On the Role of Simulation and Simulation Standards in Industry 4.0

Steffen Strassburger
Ilmenau University of Technology
Information Technology in Production and Logistics
Max-Planck-Ring 12
98693 Ilmenau, GERMANY
steffen.strassburger@tu-ilmenau.de

Keywords:
Industry 4.0, Digital Twin, Cyber-Physical Systems, Interoperability, Simulation Standards.

ABSTRACT: *This article introduces the concepts and ideas behind Industry 4.0 and discusses the role of simulation and simulation standards for implementing it. We argue that the success of Industry 4.0 highly depends on the success of interconnected cyber-physical systems (CPS) which can only be implemented with up-front simulation. This up-front simulation and development of CPS is often associated with the term of building the "digital twin" for the respective CPS. Digital twins are typically defined as digital representations which represent the real system and its current state in a digital model. For investigating their dynamic behavior, digital twins must have properties typically associated with simulation models. In this article, we discuss requirements and potential solutions for the successful implementation of digital twins as well as the implications that this has on simulation standards. As an example, digital twins as representations of a CPS will have the need to communicate with other digital twins; hence a modular approach for building federations of digital twins is needed. Beyond that, also a need for standardized communication between the digital twin and the real CPS arises. The article will therefore discuss currently available interoperability standards, like the High Level Architecture (HLA) on the simulation side, and Open Platform Communications (OPC) Unified Architecture (OPC UA) on the control hardware side and how well they match the requirements that Industry 4.0 with its CPSs and digital twins imposes. The article also includes our opinion on the need for the future evolvement of existing standards.*

1. Introduction

Industry 4.0 is a term created in 2011 within the high tech strategy of the German Federal Ministry of Education and Research. It has since received significant international attention, both in academia and industry. While the term certainly has a buzzword character and the "4.0" extension is often inadequately used in conjunction with other words, many of the original concepts behind Industry 4.0 are worthwhile investigating.

The term Industry 4.0 refers to the fourth industrial revolution that is supposed to be taking place in manufacturing industries at the current time. The three prior "industrial revolutions" and their occurrence over time are displayed in Figure 1. These prior revolutions namely were based on

- the introduction of water and steam-powered mechanical manufacturing facilities,
- the introduction of electrically powered mass production based on the division of labor, and
- the use of electronics and IT to automate manufacturing [1].

Industry 4.0 goes beyond the automation introduced in manufacturing in the 1970s/1980s by putting forward the idea of networked cyber physical systems (CPS) as the base for implementing smart factories of the future.

In this article we argue that networked CPS can only be implemented with up-front simulation. This up-front simulation and development of CPS is often referred to as building the "digital twin" for the respective CPS. Digital twins as representations of a CPS will have the need to communicate with other digital twins; hence a modular approach for building federations of digital twins is needed. Beyond that, also a need for standardized communication between the digital twin and the real CPS arises. This requires the deployment of adequate interoperability standards.

The remainder of this paper is structured as follows. Section 2 gives an introduction into the main concepts of Industry 4.0. Section 3 discusses the role of simulation for implementing Industry 4.0. Section 4 reviews currently available interoperability and communication standards, like the High Level Architecture (HLA) on the simulation side and Open

Platform Communications (OPC) Unified Architecture (OPC UA) on the control hardware side and how well they match the requirements that Industry 4.0 with its CPSs and digital twins imposes. The article concludes with our opinion on the need for the future evolvement of existing standards.

