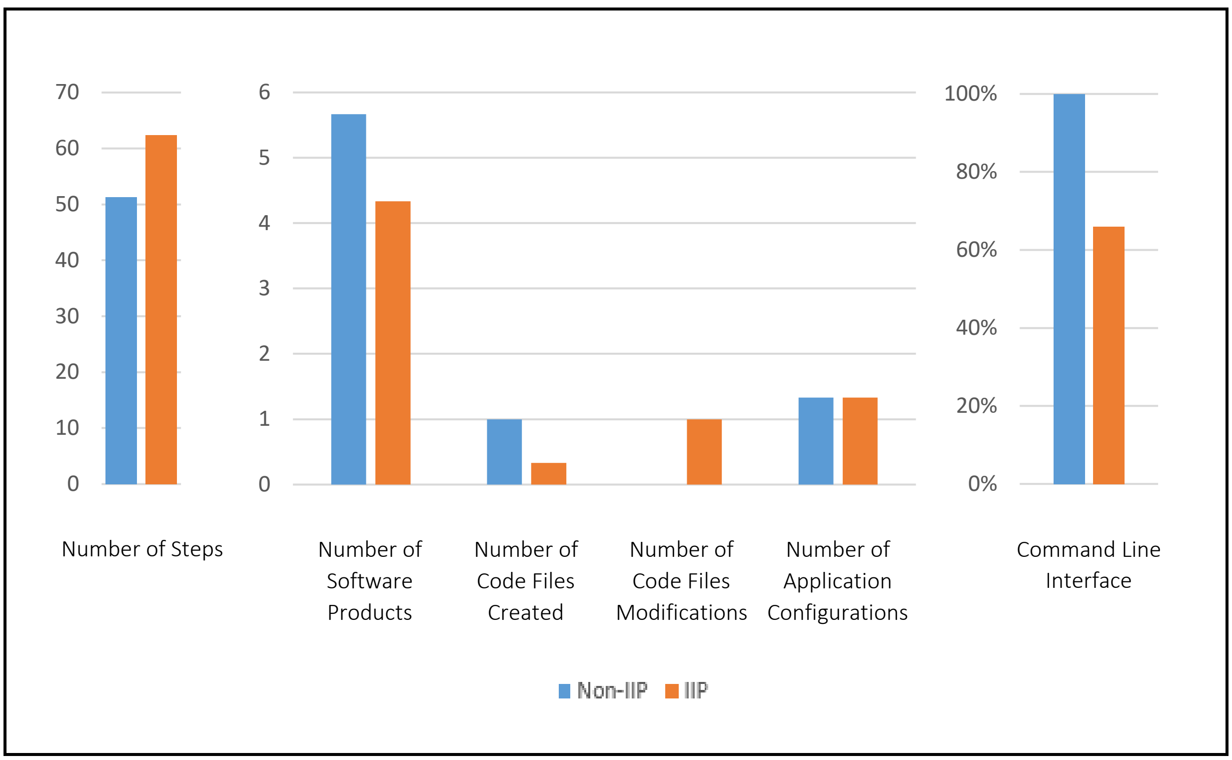

Figure 3: Non-IIP and IIP Solutions' Measures

4.2.3 Device Level IIP and Non-IIP Findings

These findings will be based on the percent change of the data using non-IIP solutions for the starting values and IIP solutions for the ending values.

The IIP PLC connection had about 20% less steps, 67% fewer software products, had no need for command line interface when the non-IIP solution does, does not have a need to create code, and had the same number of custom application configurations.

The significant reduction in almost every measure is due to the type of solution offered by the IIP. The PLC IIP solution uses a specialized driver and server interface that simplifies the task of establishing connectivity. The non-IIP solution does not benefit from the same type of end-to-end solution and requires an open source Python library to interface between an open platform communication (OPC) sever and the private server which is further discussed in the qualitative section.

Table 2: PLC IIP Solution Compared to PLC Non-IIP Solution

PLC Non IIP to IIP Relative Difference	
Number of Steps	-21%
Number of Software Products	-67%
Command Line Interface	Reduced to none
Number of Code Files Created	Reduced to zero
Number of Code Files Modified	No Change
Custom Application Configuration	No Change

