Published in final edited form as:

CIRP Ann Manuf Technol. 2018 ; 67: . doi:10.1016/j.cirp.2018.04.039.

A standards-based approach for linking as-planned to as-fabricated product data

Moneer Helu^a, Alex Joseph^b, and Thomas Hedberg Jr.^a

^aEngineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899 USA

^bVIMANA, Chennai 600 059, Tamil Nadu, India

Abstract

The digital thread links disparate systems across the product lifecycle to support data curation and information cultivation and enable data-driven applications, e.g., digital twin. Realizing the digital thread requires the integration of semantically-rich, open standards to facilitate the dynamic creation of context based on multiple viewpoints. This research develops such an approach to link as-planned (ISO 6983) to as-fabricated (MTConnect) product data using dynamic time warping. Applying this approach to a production part enabled the designer to make a more optimal decision from the perspective of the product lifecycle that would have otherwise been challenging to identify.

Keywords

Integration; Lifecycle; Digital thread

1. Introduction

Evolving industry challenges created by the increasingly distributed nature and growing complexity of modern manufacturing systems have driven the related concepts of Smart Manufacturing, Industry 4.0, and Cyber-Physical Production Systems [1]. The ultimate application of these concepts is a fully automated manufacturing system capable of flexibly producing lots of batch size one with little human interaction by measuring the current status of accessible capabilities [2–4]. Movement towards such applications has spurred the growth of standards and technologies that provide manufacturers with opportunities to leverage data to reduce cost, improve productivity, ensure first-pass success, and augment existing workforce capabilities [1],

The Digital Twin has been discussed as a key enabler of Smart Manufacturing, Industry 4.0, and Cyber-Physical Production Systems [2–7]. For example, the Reference Architectural Model Industry 4.0 defines an “administration shell” (which is a digital twin) as the interface that digitizes a physical asset [8]. The digital twin is a digital representation of a physical asset that can be used to describe the asset’s properties, condition, and behavior through modeling, analysis, and simulation [2–7]. Thus, a fully automated manufacturing system can collect data and information about its operation to update its digital twin to simulate the

effects of any decision to be made [3]. Similarly, the digital twin has been proposed for production planning, scheduling, routing, and control [3–4,9]; design of digitized manufacturing systems [4]; management of geometric variation and digital geometric assurance [6–7]; and design and operation of reconfigurable smart products [5]. Each of these applications relies on an underlying infrastructure to link data, information, and models from various systems across the product lifecycle to generate a holistic perspective of an asset; this infrastructure is the Digital Thread.

The Digital Thread links disparate systems across the product lifecycle and throughout the supply chain [10]. It supports data curation and information cultivation by linking data and information based on the context required for a specific use case. Hedberg et al. [10] describe a lifecycle information framework that explains how linked, contextualized, certified, and traceable data provided by the digital thread enables data-driven applications such as the digital twin. The challenge has been the inherent difficulty of aggregating and applying context to various types of data in different formats stored using different means in different locations [1,10]. Helu et al. [1] proposed and implemented a reference architecture to integrate heterogeneous manufacturing systems for the digital thread. The goal of this paper is to build on this work by using the implemented reference architecture to develop a technical approach to link as-planned to as-executed product data that may be used for the digital twin and other data-driven applications of the digital thread.

2. Background

Product lifecycle management (PLM) vendors offer solutions to manage data from different product lifecycle systems, but these solutions are typically expensive and lock users into homogeneous platforms, which limits their use by many manufacturers [1,10]. These solutions also tend to focus on the needs of engineering and thus do not entirely address the “silo effect” that has limited knowledge sharing between various functions across the product lifecycle [1,4,10]. The silo effect occurs with PLM because these tools are built to solve specific problems based on one viewpoint [10–11]. This limits the ability to build knowledge from the information of others to drive improvements based on the collective wisdom of the entire product lifecycle. Regli et al. [11] discuss how supporting multiple viewpoints leads to innovative (and often unexplored) solution spaces. Thus, managing the digital thread does not require one model or closed platform, but rather a means to dynamically link different data, information, and models to support multiple viewpoints for different applications [3,6,10–11]. The resulting approach must be based on the meaning of the data, information, and models with some associated uncertainty so that the established links can support appropriate context based on the viewpoint required for each relevant application [3,10–11].

