SKELLIG

4.0 WHITE PAPERS: AN EDUCATIONAL SERIES

Unlocking Industry 4.0 across Life Sciences starts with the Unified Namespace (UNS)

The second paper in the Skellig Industry 4.0 series takes a close look at the concept of the UNS and how leveraging its architecture can accelerate the benefits of Industry 4.0 solutions for Life Sciences. Then it elaborates on the architecture used to create a UNS.

ABOUT THE AUTHORS

Amy Williams is an automation engineer experienced in on-site automation support and technical leadership for distributed control systems integration projects in life sciences. She is also a certified Industry 4.0 professional from 4.0 Solutions. Amy has been working with Skellig since 2020.

Purnendu Saptarshi is a senior automation engineer with over a decade of experience in designing and implementing process control systems in life sciences. Additionally, he has experience integrating equipment with SCADA systems. He is a certified Industry 4.0 professional from 4.0 Solutions. Purnendu has been working with Skellig since 2014.

The first step towards Pharma 4.0™ is implementing a UNS architecture at the factory level

A Unified Namespace (UNS) is a software solution that acts as a centralized hierarchical repository of data, information, and context where any application or device can publish or consume data needed for its use. Each application or device is uniquely identified within a common contextual architecture. By normalizing disparate data structures across the enterprise within a single hierarchical framework, the UNS creates seamless business data connectivity.

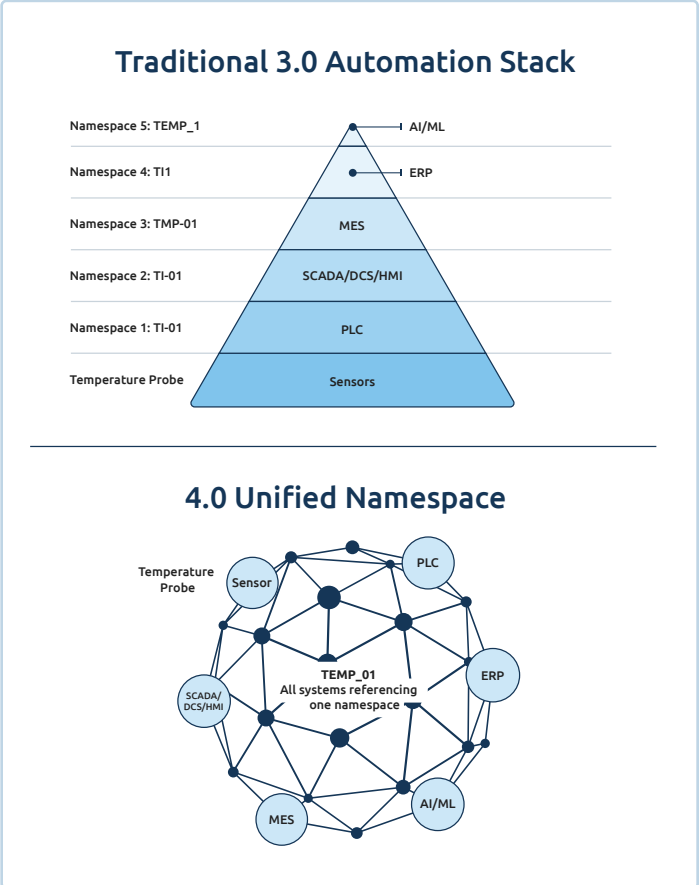

Figure 1: Ideal transformation to a Unified Namespace. **Note:** In a 3.0 automation stack, a single probe would need configuration in five namespaces, for five systems. In a 4.0 UNS architecture, all five systems share one namespace. Once the probe is connected to the UNS, the five systems are all reading the same single value from the one shared namespace.

Rather than having isolated namespaces across the business, each software system gathers its data from one Unified Namespace. Once a new data point is added to the UNS, it is immediately accessible to all nodes in the business. A node is any application, or software in the network which produces or consumes data.

As discussed in Paper 1, the main challenge for a 3.0 architecture is the cost of connecting data points across the business. A UNS eliminates the problem of needing an unattainable number of discrete connections for each data point between every namespace.