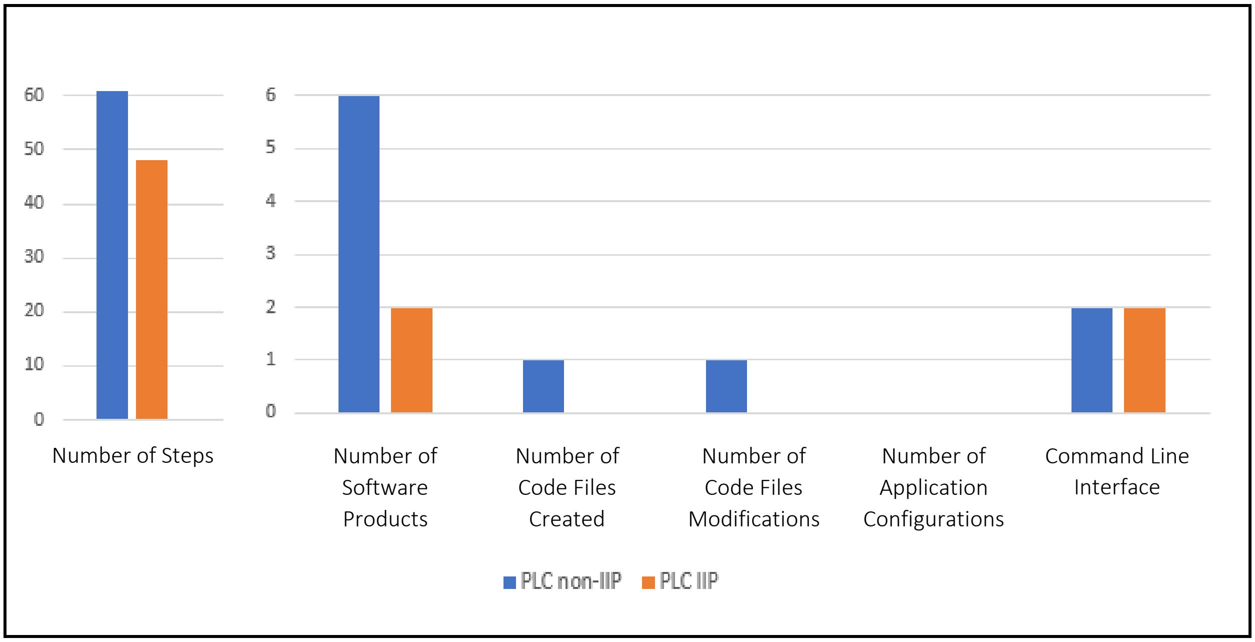

Figure 4: PLC Non-IIP and IIP Solutions' Measures

The IIP Raspberry Pi connection more than doubled the number of steps needed to establish connectivity at 118%. Further, the IIP Raspberry Pi connection used the same number of software products, continued to use a command line interface, requires three code modifications when the alternative requires none, and had the same number of custom application configurations. The non-IIP solution does however require developing custom code.

Table 3: Raspberry Pi IIP Solution Compared to Raspberry Pi Non-IIP Solution

Raspi Non IIP to IIP Relative Difference	
Number of Steps	130%
Number of Software Products	No Change
Command Line Interface	No Change
Number of Code Files Created	Reduced to zero
Number of Code Files Modified	Increased from 0 to 3
Custom Application Configuration	No Change

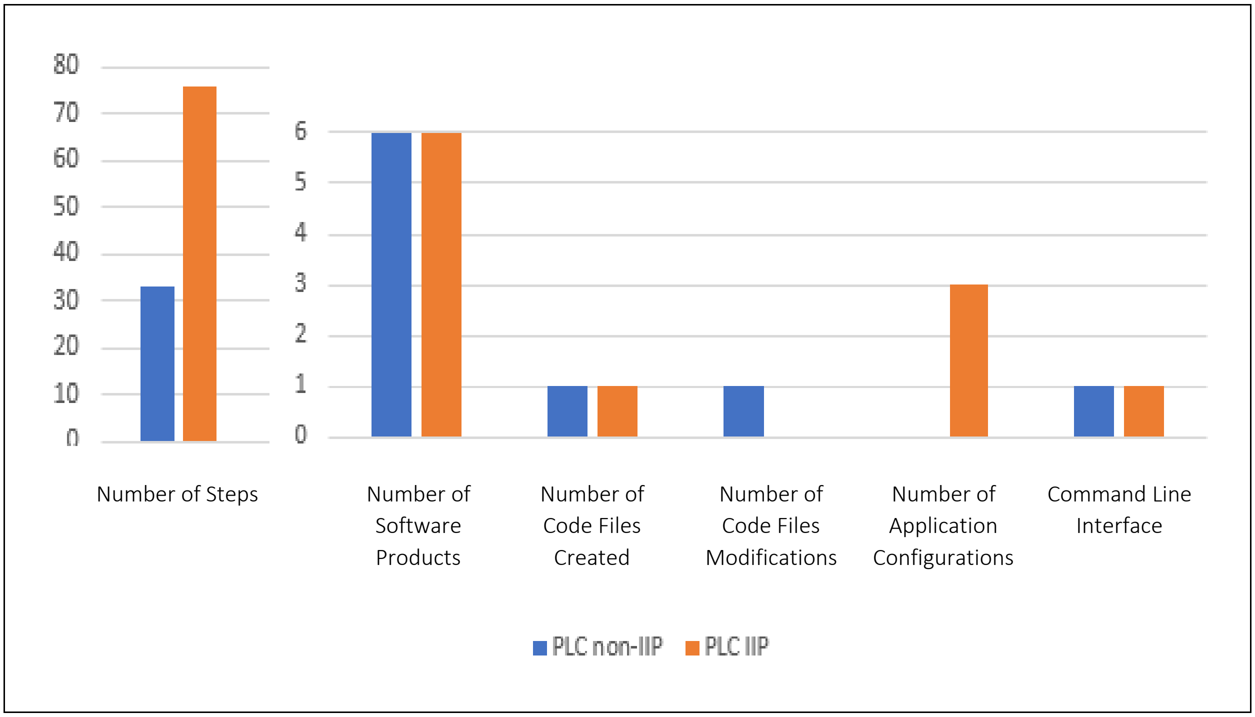

Figure 5: Raspi Non-IIP and IIP Solutions' Measures

In this case, if it were not for the metric on creating code it would seem that the more complex solution was the IIP solution. The Raspberry Pi does not have a specialized driver to establish connectivity and requires a significant amount of work to implement a connection than the PLC IIP solution. The IIP solution uses an approach that is at this time fundamentally more difficult to establish than the approach used for the non-IIP solution. Another significant difference is that the IIP solution still provides the code for the solution and only requires modification where the non-IIP solution requires building code from scratch. Consequentially, it is difficult to clearly determine the more complex implementation of connectivity and the lack of easily cross comparable metrics makes this harder. For these reasons further discussion on a conclusion continues in the qualitative section.