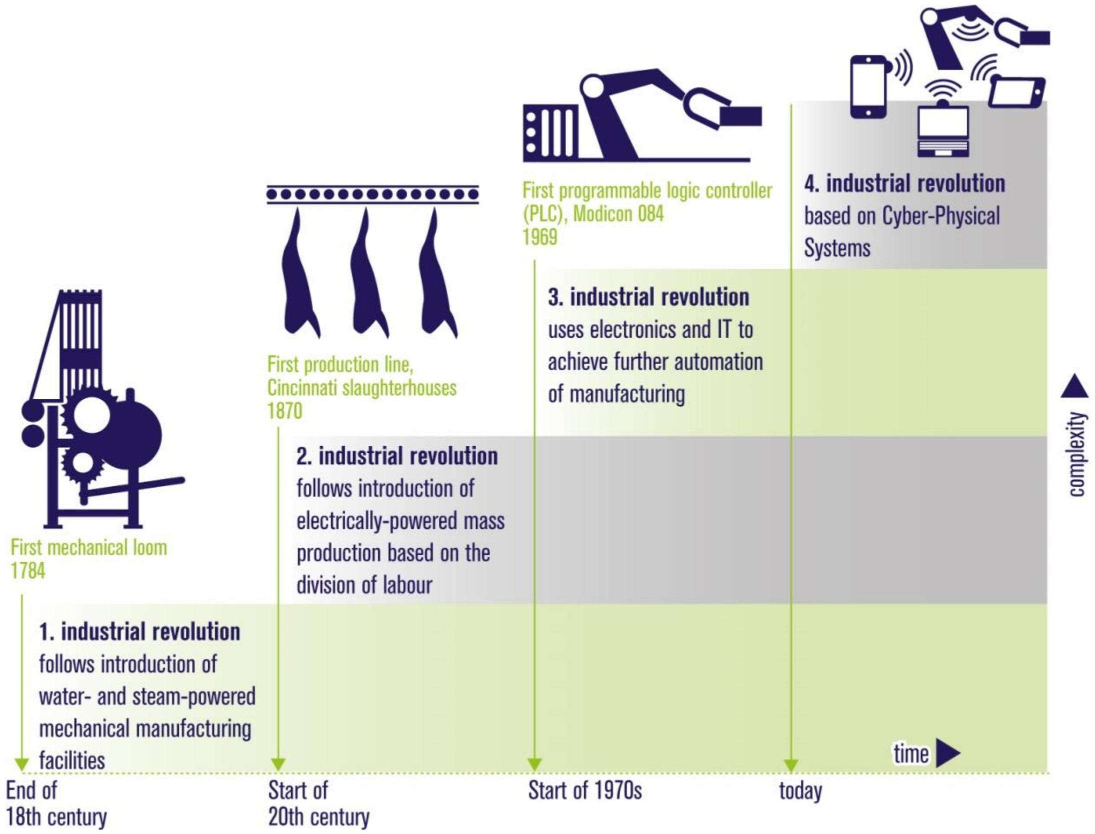

Figure 1: Occurrence of industrial revolutions over time [1]

2. Industry 4.0 – Main Concepts

Industry 4.0 (I4.0) can be considered as a collection of concepts, methods, and tools for building the smart factory of the future. This smart factory consists of networked resources (machines, robots, conveyors, storages, shelfs, …) and products. This is often referred to as bringing the Internet of Things and Services to the shopfloor. Networking machines and robots is not an entirely new concept; in fact techniques such as industrial ethernet or Profibus have been used for many years in industry to network resources on the factory floor [2].

The novelty introduced by I4.0 can be seen in the adoption of the concept of cyber-physical systems (CPS). CPS are systems with embedded software and electronics. They are connected to the outside world via sensors and actuators and can communicate with other CPS via standard network technologies. CPS can collect physical data (through its sensors), store this data, and analyze it. On this basis they can interact with their environment. CPS can be equipped with capabilities for autonomous decision making.

The latter capability constitutes a major paradigm shift: Resources (now considered a CPS) are no longer strictly networked and connected to a central control computer (which makes the decisions), rather they are allowed to communicate with other CPS and perform decentralized decision making.

A typical metaphor used to exemplify the capabilities that this approach offers is that “the product defines its way through the production”. A product, considered as a CPS, can communicate with the machine (also a CPS) that it needs for processing and request the needed processing steps. Thus, the product can negotiate with the different CPS in the production system (machines, robots, carriers, …) about its processing. It can “negotiate” its way through the production. Beyond that, machines can negotiate with each other about sequencing and tasks distribution, material requirements etc.

In reality, a typical implementation of the above mentioned metaphor has products that are equipped with RFID tags. By reading the RFID tag, a machine can determine what to do with the product. It can also store certain amounts of data onto the RFID tag of the product, e.g., to document which process steps have been completed.

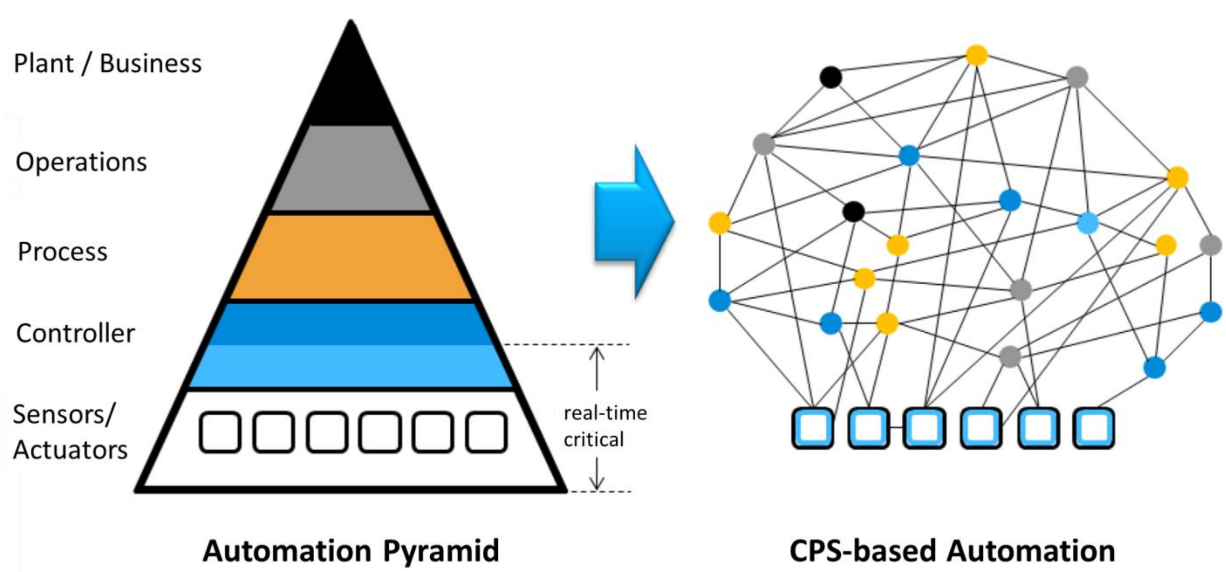

Figure 2: Industry 4.0 suggests a paradigm shift from a strictly hierarchical automation control structure towards a decentralized automation and control based on cyber-physical systems (adopted from [3]).