Every node, such as floor equipment, only requires one connection to the central UNS software. Through that connection, all data points are accessible in real time. Every node will reference the same single data point defined in the UNS. If a temperature probe value updates, every node connected to that point in the UNS will be reading the same updated value instantly.

For example, a scheduler within the ERP system might want to make a schedule based on real-time status of equipment on the plant floor. The ERP system can pull all necessary data points through the connection to the UNS. For a 3.0 factory to achieve this, engineering is required to connect every piece of data from floor equipment directly to that scheduling system.

The Life Sciences industry needs all-encompassing data access because of the regulatory requirements for traceability and context

Unlike with paper, UNS data points inherently contain a timestamp and can be historized with audit trails and configured electronic signatures. This promotes the FDA Data Integrity principles: attributable, legible, contemporaneously recorded, original, accurate, complete, enduring, consistent, and available (ALCOA+).[1]

Furthermore, since every node is referencing the same name for a data point, enterprise-level data mapping is possible.

UNS data mapping unlocks context for investigations

In a 3.0 factory, to gather all context associated with a data point, one would need multiple access points to retrieve siloed information.

For example, usually one application allows for trending historical data of floor sensors, such as temperature, over time. This trend can indicate when a value went out of range. However, it doesn't easily provide an explanation for why it happened. To gain context, one could pull the Electronic Batch Record from a different application. This can provide timestamps of what recipe was loaded, and when automated prompts were acknowledged. However, most manual activities are not hard coded into the automated recipes. Thus, one would also need to pull the paper SOP/MBR and associated documentation references to understand what activities were performed during that time. Once this information is gathered, a root cause investigation can begin.

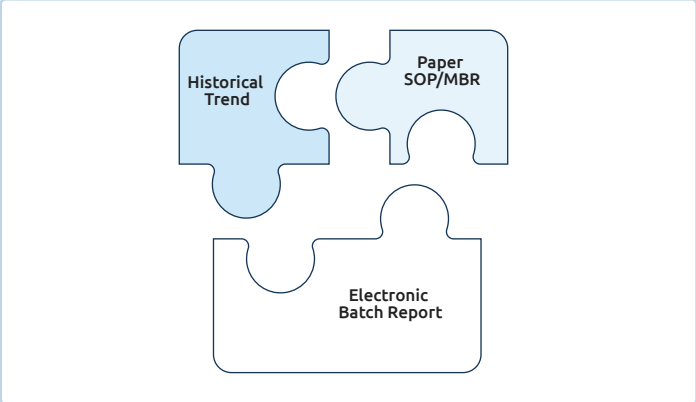

Figure 2: Building context in a 3.0 factory. **Note:** To build context during an investigation, it is common to pull data and information manually from multiple sources. Upper left: historical trends of sensors on the plant floor. Bottom: Electronic Batch Record, which documents any automated prompts and recipes. Upper right: Paper SOP or MBRs, which account for any manual manipulations and instructions performed by the operator, typically not documented in the Electronic Batch Record.

With an organized UNS, where paper instructions are digitized, contextualized data mapping is possible with a single search. Time is saved by not needing to set up multiple accounts, execute multiple queries, or manually retrieve information to gain context. Rather, one account with the appropriate security settings can be configured to query all data from the UNS. Second, with a contextualized data lake, many 4.0 analytic tools become available. Many 3.0 factories only provide time series data for a fraction of all available data, and do not readily expose it to be consumed by analytical tools. A UNS architecture, on the other hand, exposes all data required to build a data lake robust enough to successfully utilize AI and ML. Potential applications include predicting trends, optimizing processes, and analyzing data for compliance and data integrity.[2] These tools could identify abnormal patterns before they can progress to a deviation, and rapidly streamline root cause investigations when necessary.

