

Tying Together Solutions for Digital Manufacturing: Assessment of Connectivity Technologies & Approaches

Gregory Hawkrige, Marco Perez Hernandez, Lavindra de Silva, German Terrazas, Yedige Tlegenov, Duncan McFarlane, Alan Thorne
Institute for Manufacturing, University of Cambridge,
Cambridge, UK
{gth24, mep53, ldp25, gt401, yt361, dm114, ajt28}@cam.ac.uk

Abstract—This paper concerns the development of low-cost solutions to address challenges in digital manufacturing (DM). Service Oriented Architectures (SOAs) are a promising approach for addressing the requirements of a low-cost DM architecture. Interaction between services in a SOA is facilitated by a connectivity technology, i.e., a framework for interoperable data exchange between heterogeneous participants. We review a variety of connectivity technologies according to their suitability for use in an SME manufacturer’s production environment, and we assess how they have been integrated into past architectures. We then provide insights into an incremental and modular architecture for manufacturing SMEs.

Keywords—digital manufacturing, low-cost solutions, service oriented architectures, connectivity technologies

I. INTRODUCTION

A. Low-Cost Digital Manufacturing

Digital Manufacturing (DM) in its broadest terms refers to the application of digital information (from multiple sources, formats, and owners) for the enhancement of manufacturing processes, supply chains, products and services. This paper focusses on the development of very low-cost solutions to address aspects of digital manufacturing challenges. By low-cost digital manufacturing in this paper, we refer to the *development of digital solutions to meet specific operational needs and for which the total cost of deployment (purchase, integration, installation and operation) is kept low.*

We specifically focus on the use of low-cost digital solutions by manufacturing SMEs (Small to Medium Enterprises), who not only desire to keep equipment/development/deployment costs low but also require that solutions be simple to deploy and maintain. In particular, we consider opportunities for exploiting off the shelf technologies and openly available software in addressing these joint goals of simplicity and low cost.

B. An Architecture for Low-Cost DM

For a manufacturing SME there are several requirements for a low-cost DM architecture. First, DM solutions should be self-sufficient, and the architecture needs to be incremental and modular so that an SME can start with a single DM solution (e.g. order tracking) and then progressively add further solutions as their digital demands require and capital expenditure allows. Solution self-sufficiency is key since the cost of installing auxiliary infrastructure is a major deterrent. Second, the architecture should be flexible, so that there are no restrictions on the order in which solutions are added to an SME’s manufacturing environment. Third, the architecture should encourage synergy between solutions so that each solution can provide additional features or performance by

interacting with other solutions when they are present in the architecture. This synergistic interaction and data sharing is a key advantage of the DM paradigm. Fourth, the architecture needs to be interoperable with existing legacy equipment found in an SME’s production environment. Finally, the architecture needs to be self-configuring; the ability for the architecture to configure itself reduces expert consultation costs.

C. A Service Oriented Approach to a Low-Cost DM Architecture

Service Oriented Architectures (SOAs) are a promising approach for addressing the requirements of a low-cost DM architecture. A SOA is an architectural framework in which complex systems are built from autonomous, interoperable services [1]. Each service provides device-specific functionality that is exposed through a well-defined interface. Service autonomy is expected to satisfy the low-cost DM architecture’s solution self-sufficiency requirement and should facilitate modularity. Service interoperability is expected to provide flexibility and facilitate integration with legacy systems.

Interaction between services in a SOA is facilitated by a connectivity technology (Fig. 1). A connectivity technology is here understood as a communication framework that enables interoperable data exchange between heterogeneous participants [2]. In this paper, we review a variety of connectivity technologies according to their suitability for use in an SME manufacturer’s production environment, and we assess how they have been integrated into past architectures. We then provide insights into an incremental and modular architecture for manufacturing SMEs.

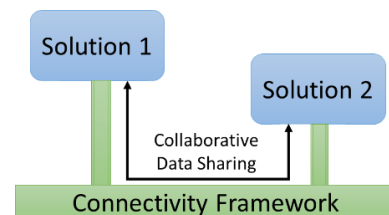


Fig. 1. Role of a connectivity framework in a SOA

D. Paper Outline

This paper is organised as follows. We begin by presenting a set of preliminary requirements for a connectivity technology in the context of a low-cost digital manufacturing SOA. We then review relevant connectivity technologies against these requirements, which is followed by an assessment of the current approaches to integrating such technologies. Finally, we provide insights into a low-cost architecture for manufacturing SMEs and draw conclusions.