If taken strictly, the suggested move towards CPS-based automation constitutes a major paradigm shift concerning the automation and control hierarchy within production and logistics (Figure 2) - away from the classical hierarchical automation pyramid towards a meshed connection hierarchy.

In summary, I4.0 puts forward the concept of networked CPS capable of negotiating, reasoning, and autonomous decision making. This goes beyond the concept of networked resources whose data can be centrally collected and which can be controlled by centrally issued instructions. With that, I4.0 intends to bring distributed intelligence and decision making to the shop floor.

3. Simulation for Industry 4.0

In I4.0, a factory can be considered as a collection of multiple more or less independent CPS. Such a factory is often referred to as a “cyber-physical production system” (CPPS). Developing a CPPS constitutes a major challenge, as the complexity of meshed systems with distributed agent-like decision making is not necessarily easier to cope with compared to centralized decision making. In fact, it is even unclear at the current time, if the suggested paradigm shift from centrally controlled factories towards decentralized control is even beneficial concerning relevant economic key performance indicators of a factory. According to [4] initial investigations show that decentralized production control may be beneficial for cycle times and flexibility, but disadvantageous towards capacity utilization and capital commitment.

In any case, the planning of smart factories independent of the chosen control paradigm requires upfront simulation of its components and their interplay. In I4.0, the individual CPS should be simulated as well as their interplay for forming the CPPS.

The up-front development and simulation of CPS is often associated with the term of building the “digital twin” for the respective CPS. Digital twins can be defined as digital representations which represent the real system *and* its current state in a digital model. For investigating their dynamic behavior, digital twins must have properties typically associated with simulation models.

Different modeling approaches can be chosen for implementing the digital twin, depending on the desired level of detail. For the digital twin of the overall CPPS, modeling approaches based on discrete-event simulation are a typical choice, e.g., based on commercial material flow simulation packages [5]. These models allow for the simulation of logistic and economic aspects of the entire factory, e.g., throughput, utilization of resources, cycle times, etc. The difference between traditional usage of such simulation models and the “digital twin” is the requirement that the digital twin also needs to reflect the current state of the real system. Such capabilities are often associated with the terms “online simulation” or “symbiotic simulation”. In both approaches the simulation model maintains some kind of data link to the real system which is used to initialize and update the model state with the state of the real system. The simulation model is supposed to adjust its state according to these updates and perform its simulation tasks between updates. In the case of symbiotic simulation, also some kind of control feedback from the simulation model towards the real system is expected.

In many material flow simulation systems, resources are often modeled as passive entities. This applies to many systems that follow the transaction-oriented world view as well as to systems with a process-oriented world view, the latter being widely used today. Modelling resources as passive entities somewhat contradicts the philosophy of I4.0, in which resources are supposed to exhibit their own “intelligent” behavior. Only few simulation packages allow for a combination of different modeling methods. Among them are Anylogic [6] and SLX [7]. Anylogic offers the option of combining the classical process-oriented world view (with passive resources; Anylogic calls it “discrete event modeling”) and an agent-based world view (allowing for state-based modeling of machines and their direct interaction). SLX offers the user complete flexibility in modelling entities and resources as active or passive objects and any combination thereof. The choice of the simulation system for modelling I4.0 compliant CPPS should therefore closely look at the options of the intended tool for modelling behavior and direct interactions of resources.

In many cases, it may be desirable or even required to not only model the overall CPPS, but to also closely look at the individual CPS and high-resolution simulation models for them. Such models can be on the level of robot simulation (with detailed kinematic models of the robot), they can be on the machining level, where behavior of machines with its NC-codes and its effect on the workpiece is simulated, they can include models of automation components, and so on. Anything that can be considered a CPS can have its own model at its required level of detail.

To simulate the overall smart factory at a detailed level of resolution, the high-resolution digital twins of each individual CPS will have the need to communicate with other digital twins; hence a modular approach for building distributed simulations (“federations”) of digital twins is needed. Beyond that, it may also be desirable to build mixed simulation federations partially consisting of digital twins and partially consisting of multiple real CPS. This raises the need for standardized communication between the digital twin and the real CPS.

Concerning the hierarchical view on the smart factory, it is obvious that the combination of CPS forms the CPPS. Not so obvious, but also a reality, is that a CPS can be an aggregation of different CPS. This also has implications on the combined simulation. What we need are approaches for combining the digital twins of individual CPS into an overall simulation of either an aggregated CPS or the entire CPPS. Considering that several hierarchy levels can be involved, a systems-of-systems approach for building federations of digital twins may be needed.

4. Interoperability Standards

4.1 Simulation Interoperability

4.1.1 History and Adoption of the HLA

The High Level Architecture for Modeling and Simulation, or HLA for short, is an IEEE standard for distributed simulation. HLA’s core objective is to facilitate interoperability and reusability among a wide range of simulation applications and types. HLA had been in launched in the mid-1990s [8] where it was the designated successor of earlier military distributed simulation standards like ALSP and DIS. HLA has its origin in the military simulation community where one of its major tasks is the networking of military training simulators. However, due to its openness and generic character it also has a large potential for non-military distributed simulation applications.