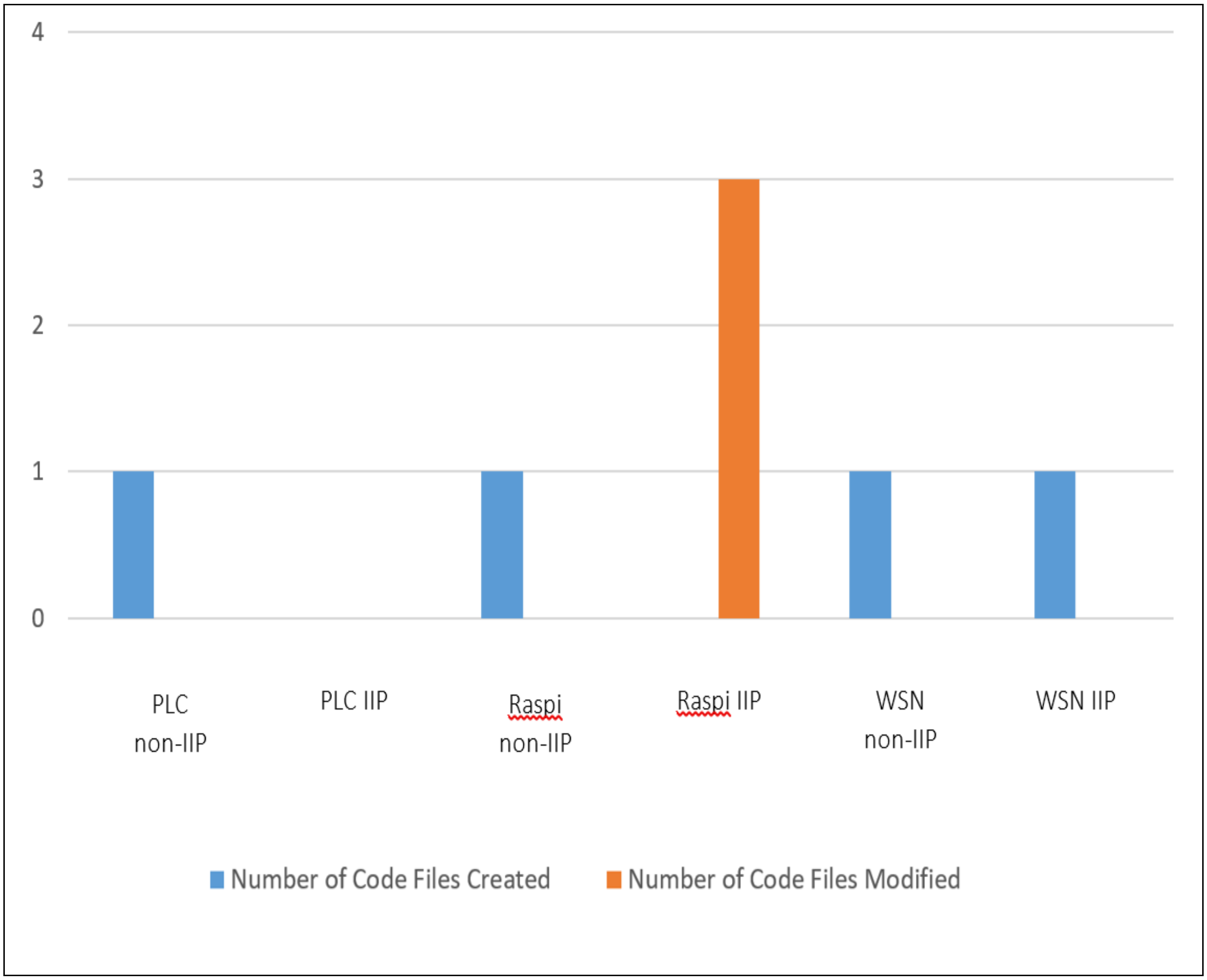

Figure 6: Solutions' Exposure to Code

The IIP WSN connection had a 5% increase in the number of steps needed to establish connectivity, while requiring the same resources on the other measures. The IIP provider had no recommended solution for connecting the WSN so the same ad hoc solution was used for connecting to the IIP server and non-IIP server. The difference from these measures was negligible indicating equal complexity.