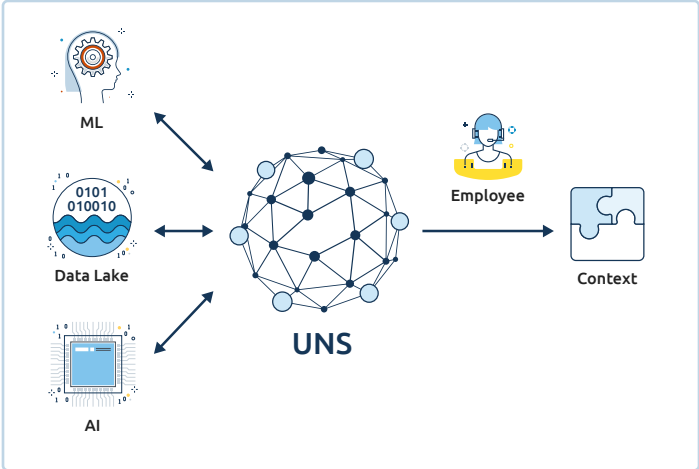

Figure 3: Building context in a 4.0 factory. **Note:** The end user can contextualize data for a specific use case with enterprise data mapping via the UNS. Data from the UNS can be stored in a data lake which can then be analyzed by AI/ML.

This connectivity can also benefit an employee's day-to-day work of implementing changes.

To be compliant, every change in a pharmaceutical factory requires change control with extensive documentation and quality review. Furthermore, often one group cannot begin executing their work until another group's work is completed. For example, facilities cannot begin installing new equipment until engineering has purchased it and updated the P&IDs. This is a common challenge that both 3.0 and 4.0 factories face.

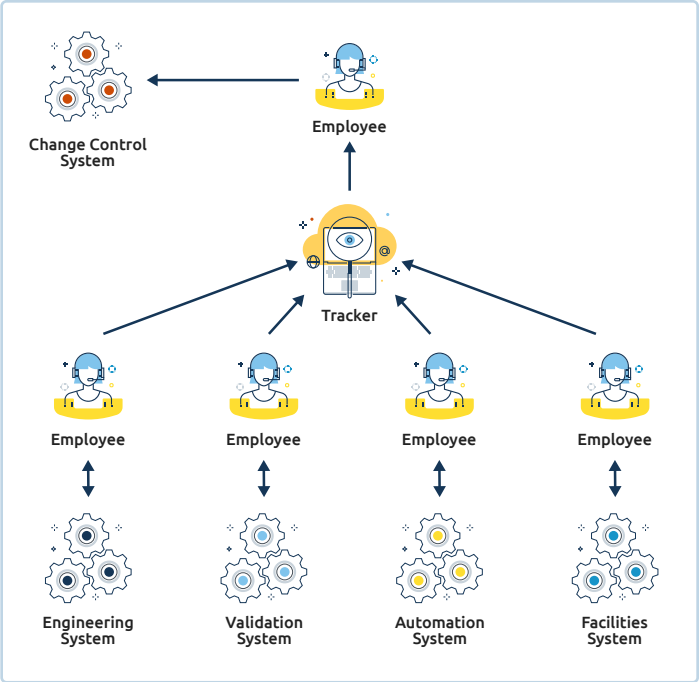

Figure 4: Diagram of traditional 3.0 Change Control tracking hierarchy.

In a traditional 3.0 architecture, it quickly becomes complicated to trace completion of these changes, as there are multiple systems involved that are siloed from each other. As a result, reoccurring meetings are often held solely to consolidate the status of different groups and systems. Managers are left manually updating Excel trackers or other proprietary task manager applications.

Typically, facilities have a master quality Computerized Maintenance Management System (CMMS). A CMMS documents initiation of change control and identifies what required work each group must complete by a specified deadline.

From there, each respective group will have its own system or systems to execute the work. For example, engineering typically utilizes a separate system to track equipment orders. Facilities must document installation progress in an unconnected electronic work order system. Meanwhile, automation has yet another system for storing automated system documents and test scripts.

Each group executes work and closes it out based on group specific Standard Operating Procedures (SOP)s. Due to heavy siloing, process steps, levels of approvals, and systems used to close out work vary between groups.

One critical and common concern is that groups have no knowledge or visibility to systems outside of their group. Closing out the master change control depends on individuals manually performing status updates.

This lack of real-time traceability and context makes it difficult to successfully close out change control processes. Furthermore, humans are error prone. It is possible that a stage may be incorrectly marked complete in a spreadsheet tracker or CMMS system, only to discover missing signatures during approval. As a result, Quality Assurance or the Project Manager must be contacted to manually rollback the work status.