II. CONNECTIVITY TECHNOLOGIES FOR LOW-COST DM

A. A Preliminary Set of Requirements

A SOA requires certain core functionalities within a connectivity technology. These functionalities include: *addressing*, each service has a unique address with which it can be referenced; *discovery*, the ability for services to find other required services; *service metadata exchange*, a description of the capabilities offered by a service; and *messaging*, for services to interact with one another [3].

To meet the requirements of a low-cost DM architecture some of these requirements need to be refined and other further requirements need to be added. These requirements are separated into necessary and preferred requirements. The necessary requirements are as follows: *addressing* and *metadata exchange*, unchanged; *discovery*, performed in a decentralised manner (i.e. using multicast mechanisms) for early stages of incremental implementation with the option of installing one or more centralised providers as the architecture grows; *messaging*, synchronous for control and asynchronous for events (ideally including a publish-subscribe mechanism for flexibility); and based on *open standards*, to avoid vendor lock-in and to improve interoperability with legacy systems.

Preferred requirements are not optional from a system perspective; however, they are aspects that can be implemented at an application level within services if the connectivity framework does not support them. The preferred requirement is *security* to provide authentication (validation of message sender and contents), authorisation (permissions required to perform specific actions) and confidentiality (prevent unintended recipients from viewing message contents).

B. Key Connectivity Technologies

We now discuss the key connectivity technologies that meet our connectivity requirements outlined in the previous section. We discuss three ‘full-fledged’ connectivity technologies: Object Linking and Embedding for Process Control Unified Architecture (OPC-UA), Device Profile for Web Services (DPWS), and Universal Plug and Play (UPnP).

OPC-UA [4] is a lightweight client-server communication protocol which originated from a collaboration between some leading automation suppliers. Consequently, OPC-UA software libraries seem to be available for some of the major operating systems, PLC brands, and programming languages, though not always as open source. An OPC-UA server provides services using a standard (object-oriented) interface which cannot be modified by the user. To this end, a service request made by a client can have an arbitrary number of input and output parameters. Nonetheless, OPC-UA does provide a mechanism for clients to discover relevant services offered by servers via ‘service signatures’. Other relevant features include recent support for the publish-subscribe protocol (via OPC-UA PubSub), and secure communication via authentication, authorisation, and encryption. However, OPC-UA cannot be used as a protocol for real-time (deterministic) communication, though some work is being done to improve this, e.g. by making OPC-UA RESTful [4].

Like OPC-UA, DPWS [5] is a client-server communication protocol developed by Microsoft, which is being used in industrial automation as part of the Web Services for Devices (WS4D) initiative. DPWS defines a minimal set of protocol features from web service standards -- such as addressing, discovery, and security -- that must be implemented by (possibly resource constrained) devices for them to be deemed DPWS compliant. Unlike OPC-UA, DPWS allows the specification of user defined service signatures using XML -- and for these services to be discovered by DPWS clients. Low latency communication between devices is currently limited by the verbosity of DPWS’s XML encoding, though there is work being done to improve this using binary XML [6].

UPnP [7] is a client-server communication protocol established by the UPnP Forum, comprising diverse member industries, for the promotion of manufacturer-independent device interconnectivity. Unlike its successor DPWS, which is aimed at industrial devices, UPnP is more suited for small home and business networks that connect appliances, wireless devices, and PCs. However, UPnP may well suit certain industrial networks or parts thereof, such as a subnetwork comprising a PC and some wireless sensors that monitor CNC machines. Unlike DPWS, UPnP is not based exclusively on web-service standards but supports features such as addressing and decentralised service discovery.

C. Other Relevant Connectivity Technologies

In this section we discuss the remaining connectivity technologies in the literature that meet the requirements in Sec. II-A, and summarise these technologies in Table 1. The table also lists some popular features that are not part of our requirements.

TABLE I. SUMMARY OF OPEN SOURCE CONNECTIVITY TECHNOLOGIES (ADAPTED FROM [8])

	Machine data acquisition	Field level data model	Built for industrial use	Publish-subscribe	Security	Custom service definitions	Real time	Open source	System & Language Support
OPC-UA	●	●	●	●	●	●	○	●	●
MTCCon...	●	●	●	●	○	●	○	●	○
DPWS	○	●	●	●	●	●	○	●	○
UPnP	○	○	○	●	○	●	○	○	●
DDS	●	○	●	○	●	○	●	○	○

● Completely ○ To a lesser extent
 ○ Implementable/Specifiable ● No/Not possible

The Data Distribution Service (DDS) [2], maintained by the Object Management Group, is a robust technology for real-time communication between devices. DDS allows user defined quality of service (QoS) parameters such as how much historical data should be stored, and the maximum latency when delivering data. DDS is available as open source under certain conditions (e.g. university licenses), and it has been used in a diverse range of domains including aerospace, defence and manufacturing. DDS implements a ‘publish-subscribe’ interaction pattern for sending and

receiving data, events, and requests between hardware/machines. Machines or components that produce information (i.e., the publishers) create messages related to ‘topics’ such as temperature, pressure and vibration, and publish data to topics. DDS can quickly and deterministically deliver the data to subscribers that have registered with those topics.