Table 4: WSN IIP Solution Compared to WSN Non-IIP Solution

WSN Non IIP to IIP Relative Difference	
Number of Steps	5%
Number of Software Products	No Change
Command Line Interface	No Change
Number of Code Files Created	No Change
Number of Code Files Modified	No Change
Custom Application Configuration	No Change

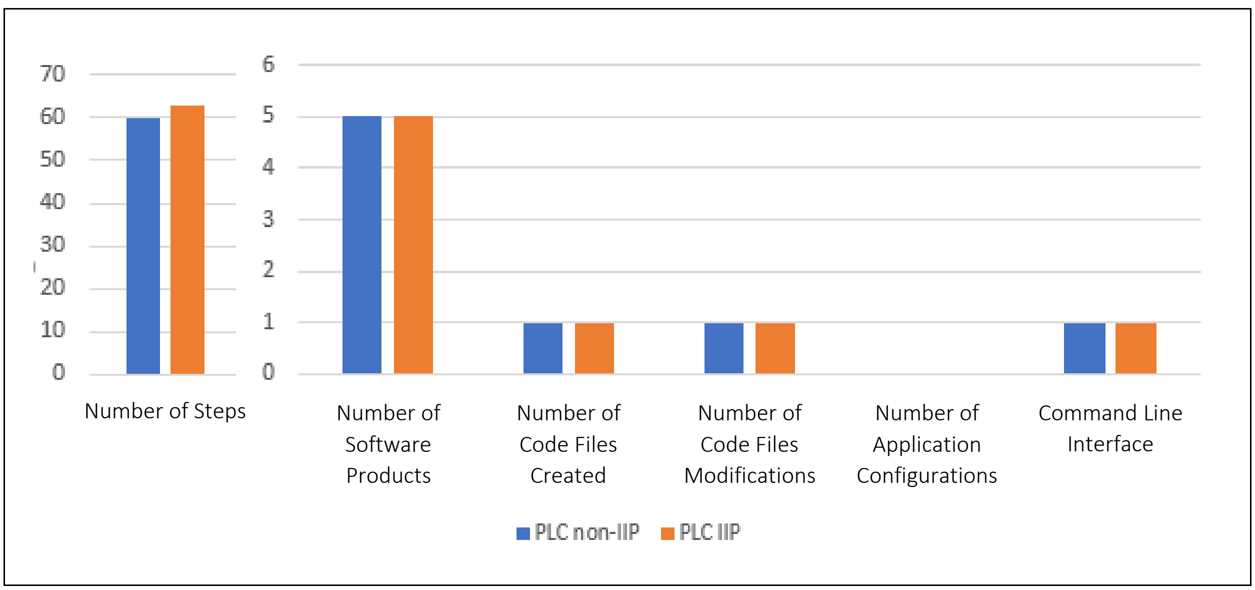

Figure 7: WSN Non-IIP and IIP Solutions' Measures

4.2.4 Quantitative Findings Summary

IIPs seem to reduce the complexity of base-line connectivity only when an end-to-end software solution has been developed. In the absence of this form of solution, the IIP solution can be more complex to implement. When an IIP solution is not available, an ad hoc solution can be implemented to connect to the IIP server, in which case, an IIP does not offer complexity reduction.

4.3 Qualitative Findings

4.3.1 IIP PLC Connection

The IIP PLC connection was broadly speaking the most seamless solution of the study to implement. Although the IIP selected a solution that uses a websocket protocol, which is typically more difficult to implement than RESTful messaging, ThingWorx IIP built a driver and server interface to deliver an end-to-end connectivity solution.

This solution allowed for most of the connectivity implementation to be done through GUI interfaces. Additionally, the software automated a large portion of the process mostly just requiring information on specifics of the PLC and Server Instance.

The IIP PLC connection used two software products (see figure 7) but it was possible to have only used “ThingWorx Industrial Connectivity 8”. In this connection, RS Logix was used to program and run the PLC and this could be achieved using the IIP software which would reduce the total number of software products to one.

1. ThingWorx Industrial Connectivity 8
2. RSLogix

Figure 8: PLC IIP Solution Software Products

According to the estimation process described in the methodology section, this process has a complexity measure of 58, an easy difficulty rating and an estimated 3 hour implementation time. The lack of community support measure was zero because the Rockwell PLCs have a strong online community and because the IIP provides an additional community and support service.

4.3.2 Non-IIP PLC Connection

The non-IIP PLC connection was a connection with a few challenges. The first step was to understand how to access information from the PLC which was complicated by the number of products advertised in this category. For this study, it was found that Rockwell's RSLinx product includes a driver to communicate with the PLC and was capable of relaying the information with an OPC server. The next step was to understand how to interface with the OPC server to transmit data out over a network.

After some research, an OPC communication Python library was identified and would allow for data to be gathered and then transmitted through RESTful messaging on a single Python application. Creating the Python script requires understanding how to write code. For an experienced developer, the hardest part would be becoming familiar with OPC and REST libraries. For a non-experienced individual, the project could quickly become cumbersome requiring learning the basics of Python and application development.

This solution requires 5 software products (see figure 8). Since most software products were directly related to building the Python application and have a simple installation process, the software products did not contribute much to the complexity of implementing the solution. This solution required a command line interface just to launch the Python application, a generally simple task. Additionally, this solution used a version of Python that required manual configuration of an environment variable in Windows, which can be a slightly tricky task.

1. RSLinx
2. Python 2.7
3. Notepad++
4. Python Requests Package
5. OpenOPC Python Package

Figure 9: PLC Non-IIP Solution Software Products

Creating an application from scratch is not a trivial task and the task alone increases the complexity significantly. For this evaluation we will assume moderate code development experience and add complexity of equivalent of 50 steps to this solution.

The resulting complexity measure was 142 resulting in a hard classification and an estimated 9 hour implementation time. The lack of community support measure was a 5 instead of zero like the IIP PLC solution because this solution lacked the IIP support service.

If the code were already created and available through a walk-through like the one found in the appendix, the code would only require minimal modifications and the overall difficulty would be considerably less. However, the overall hard classification seems fitting for the type of tasks involved in this connectivity solution.