Recent research has begun to address the need to link the semantics of data, information, and models across the product lifecycle through the digital twin. For example, Schleich et al. [6] propose a reference model for a digital twin based on concepts from geometrical product specifications, specifically skin model shapes, to manage geometric variation. Similarly, Soderberg et al. [7] discuss the functionality and requirements to develop a digital twin for real-time geometry assurance. Rosen et al. [3] discuss applications of the digital twin to

autonomous scheduling, routing, and fault handling. Stark et al. [4] present an approach to design digitized manufacturing systems that use the digital twin to control operations autonomously. The Operate, Orchestrate, and Originate (O3) Project funded by the Digital Manufacturing and Design Innovation Institute (DMDII) in the United States has developed a standards-based digital-twin approach to determine the conformance of manufacturing processes based on inspection results. Abramovici et al. [5] focus on the management and use of product lifecycle data within the digital twin to support the reconfiguration of smart products during use. In each of these examples, the focus on the digital twin highlights how current efforts develop solutions for specific problems based on one viewpoint rather than semantically linking data, information, and models to generate context for multiple viewpoints.

Much of the work focused on linking the planning and execution of manufacturing process has been in process simulation. For example, Altintas and Tulsyan [12] develop a model that can predict the cycle time to machine a part with 95% accuracy, which highlights the ability of planners to capture the behavior of process execution during planning. Altintas et al. [13] provide other examples in a thorough review of virtual machining process simulation. Despite the power of these tools, they are difficult to apply to address the silo effect between different functions and organizations across the product lifecycle because they provide one context based on one viewpoint. Because it does not enable the dynamic generation of context based on multiple viewpoints, even the best process simulation models do not necessarily help designers understand the limitations of processes or planners and machine operators understand design intent. To compliment simulation and support optimal decision making across the lifecycle, we need a means to align and contextualize as-planned and as-executed process data dynamically using semantically-rich, open standards. Such an approach enables the development of the digital thread (and subsequently the digital twin) by promoting technology-agnostic solutions to integrate heterogeneous production systems. We develop such an approach in this research.

3. Technical Approach

Figure 1 provides an overview of our technical approach. All data is collected from the NIST Smart Manufacturing Systems (SMS) Test Bed as described by Helu et al. [1]. The collected data complies with broadly-accepted, semantically-rich, open standards: ISO 10303 (STEP), ISO 6983 (G code), and MTConnect for the as-designed, as-planned, and as-executed data, respectively [1]. We transform each of these datasets into the computer-aided design (CAD) representation for the as-designed state, the expected execution for the as-planned state, and the measured execution for the as-executed state of the part. The expected execution is mapped to the measured execution and the result overlaid onto the CAD representation of the part for visualization.

To facilitate the technical approach, we developed an R package, called *mtconnectR*, to read, parse, store, and visualize G-code and MTConnect data [14]. We included all functions needed to import the as-planned and as-executed data into a common analysis environment where we can transform the as-planned data to the expected execution and map it to the measured execution.

3.1. Generating Expected Execution Dataset

We first parsed the G-code data to generate the expected execution. Several tools exist for reading and parsing G code, but no such tools were available for R. To do so natively in R, we created a G-code library that maps G codes, M codes, and other relevant letter addresses to the operation type and subtype being commanded during execution. Only commands that correspond to operations that were to be analysed were included in the library (e.g., G00, G01, G02, M03). All remaining portions of the G-code data were neglected in the analysis. Joseph et al. [14] contains further information about the composition of the G-code library.

G code is sequentially executed line-by-line by the machine-tool controller, and multiple G-code commands can exist in a single line, in which case the commands are executed based on a prescribed order of execution. Thus, it is important to preserve the order of execution when establishing the sequential structure of the G-code data. The G-code library contains a priority value so that each line of G-code data can be read and rearranged according to the prescribed order of execution. The result is a parsed G-code table that defines the full sequence of commands executed by the machine tool controller, which can then be used to create a dataset that defines the expected execution.