Another common failure occurs when groups execute work on time but fail to notify everyone involved that the change has been completed. As a result, change control actions may still be listed as pending past deadlines, which is not compliant. Worse, if one action item is a pre-requisite to another, failure to notify completion may cascade into delays for starting the next task.

With a UNS, status visibility is no longer siloed to groups or reliant on human transfer. Instead, work status for a change control process could be mapped in real time. Employees could spend time focused on executing work instead of coordinating status. For example, the moment a workflow for a document is complete, that data point of "complete" could be updated automatically in the UNS. The UNS could publish the change to all subscribed

tracking systems, removing dependency on humans to relay status. Project managers could now devote their time to analyzing these statuses and making the data-driven strategic decisions necessary to complete a change control process on time.

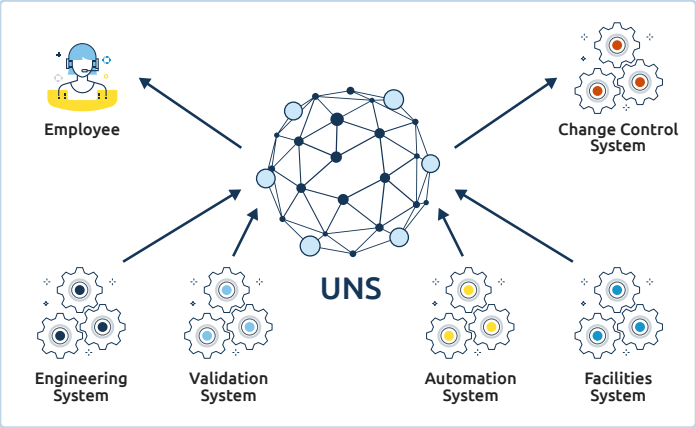

Figure 5: Diagram of 4.0 Change Control tracking hierarchy.

TECHNICAL COMPONENTS OF THE UNIFIED NAMESPACE

The fundamental architecture of the UNS is known as Publisher and Subscriber (PUB/SUB). In this PUB/SUB model, a Publisher is any device or system that creates data. Data is transferred as a message to the central server, also known as a broker. The broker stores all data points within respective "topics" without regard for how those data points are used.

That data can only be read by subscribers who subscribe to that topic. If a topic has no current subscribers, that topic's data is discarded. If data points need to be stored in a database, the historian would have to be subscribed to the topic containing it.

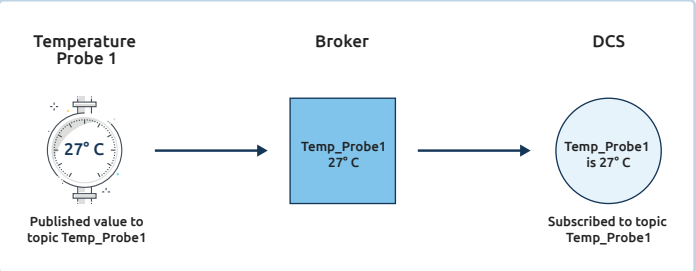

Figure 6: PUB/SUB architecture for a temperature probe being read by a DCS.

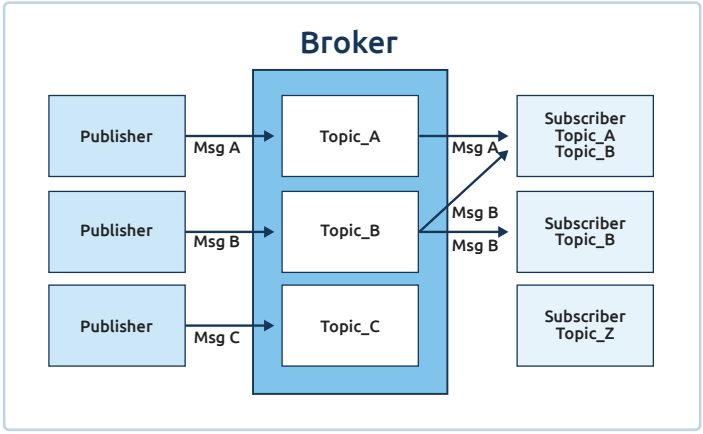

Figure 7: PUB/SUB architecture with multiple topics and subscribers.