4.3.3 PLC Solutions Complexity Comparison

As discussed, the IIP PLC solution is simple on all the accounts discussed. The non-IIP PLC solution has some more complex elements with the most notable being the need to design the solution and the need to create custom code. Of the three IIP and non-IIP device level comparisons, this is the most polarizing, suggesting large benefits in complexity reduction when using an IIP.

Table 5: PLC Solutions Overall Complexity

	IIP	Non-IIP
Aggregate Complexity	58	142
Difficulty Level	Easy	Hard
Time Estimate	3 Hours	9 Hours

4.3.4 IIP Raspberry Pi Connection

The IIP recommendation for the Raspberry Pi was to use their code that employs websocket protocol. This solution did not have a specialized driver to establish connectivity for the Raspberry Pi connection, making establishing connectivity more difficult.

This solution uses six software products that require appreciable effort making the management of the products a task worth noting (see figure 9). For example, using a disk imager to flash the operating system of the Raspberry Pi onto a SD card was a fairly in-depth process. The software product that introduced the most complexity was the EMS Zip file. This file has three code files that require opening and modifying in order to enable functionality. Additionally, this solution requires using command line interface to verify proper performance of the components of the EMS Zip file. If things proceed according to plan, the verification task is simple, but when there is a problem even experienced IT professionals can struggle to fix the issue.

1. Rasbian OS
2. Win32diskimager
3. EMS Zip file
4. Adafruit Python Library
5. Python 3
6. Build Essentials for Python

Figure 10: Raspberry Pi IIP Solution Software Products

The resulting complexity measure was 141 resulting in a hard classification and an estimated 9 hour implementation time. Raspberry Pi's have a strong online community although not tailored towards industrial connectivity or the Adafruit sensor used. Additionally, as a recommended solution, IIP support service is also available. For these reasons, the lack of community support measure was a five.

The websocket protocol used in this solution has benefits not found in RESTful Protocols and generally would be considered a more robust solution but is currently more difficult to implement than RESTful messaging.

4.3.5 Non-IIP Raspberry Pi Connection

The non-IIP Raspberry Pi solution initially follows much of the same process as the IIP solution for the Raspberry Pi, including the installation of the Raspbian operating system. Once the Raspberry Pi is setup and running, the remainder of the connection consists of creating a custom Python script.

Creating the Python script requires understanding how to write code. For an experienced developer, the hardest part would be becoming familiar with Adafruit and REST libraries. For a non-experienced individual, the project could quickly become cumbersome requiring learning the basics of Python and application development.

This solution requires 6 software products (see figure 10). Like the non-IIP PLC solution, most software products were directly related to building the Python application and have a simple installation process. As a result, the software products did not contribute much to the complexity of implementing the solution. This solution also just required a command line interface just to launch the Python application.

1. Rasbian OS
2. Win32diskimager
3. Python 3
4. Build-tools for python installation
5. Adafruit Python API Library
6. Python Requests Package

Figure 11: Raspberry Pi Non-IIP Solution Software Products

For the same reasons as discussed in the non-IIP PLC connection, creating an application from scratch is a complex task. In this solution, the custom Python code must receive information from the sensor and transmit data to the RESTful server.

The resulting complexity measure was 118 resulting in a medium classification and an estimated 6 hour implementation time. The lack of community support measure was a ten, having a similar community to the IIP Raspberry Pi solution but lacking the IIP support service.

4.3.6 Raspberry Pi Solutions Complexity Comparison

Both solutions had the same need to setup the Raspberry Pi but afterwards the approach differed significantly. The IIP solution implemented websocket protocol requiring a large number of steps and number of code files modified, leading to a high complexity rating. The non-IIP solution received only a medium complexity rating because despite having the need for custom code, it had a small measure on the other complexity metrics. In this case the complexity rating is deceiving as the IIP approach had less than a twenty percent higher complexity score which is within the reasonable error expectations from generalizing creating custom code. In fact, the uncommon Adafruit API can alone make the custom code creation a very difficult process, which issue is mostly solved for when using the IIP solution.

These two solutions, although different, have comparable complexities dependent on the knowledge and skills of the implementation entity.

Table 5: Raspi Solutions Overall Complexity

	IIP	Non-IIP
Aggregate Complexity	141	118
Difficulty Level	Hard	Medium
Time Estimate	9 Hours	6 Hours

4.3.7 IIP WSN Connection

The IIP provider had no recommended solution for connecting the WSN so an ad hoc solution was developed and applied. Like mentioned in the Non-IIP PLC section, designing a solution has a few challenges. The largest challenge as mentioned in the non-IIP solutions is the need to create custom code.

The WSN has software and a Python library that allows for a Python script to receive information from the WSN. In this solution the Python script that communicated with the WSN also used RESTful messaging to transmit data. Like the Non-IIP PLC & Raspberry Pi solutions, the 5 software products were reasonable to implement, and the use of command line interface was not complex.

1. SensorConnect
2. Python 2.7
3. Notepad++
4. MSCL Package
5. Python Requests Package

Figure 12: WSN IIP Solution Software Products

Table 6: WSN Solutions Overall Complexity

	IIP	Non-IIP
Aggregate Complexity	151	148
Difficulty Level	Hard	Hard
Time Estimate	9 Hours	9 Hours

Like the Non-IIP PLC solution, this solution used a version of Python that required manual configuration of an environment variable in Windows, a task with some complexity.