To initialize the simulation, we defined the starting state of the machine tool by setting the Position, Linear Velocity, and Rotary Velocity for each axis and Path Feedrate to 0. The initial Timestamp was 1970-01-01 00:00:00 and the Program and Tool Number were Unknown. Once the initial and final state of the machine for each command were determined, we calculated the time required for the machine tool to execute the command using the path feedrate and the distance traveled based on position. This time established the timestamp for each execution event. Timestamps were also interpolated between execution events based on the sampling resolution needed to link the expected and measured execution datasets. Other relevant data items were determined for each execution event in a similar fashion. The resulting parsed dataset that defined the expected execution including timestamps was then ready to be mapped to the measured execution dataset.

3.2. Preparing Measured Execution Dataset

The as-executed data was collected from an MTConnect Agent as an ASCII log file using a Ruby parser provided as an open-source tool with the reference MTConnect implementation. The log file was a continuous stream of data that included different operations combined into one object. To divide this data into constituent operations with the corresponding semantics that link the data to the equipment that provided it, we created a function in the mtconnectR package to map each data item from the log file to the metadata defined by the MTConnect Devices XML for the machine tool. The MTConnect Devices XML structures and labels the data provided by a piece of equipment based on the MTConnect specification and common vocabulary [15]. The resulting parsed and contextualized data representing the measured execution was then mapped from the expected execution dataset.

3.3. Mapping Expected to Measured Execution

Differences between the expected and measured execution may be due to errors during execution (e.g., incorrectly defined offsets) or discrepancies that are often not understood by

planners (e.g., the machine may not be able to provide the required capability). Differences may also occur because the physics of the machine tool may not be fully modeled (e.g., idling and tool changes were not included in this analysis). However, as we described in Section 2, our goal is not to develop an extremely accurate process model - highly-capable process models exist in the literature [13] - but rather to understand when machine dynamics or other impacts may strongly affect the execution of a planned process. Thus, it is important to create a robust mapping between the timestamps in the expected and measured execution data to identify when discrepancies may highlight opportunities for optimization from the perspective of product lifecycle engineering.

Because we expected that discrepancies between the expected and measured execution may be driven by several different causes, we decided to apply dynamic time warping (DTW) to map the two datasets (see Figure 2). DTW is an approach to create a mapping between a reference and query dataset when there may be skew in the timestamps. It is similar to applying a Euclidean distance function in a multi-dimensional space while relaxing constraints on the time axis such that one point in the query dataset can be skipped or mapped to multiple points in the reference dataset.

To apply DTW, we had to identify data items in the execution datasets that occur with high frequency and that vary rapidly relative to the dataset. It was these data items that we used to define the distance matrix for the mapping. Of the available data items, we found that Rotary Velocity and Position provided the best results since they tended to have minimal deviation (aside from differences in timestamp) between the expected and measured execution. To supplement these data items, we also applied a grouping variable that combined Tool Number and Execution (i.e., categorical event that reports controller status) after mapping this data to numerical values. While we did not expect Tool Number and Execution to vary rapidly, they could be followed exactly with minimal effort.

Determining which values of Rotary Velocity, Position, and Tool-Number/Execution to link when mapping the expected and measured execution was trivial to accomplish because both datasets were parsed, sequenced, and fully defined and contextualized. Calculating the distance matrix for the mapping, though, was challenging because the discrepancy in their timestamps was often non-uniform. The first step for the distance matrix calculation was to standardize the sampling frequency for both datasets through interpolation. The sampling frequency had to be at least twice the frequency of the smallest event expected (i.e., Nyquist rate) but not so high as to need unnecessarily high computational power and time for the mapping. We used the same sampling frequency for both datasets since DTW natively matches indices rather than timestamps. Next, we normalized the values of Rotary Velocity and Position so that neither was given undue weighting when calculating the distance matrix. This was needed because the variation in the two data items differed substantially: Rotary Velocity may vary by $\mathcal{O}(100 \text{ RPM})$ while Position may vary by $\mathcal{O}(0.001 \text{ m})$ or less. To normalize the data items, we apply a standard score value to the expected and measured execution.