In addition to a PUB/SUB structure, a UNS has the following key requirements[3]:

- Edge-Driven Data
- Report by Exception
- Lightweight Protocols
- Use Open-Source Architecture

The **first** requirement is that data flow is edge-driven: controlled by sources that generate data such as transmitters and sensors. The edge devices—the data sources—publish read-only data to topics consumed by subscribers.

The simplicity of this single connection between edge devices and the central server facilitates the maintenance of compliance and security within the UNS.

In many 3.0 architectures, an explicit, discrete connection is required for a higher level to read floor data. Establishing this connection requires opening n number of ports on the floor device, reducing security. With the UNS, only one connection is needed from the edge device to the broker. Since the subscriber reads data from the topic in the UNS, it has no knowledge or connection directly to the device publishing the data. It would not be possible for the upper levels to write to the data sources

via one-way connection unless explicitly permitted. With the UNS, the controlled flow of information from 3.0 architecture is securely maintained without requiring expensive discrete connections.

Consider a case requiring read and write capability. An operator needs to be able to turn PLC equipment on/off through an HMI from a SCADA or DCS system. A read and write pathway must be configured with proper security and audit tracking in place. These write connections most likely require validation; however, this is not the default state or main application of the UNS.

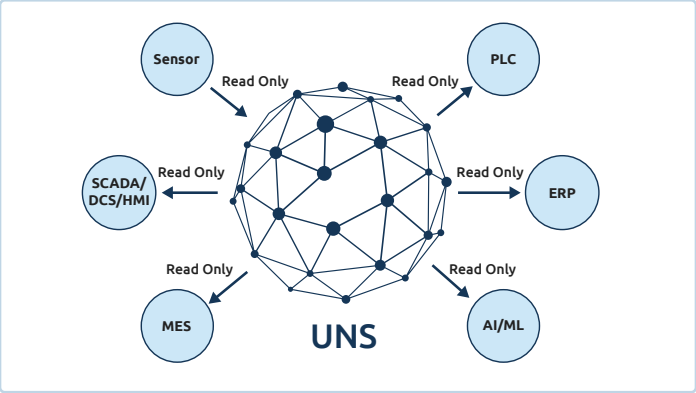

Figure 8: Data flow from Plant Floor sensors across UNS

The **second** key requirement of the UNS is that data connections are governed by "report by exception." Data is only sent to the broker if the value at the source has changed. This is different from the popular communication architecture known as "poll/response."

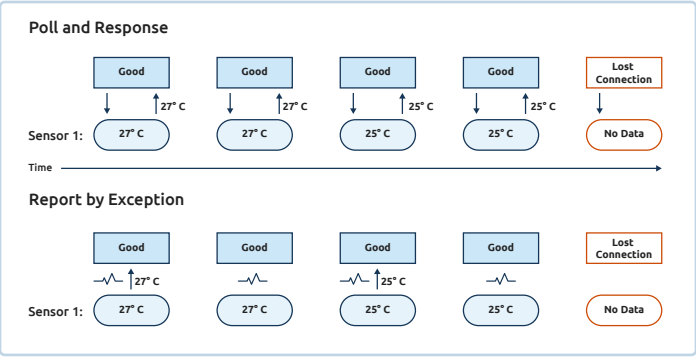

Figure 9: Comparison of Poll and Response to Report by Exception with heartbeat monitoring.

In poll/response, the reader periodically sends a read request and receives a periodic response. The difference is that report by exception produces significantly less traffic to accomplish the same transfer of useful data. Instead of transferring all data, and then performing a "merge and purge" of data with the central source to update records, only records that require change get sent to the broker.

MQTT is a data protocol that meets the PUB/SUB report by exception requirement. Part of this key requirement is to include connection status monitoring. Should a connection be disrupted, a notification would need to be published from the broker indicating the lost connection.