HLA is not undisputed in certain market niches (cf. Tena for test and training simulations [9]), but it can certainly still be considered as the leading standard for simulation interoperability.

While HLA's adoption for military simulation applications was temporarily promoted with a mandate to comply with the standard, HLA has also received significant attention from the civilian simulation community. Most of this attention originated from academia ([10][11][12][13]) and has been rather research oriented. Significant efforts have been focused on using HLA as a standard for interoperability between commercial of the shelf simulation packages [14][15][16].

HLA's adoption in manufacturing industries is a somewhat different story. Although several companies have experimented with the HLA, it is far from being used in day-to-day simulation operations. Among those who experimented with serious practical applications of HLA were Daimler from the automotive sector [17] and Deere & Co. from the agricultural and construction equipment industry [18]. Especially for the automotive industry with its large supplier networks and rather advanced use of digital planning and simulation methods within their Digital Factory efforts, HLA was accredited a substantial role for providing plug-and-play simulation interoperability.

However, reality has not seen a widespread adoption of HLA in manufacturing industries [12][19][20]. There are several reasons for this, and none of them relates to technical weaknesses of the HLA:

1) Although simulation has received a major push with the digital factory initiatives of many OEMs, simulation is still most often tackled with a monolithic approach, using a single designated simulation package for building monolithic models.
2) Manufacturing industries heavily rely on commercial-off-the-shelf (COTS) simulation packages. Enabling plug-and-play interoperability between COTS simulation packages based on the HLA requires the adoption of the HLA standard by COTS package vendors. There has been little sustained interest on the vendor side.
3) Simulation interoperability requirements in industry have shifted focus to interoperability between simulation systems and non-simulation systems. This includes planning tools (process planning, layout planning) as data deliverers, but also factory control systems (enterprise resource planning - ERP and Manufacturing Execution Systems – MES), which feed data into the simulation and receive simulation forecasts for decision support.

In summary, we are confronted with the fact that HLA, although technologically capable of solving simulation interoperability in manufacturing industry, has not gained the anticipated widespread use in this sector. It remains to be seen how that situation changes with the implementation of Industry 4.0.

4.1.2 Technical Aspects of the HLA

On the technical side, the HLA standard consists of three core components, defined by the IEEE 1516 series of standards:

- 1516-2010: Framework and Rules [21]
- 1516.1-2010: Federate Interface Specification [22]
- 1516.2-2010: Object Model Template (OMT) Specification [23]

For reasons of brevity, we limit the following discussion to the Federate Interface Specification and assume that the reader is familiar with basic HLA concepts and terms. For those who are not, a concise introduction can be found in [8]. HLA differentiates between the simulation functionality provided by the members of the distributed simulation ("federates") and the set of basic services for data exchange, communication, and synchronization provided by a runtime infrastructure (RTI). The HLA Federate Interface Specification defines theses services. The interface specification describes which services can be used by a federate and which services it has to provide [22].

This bi-directional character of the interface is encapsulated into an ambassador paradigm. A federate communicates with the RTI using its RTI ambassador. Conversely, the RTI communicates with a federate via its federate ambassador. From the federate programmer's point of view, these ambassadors are objects and the communication among the participants is performed by calling methods of these objects. Thus, the services defined in the interface specification are either methods of the RTI ambassador or of the federate ambassador.

The interface specification defines six categorizes of services, from which we only highlight three, namely time management, declaration management, and data distribution management.

The time management services provide a mechanism for coordinating simulation clocks of simulations using a wide variety of time advance mechanisms. In comparison with other technologies, where time management/synchronization is only available to a certain type of simulation, or not at all, HLA provides a general solution for all types of simulations.

Declaration management provides services for establishing publisher/subscriber relationships for categories of data (namely object and interaction classes) to be exchanged at runtime. Available classes are described in the federate and federation object models, which can also be used to verbally describe semantics. There is no limitation on the number of publishers or subscribers of a certain class, i.e., HLA allows for many to many (m:n) relationships for data transfer.

The services provided in the data distribution category (which can be used optionally) provide further mechanisms for refining the data exchange needs beyond the class based publish and subscribe mechanisms. They provide services for value-based filtering. This is achieved be defining multi-dimensional routing spaces and associating data updates to update regions. Data gets only transferred, if a publish and subscribe relation for that class exists, and update and subscription regions overlap.

The combination of both declaration and data distribution management can significantly reduce the amount of data transferred. These services are special in the regard that previous technologies (like DIS) were usually based on broadcast principles for distributing data.

Overall, HLA has its strengths in connecting simulation applications, as

- its time management services allow the synchronization of simulation clocks (a feature that other interoperability standards are missing), and
- its data management services provide efficient, and if needed time-synchronized, data exchange mechanisms.

HLA has no built-in support for the systems-of-systems concept mentioned in section 3. Individual federates are combined into a federation, but there is no notion or concept for building “federations of federations”. A potential workaround, but no equivalent concerning aggregation/deaggregation, is the option of “bridging” federations by using dedicated bridge federates [24], i.e., federates that are members in more than one federation. They can perform the tasks of exchanging data between different federations and to a certain extend synchronize these federations.