Using the prepared expected and measured execution datasets, we determined the distance matrix for the DTW. To avoid unbounded calculations, we applied a step-pattern approach,

which locally constrains the values used by an algorithm as it traverses a dataset, with a minimum variance matching algorithm. This approach enabled the distance calculation to exclude arbitrarily long stretches of the reference dataset. We needed this capability since the expected execution did not contain all data items included in the measured execution. To ensure that we excluded only the appropriate data, we applied an elasticity argument that set the maximum run length of excluded data to be at least as long as the length of the longest event that may be missing from the expected execution dataset.

Using the calculated distance matrix, we applied the DTW methodology to map the expected to the measured execution. We then overlay the result onto a CAD representation of the part to support visualization for different viewpoints.

4. Preliminary Verification and Validation

We applied the approach in Section 3 for the case study in Feng et al. [16] (see Figure 3). This part was provided by an industrial partner and modeled using Siemens NX. MasterCAM was used to plan and generate the G code for the process used to machine the part, which was performed on a GF MIKRON HPM600U five-axis machining center that is part of the NIST SMS Test Bed. To calculate the distance matrix, we used a sampling frequency of 10Hz to standardize the two execution datasets. The elasticity was set to 200 since tool changes and rapids were not simulated and were measured to occur over 20 sec periods.

Figure 4 highlights the mapping between the expected and measured X-position data for three features of the part (A, B, and C). These results show how providing additional context relevant for several viewpoints can improve decision making across the product lifecycle [16]. The difference between the plots in Figure 4 may be obvious for a trained expert with manufacturing domain knowledge to generate and understand, but non-experts may find such an analysis non-trivial. The proposed approach allows for the dynamic linking of as-designed, as-planned, and as-executed datasets, which enables the automation of data analysis and visualization regardless of expertise. Further, this approach builds the context needed for different viewpoints to understand the implications of a decision on the entire product lifecycle.

The value of the proposed approach can be seen by using it to analyze the relatively large discrepancy that occurred between the as-planned and as-executed states for feature C. By overlaying the mapping onto the toolpath (not shown) and CAD representation, we can see that the discrepancy was caused by a feedrate mismatch: the rapid zig-zag toolpath used to create the chamfer prevented the machine axes from reaching the commanded feed. While such a discrepancy may be addressed with process simulation, such tools fail to communicate the impact of machine tool dynamics on the planned process so that designers and planners may make more optimal decisions from the product lifecycle perspective. In this case, linking the data as described and shown in Figure 4 allowed the designer to see the larger impact of his design decision to base the chamfer on a legacy design not needed for the current application. While process simulation could be used to adjust the production schedule for this feature, removing the feature using an engineering change request is the

optimal solution since it was not needed for the function of the part. Without the dynamic generation of context supported by the linked data provided by the digital thread, such valuable insights are difficult to obtain.

5. Summary

The research presented in this paper enables the standards-based linking of as-planned and as-executed product data. This linking occurs by generating an expected execution dataset through parsing, sequencing, and simulating the operations commanded by the G code used to control a machine tool fabricating a part. The expected execution can then be mapped to the measured execution (generated by parsing and contextualizing the MTConnect data collected from the machine tool) using a DTW approach. The result can be overlaid onto a CAD representation of the part to generate different contexts appropriate for different viewpoints. By testing this approach using data collected during the design, planning, and fabrication of a production part, we showed how the ability to support multiple viewpoints drives more optimal decision making from the perspective of product lifecycle engineering.

The approach discussed in Section 3 has been implemented in the R programming environment through a freely available package called *mtconnectR* [14]. Our hope is that this package enables others to create solutions for the digital thread. We plan to continue our work by improving the current simulation (e.g., include events that were neglected in this initial effort, such as tool changes and idle times). This information can help identify further inefficiencies that may be addressed by influencing changes in multiple product lifecycle stages. We also plan to explore the automated generation of knowledge based on the data and information collected via the digital thread. We believe that such work will enable the promise of Smart Manufacturing, Industry 4.0, and Cyber-Physical Production Systems.