MQTT uses heartbeat monitoring to address this need. Both broker and client send a small ping request periodically to confirm the connection. If no response is received from either side, the broker is notified of the connection loss. This ping request is magnitudes smaller than a normal message, which allows for minimal bandwidth consumption.

With standardization of MQTT architecture, this can be implemented as part of native software functionality, rather than depending on user configuration to implement the handshake.

The **third** key requirement is that the UNS uses a lightweight protocol for data communication. Standards are the rules for how data is packaged and transported between computers; they usually have optional rules for flexibility. A protocol is the agreed implementation of a standard, which chooses what parts of the standard to use. A lightweight communication protocol has minimal rules and results in a small rate of data to be sent over the physical network. A heavyweight protocol offers more complex features but sends more data over the network. As a result, the implementation of heavyweight protocols requires more expensive network infrastructure.

A common real-world example of this concept is online video games. Playing a game with low resolution such as 30 frames per second places low demand on the hardware because less data must be transferred over the network and processed on the server. However, to play the same game at a resolution of 240 frames per second requires a phenomenal internet connection and a high-caliber computer system to handle the heavy amounts of data being processed.

The goal of the UNS is to connect every datapoint across the business. The only feasible way for a network to handle this massive amount of data is to use a lightweight message protocol.

The fourth and final key requirement is that the UNS and all nodes connected to it are designed utilizing open architecture. Open architecture is a structure that maximizes compatibility between systems. Most 3.0 factories have been developed on top of a closed architecture made up of proprietary software used in preferred provider stacks.

The main issue with closed architecture is that it restricts what data points are exposed. For example, many PLCs can only communicate equipment values to a vendor-specific control system. This prevents implementation of best in class solutions. Often a 3.0 factory will be pressured into buying a sub-optimal preferred vendor solution because it's the only one compatible with a system already in use. An open architecture-based UNS system provides flexibility to use the best software at every layer without such restriction.

Tying these requirements together results in a Unified Namespace architecture that reflects all data in the business in real time. That data is selectively distributed based on the needs of the subscriber. The UNS uses the ISA-95 standard to structure all topics containing published data points. The point of the UNS is to be interoperable. The ISA-95 standard is commonly used in DCS/MES/ERP off-the-shelf systems already, which makes integrating those systems into a UNS easier. This standard defines a hierarchical model of a business as "Enterprise/Site/Area/Work Centers/Production Line/Unit."[4] Furthermore, this standard provides a structure to allow a business to combine multiple factories into one enterprise UNS.

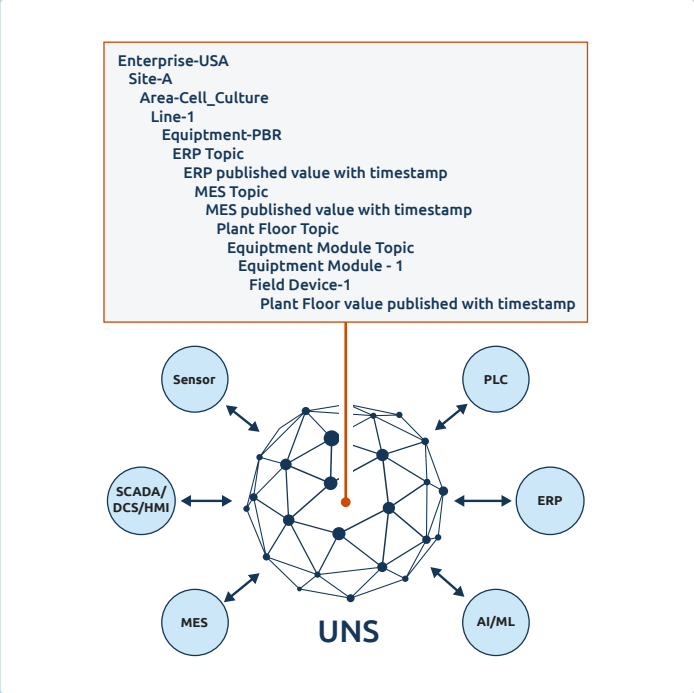

Figure 10: Example of data located within a UNS using ISA-95 hierarchy.