HLA does also not offer any specific support to connect to control hardware or life equipment. Any component, that needs to be brought into a federation must implement the HLA interface specification and behave according to the HLA rules [21]. Workarounds exist (e.g., using surrogate or “stand-in” federates), but this actually contradicts HLA’s interoperability philosophy.

Further to this, HLA does not define a wire-standard for internal communication of the RTI, and also does not set any encryption requirements in the standard.

4.2 Interoperability within Industry 4.0

4.2.1 Terminology

Standardization in I4.0 at the current time has to be considered as work-in-progress. Attempts are made for a technology-independent standardization of interoperability based on a service architecture [25]. Concerning terminology, standardization in Industry 4.0 introduces several new terms that go beyond the guiding terms CPS and CPPS introduced earlier [26].

If CPS and CPPS are to be implemented in an I4.0 compliant way, they are considered *I4.0 systems*. An *I4.0 system* is a system consisting of *I4.0 components* and non I4.0 components which serves a specific purpose, has identified properties, and supports standardized services and states [26].

An *I4.0 component* is defined as a "globally uniquely identifiable participant with communication capability consisting of administration shell and asset […] within an I4.0 system which there offers services with defined QoS (quality of service) characteristics. […] An I4.0 component can represent a production system, a single machine or station, or even an assembly within a machine." [26]. Although not explicitly discussed, the latter implies the inclusion of a "system-of-systems" concept in I4.0; whereas I4.0 components can be comprised of other I4.0 components.

An *asset* (sometimes also: *technical asset* [27]) is an item which has a value for an organization. It can be accessed through its *administration shell*. The *administration shell* (often also referred to as *asset administration shell*, ASS) is "a virtual digital and active representation of an I4.0 component in the I4.0 system" [26].

Lastly, an *I4.0 platform* is an "implementation of a standardized communication and system infrastructure with the necessary management and production services and defined QoS (quality of service) characteristics as a basis for efficient construction and integration of I4.0 systems in an application domain" [26].

A somewhat oversimplified attempt to map these I4.0 terms to terms known from the HLA is displayed in table 1.

Table 1: Comparison of terms from Industry 4.0 and the HLA

Term from Industry 4.0	Term from the HLA	Major Difference
I4.0 Platform	Runtime Infrastructure (RTI)	Services of an I4.0 Platform are only defined in a generic (and currently rather vague) form, no API standard; RTI functionality is determined by IEEE 1516 set of standards, federate interface specification defines API
I4.0 Component	Federate	I4.0 Component is typically comprised of some form of hardware (e.g., an automation component), a federate is most often considered a piece of software ("simulations, supporting utilities, or interfaces to live systems" [22])
I4.0 System	Federation	I4.0 System can contain components not capable of I4.0 compliant communication; all participants of a federation must comply with the HLA standard.
Asset	Federate Code	Asset is typically a piece of hardware; federate code is software
Asset Administration Shell	Federate Interface Specification	Both define the interface with which to interact with an asset / a federate. Formality of definition differs (ASS: generic, HLA: formal standard).

4.2.2 General Approach towards Interoperability

Attempts are made for a technology-independent standardization of interoperability in I4.0 systems based on a service architecture [25].

The scope of the standardization comprises (among other aspects) the definition of the asset administration shell (in terms of exposed data and functionality), the definition of I4.0-compliant communication, and the specification of the service architecture. The service architecture is intended to be defined in a hierarchical way using the hierarchy depicted in Figure 3, but is completely technology independent.

According to the layers of the service hierarchy, four different types of services are envisioned. Communication services are the lowest level in this layered approach and define the primitive services required to perform data transfers. On top of these, information services shall define the basic functionality required to work with information models. Higher-level services are built on-top of the information services and include platform services that define the self-management functionality of an I4.0 system and application services which provide the actual functionality for building productions systems [25].

[25] also states that “information services are defined in a technology-independent manner” in order to “be stable under the evolution and replacement of middleware technologies, and to allow interoperability between different middleware technologies. They can be considered as the conceptual, technology-independent interface of the information layer, which is then mapped onto technology-specific protocol(s) such as OPC UA.”

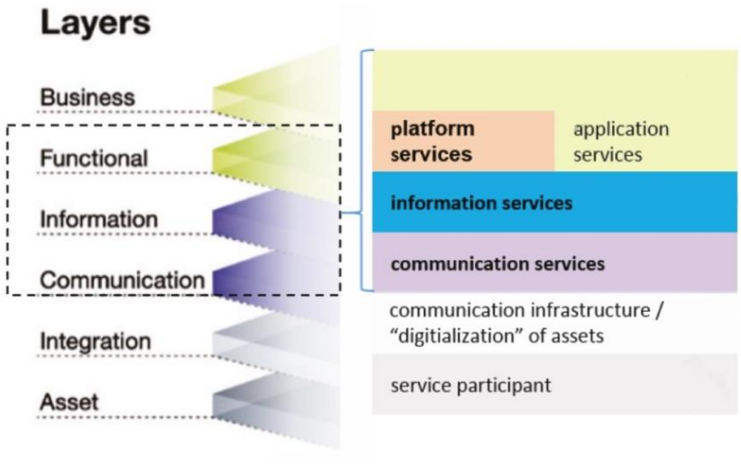

Figure 3: The Industry 4.0 Service Hierarchy [25]

The current state of the standardization of the service architecture can best be described as work in progress. Much of it even seems to be in a rather early stage. Also, it is unclear, whether the completely technology independent approach taken towards standardizing interoperability in I4.0 is productive at all.