Acknowledgements and Disclaimer

The authors would like to acknowledge Allison Barnard Feeney, William Bernstein, and Athulan Vijayaraghavan for their support. The identification of commercial systems does not imply recommendation or endorsement by NIST or that the products identified are necessarily the best available for the purpose.

References

- [1]. Helu M, Hedberg T Jr., Feeney AB (2017) Reference Architecture to Integrate Heterogeneous Manufacturing Systems for the Digital Thread. *CIRP Journal Manufacturing Science and Technology* 63 (2):585–605.
- [2]. Meudt T, Metternich J, Abele E (2017) Value Stream Mapping 4.0: Holistic Examination of Value Stream and Information Logistics in Production. *CIRP Annals-Manufacturing Technology* 66(1): 413–416.
- [3]. Rosen R, von Wichert G, Lo G, Bettenhausen KD (2015) About the Importance of Autonomy and Digital Twins for the Future of Manufacturing. *IFAC-PapersOnLine* 48(3):567–572.
- [4]. Stark R, Kind S, Neumeyer S (2017) Innovations in Digital Modelling for Next Generation Manufacturing System Design. *CIRP Annals-Manufacturing Technology* 66(1):169–172.
- [5]. Abramovici M, Gobel JC, Savarino P (2017) Reconfiguration of Smart Products during their Use Phase based on Virtual Product Twins. *CIRP Annals-Manufacturing Technology* 66(1):165–168.
- [6]. Schleich B, Anwer N, Mathieu L, Wartzack S (2017) Shaping the Digital Twin for Design and Production Engineering. *CIRP Annals-Manufacturing Technology* 66(1):141–144.

- [7]. Soderberg R, Warmefjord K, Carlson JS, Lindkvist L (2017) Toward a Digital Twin for Real-Time Geometry Assurance in Individualized Production. *CIRP Annals-Manufacturing Technology* 66(1):137–140.
- [8]. IEC PAS 63088:2017 (2017) Smart Manufacturing - Reference Architecture Model Industry 4.0. Standard, International Electrochemical Commission.
- [9]. Freitag M, Becker T, Duffie NA (2015) Dynamics of Resource Sharing in Production Networks. *CIRP Annals-Manufacturing Technology* 64(1):435–438.
- [10]. Hedberg T, Feeney AB, Helu M, Camelio JA (2017) Toward a Lifecycle Information Framework and Technology in Manufacturing. *Journal of Computing and Information Science in Engineering* 17(2):021010. [PubMed: 28265224]
- [11]. Regli W, Rossignac J, Shapiro V, Srinivasan V (2016) The New Frontiers in Computational Modeling of Material Structures. *Computer-Aided Design* 77:73–85.
- [12]. Altintas Y, Tulsyan S (2015) Prediction of Part Machining Cycle Times via Virtual CNC. *CIRP Annals-Manufacturing Technology* 64(1):361–364.
- [13]. Altintas Y, Kersting P, Biermann D, Budak E, Denkena B, Lazoglu I (2014) Virtual Process Systems for Part Machining Operations. *CIRP Annals-Manufacturing Technology* 63(2):585–605.
- [14]. Joseph A, Adapa S, Madasu N, Ravishankar S (2017) CRAN-Package mtconnectR, <https://CRAN.R-project.org/package=mtconnectR>.
- [15]. MTConnect v1.3.1 (2015) Standard, MTConnect Institute.
- [16]. Feng SC, Bernstein WZ, Hedberg T, Feeney AB (2017) Toward Knowledge Management for Smart Manufacturing. *Journal of Computing and Information Science in Engineering* 17(3):031016.

Figure 1.
Overview of technical approach to link as-planned to as-executed data and overlay results on CAD representation

Figure 2.
Dynamic time warping applied to map two related datasets

Figure 3.
CAD representation of part used to test methodology [16]

Figure 4.
Comparison of expected and measured X-position for test part with data for features A, B, and C highlighted (after [16])