The resulting complexity measure was 151 resulting in a hard classification and an estimated 9 hour implementation time. The lack of community support measure was a fifteen, due to the WSN having limited online resources supporting use of its API.

4.3.8 Non-IIP WSN Connection

This solution was identical to the IIP WSN Connection except for not needing a security key (appkey) to communicate with the intended server.

For this reason, please refer to the IIP WSN Connection section for this information.

4.3.9 WSN Solutions Complexity Comparison

These solutions were identical which means that without a recommended solution from an IIP to connect a specific device, the least complex connectivity solutions will likely be the same. In this circumstance, the complexity is equal, except for the fact that the IIP can provide the end-point server for industrial connectivity.

4.3.10 Qualitative Findings Summary

The Qualitative findings support the quantitative findings in that IIPs seem to reduce the complexity of base-line connectivity only when an end-to-end software solution has been developed. The most difficult aspect of the non-IIP solutions is designing the solution itself and creating the code needed to execute the implementation. Both aspects are hard to quantify and rely heavily on the entity implementing the solution. The need for custom code is the single largest factor in making non-IIP RESTful solutions as difficult as the IIP websocket solutions. In the absence of the end-to-end connectivity solution, the IIP can even recommend a solution, like the Raspberry Pi’s using websocket protocol, that is potentially more complex to implement.

Implementers have the option to ignore the IIP recommendation, but it is likely that if it is available, it will be used.

When using an ad hoc solution, an IIP provides the benefit of running the server that will host the data. As discussed above, establishing a server for basic connectivity is a one-time task but would need considerable time investment for any additional functionality beyond receiving and storing data.

4.4 IIP's Current State and Recent Changes Observed

Comparing the IIP PLC connections of the current ThingWorx 8.2 to ThingWorx 7, the current solution has 51% less steps, 71% fewer number of software products, had no need for command line interface when the non-IIP solution does, reduced the number of code files created from one to none, and has 33% less custom app configurations.

ThingWorx 7, did not have an end-to-end solution that supported the MicroLogix 1400 PLC. For this reason, the IIP is used to establish a message queuing telemetry transport (MQTT) topic with data from the PLC which is accessible through a MQTT server. Other than that, an ad hoc solution was made to transfer the data from the MQTT topic to the IIP server.

Table 7: ThingWorx Version Comparison of PLC IIP Solutions

PLC IIP V7 to V8.2 Relative Difference	
Number of Steps	-51%
Number of Software Products	-71%
Command Line Interface	Reduced to None
Number of Code Files Created	Reduced to Zero
Number of Code Files Modified	No Change
Custom Application Configuration	-33%

Comparing the IIP Raspberry Pi connection from 8.2 to 2017, the current solution has 4% more steps, 14% fewer software products, continues to use command line interface, requires three times the number of code file modifications, and has 67% less custom app configurations.

The changes between the two solutions does not come from the type of solution implemented as they both use websocket protocol. Rather, the ThingWorx 7 version favored using more third-party software products which were already incompatible with the solution in ThingWorx 7. To implement the solution, many highly complex issues had to be solved including renaming and modifying obscure files. The new approach uses less third-party software but requires modifications of more files which are all obtained as part of the EMS Zip file and updated by the IIP.

Table 8: ThingWorx Version Comparison of Raspberry Pi IIP Solutions

Raspi IIP V7 to V8.2 Relative Difference	
Number of Steps	10%
Number of Software Products	-14%
Command Line Interface	No Change
Number of Code Files Created	No Change
Number of Code Files Modified	300%
Custom Application Configuration	-67%

The improvement in the solution for the PLC shows the growing availability of end-to-end solutions. At the time, the MicroLogix 1400 had not been addressed with the same kind of end-to-end solution likely due to the MicroLogix being an older less common PLC.

The ThingWorx 7 solution for the Raspberry Pi shows that some solutions offered by the IIP may be broken and require special effort to fix and implement. The current 8.2 solution shows their dedication to offering working solutions. The two tasks would have similar complexities had the ThingWorx 7 solution functioned without needing fixes.

The solutions improved considerably over time. The errors in the solutions were fixed, end-to-end connectivity solutions were developed and there was less dependence on third party software.

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 The Impact of the Research

The purpose of this study was to gage the ability to reduce the complexity for SMEs in implementing industrial connectivity. An extensive literature review was conducted in order to provide context to why industrial connectivity is necessary and to illustrate why there are difficulties for SMEs in implementing industrial connectivity. Connectivity solutions were developed that are representative of connectivity needs for industrial SME. The solutions were recorded and documented for analysis and complexity data was presented in both qualitative and quantitative formats.

The study showed that IIPs are valuable in reducing complexity under certain circumstances. When the IIP has developed a special driver and server interface to establish end-to-end connectivity, IIPs are very effective in reducing complexity. When an end-to-end solution is not available but another IIP solution is, creating ad hoc solutions can have similar complexities to implementing the recommended IIP solution. In scenarios where IIPs do not provide solutions, there is little complexity reduction benefit in using IIP's to establish base-line connectivity.

When a customized driver and server interface is built to provide an end-to-end connectivity solution, like in the case of an IIP PLC solution, the solution is simple to implement. The solution requires very little effort to configure and setup. No code modification, code creation or even manual command line interface is required to implement the connectivity solution. According to this study, this type of solution clearly leads to the least complex connectivity implementation.