In the end, the technology-independent service architecture has to be implemented somehow. At the current time, the prevailing idea is to map the operations and service calls of the generic service architecture onto one or more existing technological solutions. The OPC UA standard (Open Platform Communications Unified Architecture) is a candidate frequently suggested for this purpose. OPC UA is internationally standardized as IEC 62541 and already has a good acceptance and adoption rate from automation equipment manufacturers.

As future interoperability in Industry 4.0 seems to strongly hinge on the capabilities of OPC UA, a closer discussion of OPC UA seems to be appropriate and potentially more productive than discussing the current state of standardization of Industry 4.0.

4.2.3 Open Platform Communications (OPC) Unified Architecture (OPC UA)

OPC UA is a multipart standard defined in the IEC 62541 series of standards. OPC’s roots date back to 1996, where OPC attempted to standardize manufacturer independent interfaces for data exchange in automation engineering. OPC was bound to Microsoft’s DCOM (distributed component object model, earlier: OLE, object linking and embedding) technology and was thus not platform independent.

OPC UA was first released in 2006 and replaces all earlier OPC specifications. It is defined in a platform- and DCOM-independent way. The objective of OPC UA is still to enable interoperability between and access to automation components. The general approach taken towards standardization is to define standardized services [28] which are used to interact with an information model [29] on a remote server. Every service is defined as a request and a response message, whereby message responses occur asynchronously. Taken strictly, OPC UA is therefore a pure client-server-communication protocol. OPC UA does have built-in mechanisms for push notifications, but in its base form, this always relates to a client-server relationship and is based on TCP/IP communication. OPC UA is therefore regularly “abused” to usages where many clients are listening to changes in a single server. Only an optional extension to OPC UA (“PubSub” [30]) alleviates the ramifications of this design choice.

The OPC UA information model specifies eight types of nodes, among them are object nodes which can consist of variables (“variable nodes), methods (“method nodes”), and further objects. The information model also allows the definition of reference nodes that can be used to model relations, e.g., between object nodes. Overall, the expressiveness of the OPC UA information model is comparable to the features offered by the HLA object model template (OMT) [23], although the HLA OMT does not directly allow the specification of methods of federates. The OPC UA information model allows such specification of methods. They always belong to object nodes and are callable by clients via the OPC UA service specification.

OPC UA information models are specified in an XML format and can be imported by OPC UA servers – comparable to the treatment of HLA simulation / federation object models.

OPC UA has security mechanisms built-in into the standard [31]. They provide authentication of users and application instances as well as confidentiality and integrity by signing and encrypting messages. Security includes a secure channel and session concept for communication between clients and servers and is based on state-of-the art encryption technologies.

Unlike the HLA, OPC UA also provides an entire communication stack (the OPC UA stack) that implements the different OPC UA transport mappings defined in [32]. This definition includes communication and transport details down to the wire transport of integers and floating point numbers (“little endian”).

The service specification of OPC UA is structured into service sets. These service sets include mechanisms for discovering servers, securing communication channels, and management of sessions. The actual data transfer services include pull and push mechanisms for accessing data on OPC UA servers. Pull mechanisms allow clients to read attribute values from nodes/objects residing on a server. Push mechanisms allow clients to create subscription sets of monitored items (e.g. attributes from server nodes/objects). Servers fulfil these subscription requests by issuing notifications.

In essence, the described data distribution mechanism has some significant differences compared to one of the HLA. Although it does include a class and attribute based publish/subscribe mechanism, this mechanism is always targeted at one-to-one client-server communications. If many clients create subscriptions for the same data objects on a server (a likely scenario, e.g., when many clients listen for changes on the server), there is always the need to service *n* communication links for *n* clients. This also limits the capabilities for maintaining hard real-time requirements.

The “PubSub” extension to OPC UA [30], first released in 2018, addresses this problem by introducing true publisher-subscriber relationships between servers and clients as known from the HLA. It also introduces a middleware component to enable a loose coupling between subscribers and publishers. However, PubSub is merely an extension of OPC UA and does not replace the client/server protocol that is integral part of OPC UA.

4.3 Discussion

Both interoperability standards introduced in the previous sections (HLA and OPC UA) target at enabling interoperability within their respective domains of application. While HLA is targeted at interoperability of simulation systems, OPC UA is targeted at interoperability of automation and control equipment. Both standards have unique selling points crucial for their application domains: HLA provides dedicated synchronization support for simulation systems and seems to have the more sophisticated data distribution concepts. OPC UA has included security mechanisms that are mandatory when access to productive equipment in manufacturing is concerned. Also, OPC UA provides detailed mappings for the transportation layer (the “wire standard” that HLA is missing). OPC UA in its base form does not truly support the idea of data exchange between many participants, where each participant can act as data publisher and data subscriber. To enable this, applications have to act as both OPC UA server and OPC UA client in one application.