MTConnect [8] is a ‘read-only’ protocol built for data acquisition from machines and devices, and the streaming of data via a REST interface from the shop floor up to higher level systems such as databases. To this end, MTConnect ‘servers’ represent devices, and higher level systems use MTConnect ‘clients’ to stream device data at a sample rate that is determined by the application (though not in real-time). Device data is represented in XML, using an extensible schema provided by MTConnect; the schema can be customised by the user to model device specific data, while re-using standard MTConnect-defined data elements. MTConnect does not inherently support security features such as authentication before data retrieval, but such features could be added by implementing them as part of a separate software layer. Finally, while MTConnect does provide client-server features, it does not support full-fledged service orientation, e.g. the ability for two higher level systems to discover each other and communicate.

Besides the open source technologies discussed in Table 1, there are some proprietary ones that may well be affordable to some SMEs. In particular, the National Instruments Shared Variable Engine (SVE) is a software framework that allows variables to be exposed by devices, discovered by other devices, and shared between devices [8]. Like the DDS, SVE allows fast data acquisition from devices, as well as data logging, distribution, routing and data replication; data is shared by publishing them to software components that subscribe to the data. Importantly, experiments show that SVE can meet (hard) real-time requirements [10]; it is therefore a viable platform for monitoring and distributing real-time field-level data. Some of the other relevant proprietary technologies include WinCC [11] by Siemens, the mBS SDK [12] by ProSyst (recently acquired by Bosch), dataFEED [13] by Siemens, and the Integration Bus Manufacturing pack [14] by IBM.

III. CURRENT APPROACHES TO INTEGRATING DM TECHNOLOGIES

We now describe how past work in DM has used some of the connectivity technologies discussed above for the purpose of connecting hardware and software components into an integrated manufacturing environment.

A. Incrementally Migrating Legacy Devices

An important aspect of migrating a legacy manufacturing environment has been to associate the relevant devices such as PLCs, and (non-connectivity) software components such as databases and data analytics tools, to connectivity technologies. As an example, [13] use the Siemens dataFEED hardware module to associate a Siemens PLC with an OPC-UA server. The server is accessed by an OPC-UA client, which publishes the data to an Azure cloud platform for

analytics. On the other hand, [6] directly ‘wrap’ PLCs and robots as OPC-UA services (without the use of extra hardware). The services publish extracted data, which higher level software services subscribe to. Similarly, [15] wrap a Siemens CNC machine using an OPC-UA server in an industrial aerospace drilling application. The CNC data from spindle and motion drives is transmitted at 10 samples per second to an OPC-UA client, which in turn wraps a MATLAB application that is used for assessing hole making quality.

Since 10 samples per second might not be acceptable in applications requiring real-time response, [15] suggest using an industrial Ethernet protocol such as Profibus, in order to get faster sample rates of up to 100 samples per second. This motivates the need for larger wrappers, representing an entire industrial (real-time) subnetwork, such as a network comprising a PLC connected to a group of field devices. The wrapper would correspond to one connectivity technology, representing a service interface for higher level applications.

As noted in the IMC-AESOP project [16], it is not always desirable to digitalise an entire legacy facility, particularly if doing so might be too disruptive or risky. In addition, digitalisation should be carried out incrementally, by starting with the least ‘sensitive’ parts of the facility, e.g. software applications and databases, and then moving on to the more sensitive parts, e.g. those that need to be retrofitted with sensors. While doing the latter, disruption to processes could be mitigated by using non-intrusive sensors for diagnostics and monitoring [17], such as wireless sensors that are based on the EXI and CoAP technologies. EXI (Efficient XML Interchange) and CoAp (Constrained Application Protocol) target resource-constrained devices (e.g. sensors); the former provides a format for data interchange and the latter offers a communication protocol over IP networks using the REST architectural style [18].