HOW THE UNS ARCHITECTURE IS AVENTAGEOUS

First, the UNS is designed to scale easily, reducing the time and money required to expand the business. Efficiencies in scale are created because the pathway to access data is only configured once between source and broker. Rework or additional discrete connections to other systems as the system grows are not required in a UNS architecture.

If a node needs to be connected to multiple data points in the broker, it does not require configuring any additional discrete connections. It only requires subscribing to those topics from the publishing entity.

In a traditional 3.0 architecture, data does not flow directly between a PLC and an ERP system. Rather, to get data from a PLC to an ERP system, PLC data is mapped to a SCADA system, then an MES system, and from there, mapped to an ERP system. This requires creation of thousands of discrete connections.

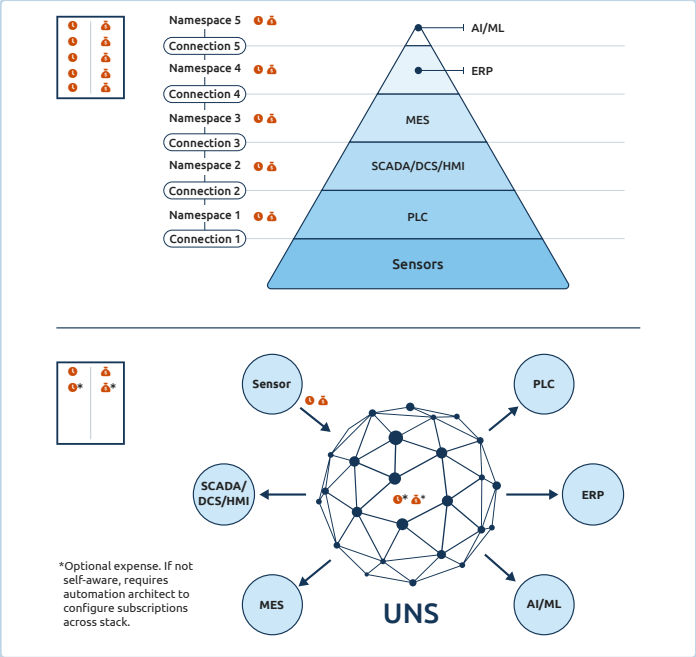

Figure 11: Simplified comparison of the time and money required to add an identical line to a 3.0 vs 4.0 factory.

In the UNS, if the ERP needs plant floor data, all that's required is to subscribe to the plant floor topic. No additional discrete connections are necessary.

Second, the UNS has the capacity to become a self-aware ecosystem. The UNS can monitor for any new published data. Upon detection, scripts that automatically configure the connections and design can trigger, based on the new data points. For example, an open-source SCADA system can auto-generate HMI displays for equipment the moment the tags are connected to the UNS. The extent of system self-awareness depends on the openness of the software utilized with the UNS. A self-aware ecosystem drastically reduces the amount of time and money needed to add new equipment to a facility, since the typical 3.0 automation connections are manually configured by engineers at present.

Third, UNS architecture can be compliant with 21 CFR Part 11 standards for Data Integrity in the Life Sciences. Software used in UNS architecture has security settings, access restrictions based on user roles, and audit tracking capabilities.

The technology necessary to implement a UNS compliantly in the Life Sciences exists today. There is already a precedent for the fundamental UNS structure discussed in this paper.

Most digitization strategies fail because they do not address the data access problem inherent to 3.0 architecture, whereas a UNS does

For all the promise that Industry 4.0 offers Life Sciences, 4.0 tools are only as good as the data they read. The problem facing 3.0 facilities is the inability to access this data. A survey conducted by Forrester of 1,805 business intelligence data users estimated that up to 73% of generated data is not utilized.[5] Increasing this data interoperability must be the first step for any successful digitization strategy. For example, implementing an AI program that can only access 30% of business data will perform significantly worse than the exact same AI program at a UNS factory that grants >80% access.

4.0 is about data-driven choices. Tools require data connections, which become expensive to make discretely. FDA-funded research indicates price as one of the biggest obstacles to implementing 4.0 technology.[6] A big chunk of the perceived cost of a 4.0 implementation may be alleviated by investing in UNS infrastructure that minimizes the amount of discrete connections to be engineered and validated.