For enabling Industry 4.0 and enabling seamless integration of different CPS (or “I4.0 components”) and their digital twins both standards will need to evolve. With the PubSub extension, OPC UA seems to have taken a much needed step away from the strict client-server communication paradigm, towards a more performant publish-subscribe based data exchange using some kind of middleware (providing “many to many” communication), but still allowing for direct 1:1 communication between participants that may be needed for “negotiating” in I4.0-style.

On the other hand, OPC UA has no features for facilitating interoperability to or between simulation systems / digital twins. While there is nothing to prevent individual simulation systems to act as an OPC UA client (or even as an OPC UA server) and connect to control equipment (or even their digital twins) is this fashion, OPC UA alone is no appropriate concept for interconnecting simulation systems.

At the current time, no final recommendation can be given for the best path to choose. The ideal standard would provide a unified interoperability solution for both domains. Such a standard would provide a unified service and information model definition and allow the seamless switching between real system and its digital twin.

Considering both the experience with acceptance of HLA in manufacturing industries, and the rather early stage of I4.0 standardization, a convergence of both interoperability standards is not very likely to happen in the near future. As always, successful standardization needs an active and engaged group of people backed by an industry demand. With OPC UA, an industry demand (driven by automation equipment providers and their users) is given. Raising awareness of simulation capabilities in this group might be a path towards achieving extensions of OPC UA for facilitating simulation interoperability.

A different approach towards convergence of both standards could be to enable the HLA to allow federates to easily (or even transparently) connect to OPC UA servers and act as OPC UA clients. Initial attempts could focus on bridge federates and unified information models. In HLA terms, an "Industry 4.0 Reference FOM" could be a first step towards bringing both standards together. A second step could be an HLA add-on for OPC UA communication.

5. Summary

This paper has identified simulation based on digital twins as an important success factor for implementing the I4.0 vision. I4.0 brings about the idea of the smart factory consisting of interconnected CPS. Their digital twins have the need to communicate with each other and with their real counterparts. Interoperability is therefore crucial for the success of the I4.0 vision. The paper has reviewed two leading interoperability standards from the respective domains, namely HLA for simulation interoperability and OPC UA for interoperability of automation equipment. Both standards have their unique selling points within their domains of application, but none of them alone can fulfill the interoperability requirements of both domains. The paper has suggested different options towards a convergence of the standards.

6. References

[1] Kagermann, H.; Wahlster, W.; Helbig, J.: *Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group*. https://www.acatech.de/wp-content/uploads/2018/03/Final_report__Industrie_4.0_accessible.pdf

[2] Lin, Z.; Pearson, S.: *An inside look at industrial Ethernet communication protocols*. Texas Instruments. 2018. http://www.ti.com/lit/wp/spry254b/spry254b.pdf

[3] N.N. *Statusbericht; Industrie 4.0; Cyber-Physical Systems - Chancen und Nutzen aus Sicht der Automation*. VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik, Düsseldorf, VDI e.V., 2014.

[4] Mertens, P.; Barbian, D.: *Digitalisierung und Industrie 4.0 – Moden, modische Überhöhung oder Trend?* Arbeitsbericht Nr. 1/2016, 2. Aufl. Friedrich-Alexander-Universität Erlangen-Nürnberg. http://wi1.uni-erlangen.de/sites/wi1d7.wi1projects.com/files/publications/digi_kurz_02022016.pdf

[5] Swain, J. J.: *Simulation software survey: Simulated worlds*. OR/MS Today, Vol. 42, No. 5, October 2015, pp. 36-49.

[6] Borshchev, A.: *The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6*. Anylogic North America. 2014.

[7] Henriksen, J.O.: *SLX: the X is for extensibility*. In: Proceedings of the 2000 Winter Simulation Conference, eds. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, pp. 183-190, Orlando, USA, December 10-13, 2000.

[8] Strassburger, S.: *Overview about the High Level Architecture for Modelling and Simulation and Recent Developments*. In: Simulation News Europe, Vol. 16, No. 2, pp. 5-14, September 2006.

[9] Noseworthy, J.R.: *The Test and Training Enabling Architecture (TENA) Supporting the Decentralized Development of Distributed Applications and LVC Simulations*. In: Proceedings of the 12th IEEE/ACM

International Symposium on Distributed Simulation and Real-Time Applications (DS-RT 2008). Vancouver, BC, Canada. October 27-29, 2008.

[10] Schulze, T., Strassburger, S.; Klein, U.: *Migration of HLA into Civil Domains: Solutions and Prototypes for Transportation Applications*. SIMULATION, Vol. 73, No. 5, pp 296-303, November 1999

[11] Strassburger, S.: *Distributed Simulation Based on the High Level Architecture in Civilian Application Domains*. SCS-Europe BVBA, 2001.

[12] Boer, C.A.: *Distributed Simulation in Industry*. Rotterdam: Erasmus Research Institute of Management, 2005.

[13] Lendermann, P.; Heinicke, M.; McGinnis, L.; McLean, C.; Strassburger, S.; Taylor, S.: *Distributed Simulation in Industry - a Real-world Necessity or Ivory Tower Fancy?* In: Proceedings of the 2007 Winter Simulation Conference, eds. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, pp. 1053-1062. December 9-12, 2007. Washington, D.C., USA.