B. Incrementally Bridging Connectivity Technologies

Since a manufacturing SME may have a preference for some connectivity technologies over others, e.g. because their factory PLCs only support OPC-UA, or because they are already using MTConnect for data collection, the most appropriate integration strategy appears to be to use any number of suitable connectivity technologies in the manufacturing environment, and to incrementally link them via ‘bridges’. A bridge is a software module -- possibly coupled with a standalone hardware module (e.g. dataFeed [13]) -- that performs protocol (including data and message) translation between two connectivity technologies in order to enable communication.

There are a number of approaches that advocate the use of bridges in the context of SOAs for manufacturing. In particular, [9] stream field device data using bridges to translate from Modbus, Serial, MQTT and other protocols to OPC-UA specific data structures. In the SOCRADES project [19], devices are integrated with ERP systems by using bridges between DPWS and OPC-UA clients and servers. Similarly, [20] propose using bridges between OPC-UA and

DPWS services in order to enable communication between networks that only use one or the other. There are also bridges between OPC-UA and SVE [21], and efforts to create a bridge between OPC-UA and DDS [22]. Perhaps most importantly, bridging between connectivity technologies, as opposed to supporting a single connectivity standard, is also recommended by the Industrial Internet Consortium [2].

IV. A PRELIMINARY ARCHITECTURE FOR LOW-COST DM

Based on the integration strategies discussed above, we suggest the use of an incremental architecture for low-cost DM, based on open source connectivity technologies, low-cost hardware and software components (e.g. cloud based data analytics tools), wrappers for legacy as well as newly added components, and bridges between connectivity technologies. The most suitable set of connectivity technologies for the architecture should be chosen based on the needs of the specific SME's manufacturing environment. Fig. 2 depicts a possible instance of the architecture that we propose, set in the context of the ISA-95 industrial automation hierarchy [23]. The instance comprises some connectivity technologies (within blue circles), some low level industrial communication protocols (within a green circle), and some database technologies (within an orange circle). Each database technology is wrapped within exactly one DPWS server, and each connectivity technology can be added incrementally, for example by first wrapping the CouchDB database within a DPWS server (possibly after first installing the database on a local private network), as this would be the least disruptive step, then wrapping the field device within a DDS service, and then bridging DDS with OPC-UA.

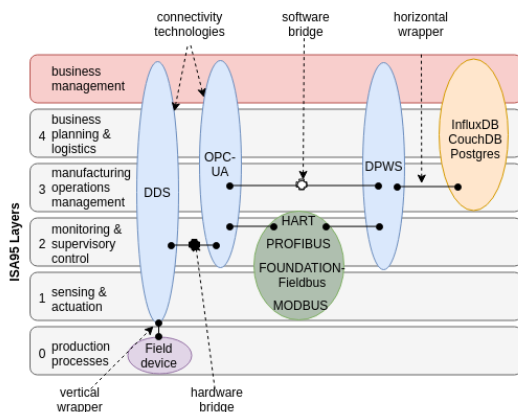


Fig. 2. An instance of our architecture (adapted from [2,16]).

V. CONCLUSIONS

We have presented a preliminary set of mandatory and preferred connectivity requirements for integrating low-cost solutions for manufacturing SMEs. Based on a review of the current state-of-the-art, we provided an assessment of the main connectivity technologies in meeting these requirements. Finally, we reviewed existing strategies to integrating DM technologies, and proposed a preliminary architecture for low-cost DM, that incorporates wrappers and bridges for integrating technologies.