UTILIZING UNS IN LIFE SCIENCES

Although any business can benefit from Industry 4.0, it is particularly urgent that Life Sciences embrace UNS architecture. The COVID-19 pandemic exposed how the lack of real-time data can lead to poor visibility of market demand and production supply chain disruptions. When these disruptions cause drug shortages, patients' health is put at risk.[7]

The FDA recognizes that the key to improving the supply chain resilience is with data-driven technologies such as machine learning and AI.[8] This means access to accurate information across different business layers becomes critical. As explained earlier, a UNS provides a way for accessing relevant data securely across the stack. This data is normalized, uniquely identified, and traceable as mandated per 21 CFR Part 11. As each asset of the plant is uniquely identified in the namespace, this essentially creates a digital twin of the physical system. Data produced by such systems can then be utilized effectively by AI and Machine Learning (ML) algorithms to improve the efficiency of decision-making processes in real time.

The UNS approach to data normalization is a successful way for Life Sciences to enter Industry 4.0. Of course, this strategy still comes with its own set of challenges. The next paper in Skellig's Industry 4.0 series will both discuss how to implement a UNS and identify possible obstacles in doing so effectively.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Walker Reynolds. Walker Reynolds is the CEO and founder of 4.0 Solutions, and IIOT University. He has pioneered UNS implementation for manufacturing and played an influential role in growing the 4.0 movement. The technical foundation of the UNS guidelines discussed in this paper leverages the teachings of Walker Reynolds. However, any extrapolation to the life sciences is based off FDA guidelines, ISPE trends, and years of firsthand experience working as a systems integrator and site support within the industry. 4.0 Solutions did not fund or author this paper and are not responsible for the content or claims.

CITATIONS (APA)

1. FDA. (2016, April). Data integrity and compliance with CGMP guidance for industry. Data Integrity and Compliance With CGMP Guidance for Industry. FDA. Retrieved March 10, 2022, from https://www.fda.gov/files/drugs/published/Data-Integrity-and-Compliance-With-Current-Good-Manufacturing-Practice-Guidance-for-Industry.pdf.

2. Getting Ready for Pharma 4.0TM. (2022, October 24). ISPE | International Society for Pharmaceutical Engineering. https://ispe.org/pharmaceutical-engineering/september-october-2018/getting-ready-pharma-40tm.

3. 4.0 Solutions. (2020, November 6). Why your Industry 4.0 Applications are NOT scaling. YouTube. https://www.youtube.com/watch?v=8psBOqInulc.

4. Instrument Society of America. (2000). Equipment hierarchy model. In ANSI/ISA–S95.00.01–2000 Enterprise-Control System Integration Part 1: Models and Terminology (pp. 18–25). essay.

5. Gualtieri, M., Yuhanna, N., Kisker,Ph.D, H., Curran, R., Purcell, B., Christakis, S., Warrier, S., & Izzi, M. (2016). (rep.). The Forrester Wave™: Big Data Hadoop Distributions, Q1 2016 Five Top Vendors Have Significantly Improved Their Offerings (pp. 1–13). Cambridge, MA: Forrester Research, Inc.

6. Department of Health and Human Services, Reed, D., Pierson, P., Del Sesto, T., Hourigan, M., Jeffers, M., Sommer, S., Dove, M. C., & Faiga, B., Analysis of the Advantages of and Barriers to Adoption of Smart Manufacturing for Medical Products – Focus on Response to Emerging and Pandemic Threats such as SARS-CoV-2 MxD 20-19-01 – FDA OCET Project - Executive Summary (2021). FDA/MXD. Retrieved March 10, 2022, from https://www.fda.gov/media/152569/download.

7. Ventola C. L. (2011). The drug shortage crisis in the United States: causes, impact, and management strategies. P & T : a peer-reviewed journal for formulary management, 36(11), 740–757.

8. FDA. (2021, January 15). Mou 225-21-006 | FDA. FDA. Retrieved February 4, 2022, from https://www.fda.gov/about-fda/domestic-mous/mou-225-21-006.