When a similar end-to-end solution is not available by the IIP, like in the case of the IIP Raspberry Pi solution, the resulting solution is significantly more demanding. Not only is the solution more complex than the IIP counterpart, the complexity can rival that of an ad hoc solution. This is seen in the qualitative comparison of the IPP and non-IIP Raspberry Pi solutions. IIPs likely selected their connectivity solutions using differing criteria than the Non-IIP solutions in this study. The design of Non-IIP solutions looked to be practical and be comprised of software products with a strong developer community. By contrast, the IIP solution for the PLC and the Raspberry Pi had a greater focus on increased data throughput and security.

When no connectivity solution is offered by the IIP, the solutions are for these purposes identical. Without the IIP's guidance, a solution needs to be designed, created and implemented. Perhaps the simplest solutions to create use RESTful protocol and the connectivity solution is the same whether or not an IIP is available. To understand the complexity difference of IIP and non-IIP connections when an IIP solution is not offered, the WSN connections are a good benchmark. In comparing these solutions, the only measured difference was 3 additional steps for the IIP solution which compromised identifying the security key for obtaining server access.

In general, the biggest complexity factors faced by non-IIP solutions were design of the solutions and the creation code files. IIPs will continue to expand their connectivity solutions. For some connections this may include streamlined end-to-end solutions that will greatly reduce complexity of establishing connectivity. However, connections like these will not be available for every potential industrial equipment or even a large fraction. These types of solutions are likely to be created for the most pressing applications like control systems. This study has also shown that some solutions may not work and could require additional work to fix issues that arise.

The connectivity solutions developed for this study suggest that industrial connectivity is within reach for any industrial SME. IIPs provide very simple connectivity solutions for select and likely the most popular industrial equipment. Beyond that, base-line connectivity can be accomplished using free methods that are well understood and supported by a healthy online community. Although ad-hoc approaches can be accomplished, they do require a significant amount of labor when the solutions require connectivity design, creation and implementation which can be reduced if a step-by-step guide or relevant example code is available. The results of this study are of value to decision makers in industrial SMEs who are considering implementing industrial connectivity and provides insight into approaches for reaching base-line industrial connectivity. These results likewise open new opportunities for further research which could lead to understanding additional benefits of IIPs beyond reducing the complexity of establishing base-line connectivity.

5.2 Suggestions for Further Research in Industrial Connectivity for SMEs

This study shows a comparison to establishing base-line connectivity with and without the aid of IIPs, and yet there are many topics that can benefit from further research. A study could be done to learn of the features offered by IIPs that are difficult to obtain without.

Another study could focus on of the most pressing needs of industrial SMEs and related features and correlate those data to implications on the type of industrial connectivity needed.

Some industry analysts have begun to evaluate desired applications of smart manufacturing for SMEs which could be used to establish a basis for discovering the features necessary in connectivity solutions to support those needs.

The ad-hoc connectivity measures in this study relied heavily on RESTful messaging protocols, yet there is a wide array of protocols that may be selected to create ad-hoc connectivity solutions. Studies revealing the complexity of implementing solutions using different protocols and outlining their respective benefits could provide a great deal of insight for decision makers.

It is possible that more information could be gained to understand the future solutions to be offered by IIPs. A comprehensive study on this topic could provide insight into SME needs of smart manufacturing and associated smart manufacturing features. Further the study could show which of the features is most likely to be addressed in the near future by IIPs.

5.3 Suggestions for SMEs Looking to Implement Industrial Connectivity

If SMEs are only interested in connecting popular industrial control systems to a server, this study suggests that the complexity of theses connections will be greatly reduced with an IIP. If the SME is only interested in establishing base-line connectivity and has equipment to connect other than popular industrial control systems, ad-hoc solutions may be a more appropriate solution depending on the circumstances. For situations that do not fit into either of these scenarios, more research and understanding of motivations is needed for a recommendation.

For a SME with multiple brands or versions of PLCs, greater research into the IIP is warranted. If the IIP provides end-to-end connectivity for the PLCs in question, the IIP can greatly simplify establishing connectivity. Otherwise, the non-IIP and IIP solutions can have similar complexities. In the case where at least one ad-hoc solution is pursued, the same type of OPC server can likely be used to connect to all the PLCs and once a Python application is developed to interface with that OPC server, minimal changes are needed to gather and transmit additional data.

SMEs with one PLC and several other pieces of industrial equipment that has not connected to the PLC, need to do additional research. They need to know if the IIP has developed specialized end-to-end solutions for their equipment if not do they offer any type of solution walk-through. If there are solutions available, an understanding of the skill level available is needed to know if the IIP will reduce complexity for base-line connectivity. If solutions are not available through an IIP provider, the use of an IIP will most likely not reduce complexity.

Due to the scope of this research, little was done to evaluate the benefits of IIP beyond reducing the complexity of base-line connectivity. Although the setup time for RESTful messaging for IIP and Non-IIP methods may be similar, IIPs have other potential advantages. Some likely advantages are, a hosted server available on the via the world wide web, security precautions, implementations designed to scale, and continuously adopting the latest technology. Additionally, IIPs are designed to support features such as helping with factory wide data organization and visualization, receiving data and sending commands to equipment, enabling cross-business unit applications, enabling cross-sources analytics (supply chain evolution), running data science algorithms, building augmented reality factory floor interface, providing a data historian, and more. SMEs would do well to understand those benefits as they pertain to their respective business.

