NIST Grant/Contractor Report
NIST GCR 24-050

Spatial and Temporal Data Alignment from Disparate Sources for Feature Association

Benjamin Standfield
Eric Holterman, P.E.
Russell Waddell

This publication is available free of charge from:
https://doi.org/10.6028/NIST.GCR.24-050

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Grant/Contractor Report
NIST GCR 24-050

Spatial and Temporal Data Alignment from Disparate Sources for Feature Association

Benjamin Standfield
Digital Systems Researcher – CCAM

Eric Holterman, P.E.
Principal Systems Engineer - CCAM

Russell Waddell
Digital Systems Consultant – CCAM

This publication is available free of charge from:
https://doi.org/10.6028/NIST.GCR.24-050

April 2024

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

NIST GCR 24-050
April 2024

Disclaimer
This publication was produced as part of cooperative agreement 70NANB21H158 with the National Institute of Standards and Technology. The contents of this publication do not necessarily reflect the views or policies of the National Institute of Standards and Technology or the US Government.

Certain commercial equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

How to Cite this NIST Technical Series Publication
Standfield B, Holterman E, Waddell R (2024) Spatial and Temporal Data Alignment from Disparate Sources for Feature Association. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Grant/Contractor Report (GCR) NIST GCR 24-050. https://doi.org/10.6028/NIST.GCR.24-050

Abstract

Among the chief topic areas of Industry 4.0, digital thread technology offers the opportunity for increased productivity, efficiency, and traceability throughout a product's lifecycle. While the basic concept of digital thread is easy to realize by specifying aggregation of data from multiple stages of an entity's life cycle, digital threads must contain internal representations and associations of data across the life cycle to bring about improved traceability features and promised productivity. In this work, we bring missing pieces into digital threads to realize future production, traceability, and efficiency-increasing capabilities within the design, planning, execution, and quality stages of discrete part manufacturing. Namely, we (1) developed a standards-based digital thread using standards commonly found in the industry, specifically STEP AP242, unstandardized NC, MTConnect, and QIF standards, (2) perform spatial alignment within the four stages to geometries of interest defined through STEP AP242 shape aspects, and (3) perform temporal alignment of execution data by identifying ISO 8061 timestamp intervals where the geometric features were actively worked on. With this work, we hope to influence ongoing standards development in the four previously mentioned standards and standardize the development of future digital threads, such as ISO 23247, by providing a standards-based methodology that allows for internal association of the data within the thread.

Keywords

Industry 4.0; QIF; MTConnect; STEP AP242; Digital Thread; Spatial Alignment; Temporal Alignment; Manufacturing Life Cycle.

Table of Contents

List of Figures

Acknowledgments

CCAM would like to thank NIST, Mitutoyo, Hurco, Cisco, Metalogi, and nVariate for contributing to this work. Additionally, CCAM would like to thank Origin International for their aid in converting the CMM results to QIF 3.0 format.

Executive Summary

1.1. Introduction

- Digital thread is an emergent approach to enhance process and product analytics by connecting data such as designs, production plans, execution logs, and inspection results throughout the lifecycles of products and processes.
- Because the data that make up a digital thread are sourced from heterogeneous equipment, systems, and software, it is critically important that time and space components of the data be synced and accurate across data sources.
- In this project, CCAM constructed a spatiotemporally-aligned standards-based digital thread from the manufacturing process lifecycle stages: as-designed, as-planned, as-executed, and as-inspected.
- This implementation establishes a foundational digital thread. This foundational thread can be expanded horizontally by adding more data sources to each lifecycle stage or vertically by adding additional stages.
- Alternatively, this work is a prerequisite for digital thread-based process analytics. The thread contains a facsimile of part production and can, therefore, be interrogated to identify process errors and inefficiencies.