REFERENCES

- [1] Jammes, F. and Smit, H., 2005. Service-oriented paradigms in industrial automation. *IEEE Transactions on industrial informatics*, 1(1), pp.62-70.
- [2] C. T. Joshi Rajive, Didier Paul, Jimenez Jaime, "The Industrial Internet of Things Volume G5: Connectivity Framework," 2017.
- [3] Papazoglou, M.P. and Van Den Heuvel, W.J., 2007. Service oriented architectures: approaches, technologies and research issues. *The VLDB journal*, 16(3), pp.389-415.
- [4] S. Grüner, J. Pfrommer, and F. Palm, "RESTful Industrial Communication with OPC UA," *IEEE Trans. Ind. Informatics*, vol. 12, no. 5, pp. 1832–1841, 2016.
- [5] G. Cândido, A. W. Colombo, J. Barata, and F. Jammes, "Service-oriented infrastructure to support the deployment of evolvable production systems," *IEEE Trans. Ind. Informatics*, vol. 7, no. 4, pp. 759–767, 2011.
- [6] F. Jammes, "Real time device level Service-Oriented Architectures," *Proc. - ISIE 2011 2011 IEEE Int. Symp. Ind. Electron.*, pp. 1722–1726, 2011.
- [7] H. Bohn, A. Bobek, and F. Golatowski, "SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains," *Proc. Int. Conf. Networking, Int. Conf. Syst. Int. Conf. Mob. Commun. Learn. Technol.*, 2006.
- [8] J. Morgan and G. E. O'Donnell, "Enabling a ubiquitous and cloud manufacturing foundation with field-level service-oriented architecture," *Int. J. Comput. Integr. Manuf.*, vol. 30, no. 4–5, pp. 442–458, 2017.
- [9] O. Givehchi, K. Landsdorf, P. Simoens, and A. W. Colombo, "Interoperability for Industrial Cyber-Physical Systems: An Approach for Legacy Systems," *IEEE Trans. Ind. Informatics*, vol. 13, no. 6, pp. 3370–3378, 2017.
- [10] J. Morgan and G. E. O'Donnell, "A Service Oriented Reconfigurable Process Monitoring System – Enabling cyber physical Systems," *J. Mach. Eng.*, vol. 14, no. 2, pp. 116–129, 2014.
- [11] SIMATIC WinCC V7 - System overview, Siemens Aktiengesellschaft, Munich, Germany, 2019. Accessed on: Jun. 10, 2019. [Online]. Available: <https://w3.siemens.com/mcms/human-machine-interface/en/visualization-software/scada/simatic-wincc/pages/default.aspx>
- [12] mBS SDK 8.2: An Overview, Robert Bosch GmbH, Stuttgart, Germany, 2019. Accessed on: Jun. 10, 2019. [Online]. Available: http://documentation.bosch-si.com/iot/SDK/v8.2/SDK_mBS/getting_started/product_overview.html
- [13] H. Haskamp, F. Orth, J. Wermann, and A. W. Colombo, "Implementing an OPC UA interface for legacy PLC-based automation systems using the Azure cloud: An ICPS-architecture with a retrofitted RFID system," *Proc. - 2018 IEEE Ind. Cyber-Physical Syst. ICPS 2018*, pp. 115–121, 2018.
- [14] IBM Integration Bus Manufacturing Pack overview, IBM, New York, United States, 2019. Accessed on: Jun. 10, 2019. [Online]. Available: https://www.ibm.com/support/knowledgecenter/en/SSMKHH_9.0.0/com.ibm.manufacturing.doc/ma00010.htm
- [15] M. Eckstein and I. Mankova, "Monitoring of drilling process for highly stressed aeroengine components," *Procedia CIRP*, vol. 1, no. 1, pp. 587–592, 2012.
- [16] S. Karnouskos et al., *The IMC-AESOP architecture for cloud-based industrial Cyber-physical Systems*, vol. 9783319056. 2014.
- [17] J. Delsing et al., "A migration approach towards a SOA-based next generation process control and monitoring," *IECON Proc. (Industrial Electron. Conf.)*, no. May 2014, pp. 4472–4477, 2011.
- [18] A. P. Castellani, M. Gheda, N. Bui, M. Rossi, and M. Zorzi, "Web Services for the Internet of Things through CoAP and EXI," no. Xml, p. 10783, 2014.
- [19] L. M. S. De Souza, P. Spiess, D. Guinard, M. Köhler, S. Karnouskos, and D. Savio, "SOCRADES: A Web service based shop floor integration infrastructure," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4952 LNCS, pp. 50–67, 2008.
- [20] B. Bony, M. Hamischfeger, and F. Jammes, "Convergence of OPC UA and DPWS with a cross-domain data model," *IEEE Int. Conf. Ind. Informatics*, pp. 187–192, 2011.
- [21] National Instruments, "How LabVIEW Uses I/O Servers", National Instruments Corporation (U.K.) Ltd, Newbury, UK, 2019. Accessed on: Jun., 8, 2019. [Online]. Available: <https://www.ni.com/en-gb/innovations/white-papers/12/how-labview-uses-i-o-servers.html>
- [22] J. Pfrommer, S. Gruner, and F. Palm, "Hybrid OPC UA and DDS: Combining architectural styles for the industrial internet," *IEEE Int. Work. Fact. Commun. Syst. - Proceedings, WFCS*, vol. 2016-June, 2016.
- [23] ANSI, ISA-95.00. 01: Enterprise-control System Integration–Part 1: Models and Terminology. Washington, 2010.