REFERENCES

Alaybeyi, S. (2016). Pragmatic strategies to improve industrial iot security.

Berlind, D. (2015) What is an API, Exactly.

Booth, D. H., Hugo; McCabe, Francis; Newcomer, Eric; Champion, Michael; Ferris, Chris; Orchard, David (2004) Web Services Architecture.

Cecílio, J. and P. Furtado (2014). Wireless sensor networks: concepts and components. Wireless Sensors in Heterogeneous Networked Systems, Springer**:** 5-25.

ControlNet (2016) ControlNet, CIP on CTDMA Technology. Technology Overview Series

Galloway, B. (2013). "Introduction to industrial control networks." **15**(2): 860-880.

Gartner (2019) IT Glossary.

GE (2016) The Future of Electricity is Digital. GE Power, Digitial Solutions

Givehchi, O. (2017). "Interoperability for industrial cyber-physical systems: An approach for legacy systems." **13**(6): 3370-3378.

Goodness, E. (2018) Magic Quadrant for Industrial IoT Platforms.

GS (2015) Why the REST don't use WebSockets.

Hagerty, J. R. (2014) U.S. Manufacturing Is Rolling on Aged Wheels. Wall Street Journal

Helu, M. (2016). Enabling smart manufacturing technologies for decision-making support. ASME 2016 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers.

Horth, C. (2017) IIoT: Let's Sound a Loud R.I.P. for Rip-and-Replace.

Instruments, N. (2017) Wireless Sensor Network Frequently Asked Questions.

Jabbar, S. (2018). "A REST-based industrial web of things' framework for smart warehousing." **74**(9): 4419-4433.

Janarthanan, A. (2015) Make in india initiative for smart manufacturing with automation. Market Insight

Jirkovský, V. (2017). "Understanding data heterogeneity in the context of cyber-physical systems integration." **13**(2): 660-667.

Lydon, B. (2016). "Industry 4.0: Intelligent and flexible production." 12-17.

Manyika, J. C., Michael; Bisson, Peter; Woetzel, Jonathan; Dobbs, Richard; Bughin, Jacques; Aharon, Dan (2015). The internet of things: Mapping the value beyond the hype. McKinsey Global Institute.

McAvoy, J. (2018) Recognition of PTC's ThingWorx Industrial Innovation Platform Continues with Praise from Industry Experts.

Mittal, S. (2018). "A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs)." **49**: 194-214.

Mui, C. (2016) Thinking big about industrial internet of things.

Panetto, H. and A. Molina (2008). "Enterprise integration and interoperability in manufacturing systems: Trends and issues." **59**(7): 641-646.

Patel, m. S., Jason; Thomas, Christopher (2017) What's new with the internet of things?

Paul, F. (2018) How to add IoT functions to legacy equipment.

Pautasso, C. (2008). Restful web services vs. big'web services: making the right architectural decision. Proceedings of the 17th international conference on World Wide Web, ACM.

PTC (2019). "Connect Your Data to ThingWorx." from https://developer.thingworx.com/en/platform/connect.

PubNub (2015) REST vs. WebSockets.

Radziwon, A. (2014). "The smart factory: exploring adaptive and flexible manufacturing solutions." **69**: 1184-1190.

Rouse, M. (2018) Industrial Internet of Things.

Rouse, M. (2018) Presentation layer.

Statista (2017) Global PLC market share as of 2017, by manufacturer.

Sullivan, F. a. (2016). Navigating Digital Transformation in Manufacturing.

TMR (2016). Smart Manufacturing Market-Global Industry Analysis, Size, Share, Growth, Trends and Forecast 2016-2024, Factory Automation**:** 129.

Wank, A. (2016). "Using a learning factory approach to transfer Industrie 4.0 approaches to small-and medium-sized enterprises." **54**: 89-94.

Zhao, C. W. (2015). "Exploring iot application using raspberry pi." **2**(1): 27-34.

APPENDICES

APPENDIX A. DATA CHART

Solution and RESTful Server Metrics

	PLC non-IIP	PLC IIP	Raspi non-IIP	Raspi IIP	WSN non-IIP	WSN IIP	PLC w/ IIP old	Raspi** w/ IIP ol	RESTful Serve
Number of Steps	61	48	33	76	60	63	97	69	35
Number of Software Products	5	2	6	6	5	5	7	7	3
Text Interface	1	0	1	1	1	1	1	1	1
Number of Code Files Created	1	0	1	0	1	1	1	0	1
Number of Code Files Modified	0	0	0	3	0	0	0	1	0
Number of Application Configurat	2	2	1	1	1	1	3	3	1
Weighted Score of metrics	137	58	108	131	133	136	180	112	104
Lack of Community Prominence o	5	0	10	10	15	15	5	10	5
Complexity Measure	142	58	118	141	148	151	185	122	109
Estimated Difficulty*	H	E	M	H	H	H	H	H	M
Estimated Labor Hours*	9 hours	3 hours	6 hours	9 hours	9 hours	9 hours	9 hours	14 hours	6 hour

APPENDIX B. TIME TAKEN TO IMPLEMENT CONNECTIVITY SOLUTIONS

Original Raspi IIP	24 hours
Original PLC IIP	20 hours
Raspi IIP	4 hours
PLC IIP	1 hour
WSN IIP	7 hours
Raspi non-IIP	6 hours
PLC non-IIP	8 hours
WSN non-IIP	7 hours