1.2. Methodology

- The digital thread developed in this project targets the design, planning, execution, and inspection stages and is based on existing industry standards. The work uses a web server for coordination and computation, a database for storing document data (namely shape aspect definitions and occurrences), and a database for storing the time series as-executed data.
- STEP AP242 represents the as-designed stage due to its support for marking geometries of interest through shape aspects. The NC data represents the as-planned stages and was selected based on its ubiquity rather than conformance to any particular manufacturing standard. MTConnect represents the as-executed stage due to its growing popularity and support for the NC line number and program name. The QIF 3.0 format represents the inspection data due to its strong support among CMM manufacturers.
- The thread begins with the as-planned data in the form of a STEP AP242 file. The file is uploaded to the spatiotemporal feature (SF) API, and shape aspects are extracted and assigned RFC4122 version 5 UUIDs based on the entity ID of the shape aspect within the file and file name. This creates a shape aspect definition for each shape aspect in the STEP AP242 file.
- The as-planned data is added to the digital thread when an NC file is uploaded to the SF API. Subsequently, the NC file is simulated to correlate NC lines and shape aspects. This

generates an annotated NC file saved to the SF API. The NC line numbers are also appended to their respective shape aspect definitions as start-stop pairs.
- The as-executed data can be added in two ways, online and offline. The offline method takes an MTConnect sample file and observes which program is running, the machine state, and the current NC line number to create shape aspect occurrences and translate the NC line start-stop pairs to start-stop timestamp pairs in which all data from a machine belongs to a set of shape aspects. The online extraction follows a similar process, but the URL of the MTConnect agent is uploaded to the server instead of an MTConnect sample file.
- The as-inspected data is aligned using the QIF plan file. Within the QIF plan file, the QIF features and relevant data, such as feature type and internal UUID, are aligned to the nominal geometry within the STEP AP242 file and saved to the shape aspect definition. The characteristic item ID is noted for aligning the QIF results to the shape aspect occurrences.
- An alternative methodology is presented that constructs a watertight model from the STEP AP242 design model and builds the thread based on the execution data rather than the planning data.

1.3. Results

The major achievements of this project include:
- The construction of a spatiotemporally-aligned standards-based digital thread.
- Standards integration between STEP AP242, unstandardized NC, MTConnect, and QIF to describe disparate geometries within an object.
- Instantiation of as-planned and as-executed data into the digital thread.

1.4. Conclusions and Recommendations

- CCAM is in an ideal position to continue digital thread research by either (1) horizontally scaling the thread to include more standards or (2) performing analytics based on the spatially aligned digital thread.
- A standards-based digital thread can be used for spatiotemporal alignment of data sourced from disparate devices.
- Geometry and process data from across the manufacturing process lifecycle stages can be cohesively merged ex-post, even where spatiotemporal data integration was not explicitly designed into equipment, software, or systems. The demonstrated approach does, however, require support for the referenced standards in place before the thread is assembled.

2. Introduction and Background

Digital thread is a pinnacle technology of the fourth industrial revolution, offering consolidated and cohesive access to part and process data from across the entire lifecycle [1]. A digital thread opens the door for causal analytics, data-informed process and design changes, efficiency gains, and process improvement by linking as-designed, as-planned, as-executed, and as-inspected lifecycle data with part data in production. Significant research efforts are dedicated to implementing and determining best practices for creating, maintaining, and using digital threads, and published approaches remain far-reaching and diverse [2], [3], [4]. One common goal is identifying, building, and testing various technology stacks to link the data making up a “thread.” Another unifying theme among modern implementations of digital threads is to collect data and assign it to part lifecycle stages. One example utilizes STEP [5], NC, MTConnect [6], and QIF [7] formatted data as the as-designed, as-planned, as-executed, and as-inspected data, respectively [8].

- Digital thread is an emergent approach to enhance process and product analytics by tying a single “thread” between life cycle data such as designs, production plans, execution logs, and inspection results [9].
- Because the data that make up a digital thread are sourced from heterogeneous equipment, systems, and software, it is critically important that time and space components of the data be synced and accurate across data sources.
- In this project, CCAM constructed a spatiotemporally-aligned standards-based digital thread from the manufacturing process lifecycle stages: as-designed, as-planned, as-executed, and as-inspected [10].
- This implementation establishes a foundational digital thread. This thread can be expanded horizontally by adding more data sources to each lifecycle stage and vertically by adding additional stages.
- Alternatively, this work is a prerequisite for digital thread-based process analytics. The thread contains a facsimile of part production and can, therefore, be interrogated to identify process errors and inefficiencies.

Despite advances in research and best practices over the past decade, many issues have yet to be addressed. This includes the communication of digital threads [9], the alignment of multiple data sources allocated to a single lifecycle stage [10], and the spatial alignment of data to “features” within a part’s geometry. These objectives can be addressed by introducing a standards-based digital thread, employing temporal alignment techniques to data streams, and using features from the as-designed geometry as the foundation for spatial alignment of subsequent lifecycle stages.