[14] Taylor, S.; Turner, S.; Strassburger, S.: *Guidelines for Commercial Off-The-Shelf Simulation Package Interoperability*. In: Proceedings of the 2008 Winter Simulation Conference, eds. S. J. Mason, R. R. Hill, L. Moench, O. Rose, pp. 193-204. December 7-10, 2008. Miami, USA.

[15] Taylor, S. J. E.; Turner, S. J.; Strassburger, S.; Mustafee, N.: *Bridging the Gap: A Standards-Based Approach to OR/MS Distributed Simulation*. ACM Trans. Model. Comput. Simul. 22, 4, 1–23.

[16] Strassburger, S.; Taylor, S.: *A Comparison of the CSPI and CMSD Standards*. In: Proceedings of the 2012 Spring Simulation Interoperability Workshop. Orlando, USA, March 26-30, 2012.

[17] Strassburger, S.; Schmidgall, G.; Haasis, S.: *Distributed Manufacturing Simulation as an Enabling Technology for the Digital Factory*. Journal of Advanced Manufacturing Systems (JAMS). Vol. 2, No. 1 (2003) 111-126.

[18] Raab, M.; Schulze, T.; Strassburger, S.: *Management of HLA-Based Distributed Legacy SLX-Models*. In: Proceedings of the 2008 Winter Simulation Conference, eds. S. J. Mason, R. R. Hill, L. Moench, O. Rose, pp. 1086-1093. December 7-10, 2008. Miami, USA.

[19] Strassburger, S.; Schulze, T.; Fujimoto, R.: *Future Trends in Distributed Simulation and Distributed Virtual Environments: Results of a Peer Study*. In: Proceedings of the 2008 Winter Simulation Conference, eds. S. J. Mason, R. R. Hill, L. Moench, O. Rose, pp. 777-785. December 7-10, 2008. Miami, USA.

[20] Strassburger, S.; Schulze, T.; Fujimoto, R.: *Future Trends in Distributed Simulation and Distributed Virtual Environments*. In: Advancing the Frontiers of Simulation: A Festschrift in Honor of George Samuel Fishman, eds. C. Alexopoulos, D. Goldsman, J. R. Wilson, pp. 231-261. Springer 2009.

[21] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)*. *Framework and Rules*. Institute of Electrical and Electronics Engineers, New York, IEEE Std 1516-2010.

[22] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)*. *Federate Interface Specification*. Institute of Electrical and Electronics Engineers, New York, IEEE Std 1516.1-2010.

[23] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)*. *Object Model Template (OMT) Specification*. Institute of Electrical and Electronics Engineers, New York, IEEE Std 1516.2-2010.

[24] Dingel, J.; Garlan, D.; Damon, C.: *Bridging the HLA: Problems and Solutions*. In: Proceedings of the Sixth IEEE International Workshop on Distributed Simulation and Real Time Applications (DS-RT 2002), Fort Worth, Texas, USA. October 11-13, 2002.

[25] N.N. *Status Report; Industrie 4.0 Service Architecture; Basic concepts for interoperability*. VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik, Düsseldorf, VDI e.V., 2016. https://www.vdi.de/index.php?id=58655&pubid=48

[26] N.N. *Status Report; Industrie 4.0 Begriffe/Terms*. VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik, Düsseldorf, VDI e.V., 2017. https://www.vdi.de/index.php?id=62349

[27] N.N. *Status Report; Industrie 4.0 – Technical Assets; Basic terminology concepts, life cycles and administration model*s. VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik, Düsseldorf, VDI e.V., 2016. https://www.vdi.de/index.php?id=58664&pubid=11

[28] *OPC Unified Architecture – Part 4: Services*. International Electrotechnical Commission, Geneva, Switzerland, IEC 62541-4:2015.

[29] *OPC Unified Architecture – Part 5: Information Model*. International Electrotechnical Commission, Geneva, Switzerland, IEC 62541-5:2015.

[30] *OPC Unified Architecture – Part 14: PubSub*. International Electrotechnical Commission, Geneva, Switzerland, IEC 62541-14:2018 (To be released).

[31] *OPC Unified Architecture – Part 2: Security Model*. International Electrotechnical Commission, Geneva, Switzerland, IEC TR 62541-2:2016.

[32] *OPC Unified Architecture – Part 6: Mapping*. International Electrotechnical Commission, Geneva, Switzerland, IEC 62541-6:2015.

Author Biography

STEFFEN STRASSBURGER is a professor at the Ilmenau University of Technology and head of the Group for Information Technology in Production and Logistics. Previously he was head of the “Virtual Development” department at the Fraunhofer Institute in Magdeburg, Germany and a researcher at the DaimlerChrysler Research Center in Ulm, Germany. He holds a Doctoral and a Diploma degree in Computer Science from the University of Magdeburg, Germany. He is further an associate editor of the Journal of Simulation. His research interests include distributed simulation, automatic simulation model generation, and general interoperability topics within the digital factory and Industry 4.0 context. He has been engaged in SISO and involved with the HLA standard since 1997. His web page can be found at www.tu-ilmenau.de/itpl.