CCAM developed a spatiotemporally aligned digital thread for this project using the Flexible Manufacturing Cell equipped with a Hurco VM10i milling machine and a Mitutoyo MiSTAR 555 CMM. Specifically, CCAM evaluated the following:

Task 1 – IEEE 802.1AS Time Synchronization for temporal alignment of manufacturing devices connected to ethernet networks.
Task 2 – Machine learning-based alignment of disparate sensor streams.

Task 3 – Standards-based spatiotemporally aligned digital thread.

The presented standards-based spatiotemporally aligned digital thread is intended to serve as the foundation for additional digital thread research by involving additional product life cycle stages and including new standards. However, this work can also be expanded to include advanced analytics using the contained manufacturing lifecycle data for causal analytics or process improvement.

3. Methodology

Spatiotemporal alignment can be split into spatial and temporal data components to produce a coherent and practicable digital thread. The standards-based approach is carried throughout and intended to allow scalability and portability beyond the target equipment and devices represented in this work. Spatial alignment means matching collected data to geometric features via applicable standards; temporal alignment fits multiple discrete sensor streams into the single digital thread and targeted IEEE 802.1AS time synchronization protocol for microsecond-level alignment.

3.1. Network-based Time Synchronization with IEEE 802.1AS and TSN

Time-Sensitive Networking (TSN) is a major area of applied research and commercial development in the industrial and manufacturing sectors intended to provide low latency and jitter via a set of sub-standards under the IEEE 802.1 Ethernet standard. These features are important for automation and control tasks and, in the case of the digital thread, potentially enable very rapid feedback and adjustment of industrial processes, safety-critical systems, and other deterministic applications.

Real-time synchronous data transmission over Ethernet in IEEE 802 dates to at least 2008 with the Audio and Video Bridging Task Group [11] and a 2012 renaming as Time-Sensitive Networking Task Group. National Instruments deployed their TSN testbed for industrial Internet of Things (IIoT) devices in 2018 [12]. Follow-up deployments in industrial settings have not been widely published. Still, microsecond-level alignment remains appealing, where discrete sensor data can be blended with other sources, such as digital thread.

3.2. ML-based Temporal Alignment of Disparate Data Streams

Data sources for industrial and manufacturing systems are generally internally time-synched by design. Most contemporary production equipment will include multiple clocks and timers, and for basic operation and function, temporal alignment across multiple internal sources is required. Building a digital thread, however, draws data from multiple discrete systems or systems-of-systems that are not externally time-synched and require an additional layer of temporal alignment.

During the as-executed stage of the manufacturing life cycle, two ML approaches are considered to align disparate data streams: 1) change point detection (CPD) [13] and 2) dynamic time warping (DTW) [14]. The targeted sensor streams include an MTConnect stream from the machine tool's built-in MTConnect agent, a dynamometer attached to the spindle, and an acoustic emissions sensor mounted within the machine tool's workspace.

3.3. Spatiotemporal Feature API

An application programming interface (API) is a tool to enable communication between disparate systems. In this case, a dedicated mechanism is needed to connect to the disparate sources, carry data, merge data, and enable the creation of the digital thread in a database. That mechanism is the spatiotemporal feature (SF) API, which is custom-built for this purpose. Approaches to generalize such an API are addressed in recommendations for future work below.

The SF API links data originating from various phases of the manufacturing lifecycle. The SF API is connected to MongoDB and InfluxDB, as illustrated in Figure 1. MongoDB stores the links between lifecycle data. InfluxDB stores time series data from the as-executed phase. MongoDB stores two types of documents: shape aspect definitions and shape aspect occurrences. A shape aspect definition is the nominal shape aspect defined by the STEP AP242 file and contains data from the STEP AP242 file and its corresponding NC and QIF Plan files. The shape aspect definition, as stored in MongoDB, holds the UUID assigned to the shape aspect, references to the STEP, NC, annotated NC, and QIF Plan files, block numbers from the NC file associated with the shape aspect, and characteristics from the QIF Plan associated with the shape aspect. In addition to storing the QIF Plan file, the features within the QIF Plan are rendered and associated with shape aspects defined in the STEP AP242 file. After determining the correlated shape aspect for each QIF feature, the QIF characteristics are attached to the shape aspect definition if it exists. The shape aspect occurrences document stores data relevant to a physical instance of the shape aspect. This includes references to the ShapeAspectDefinition, the UUID assigned to the part occurrence, as-executed data, and as-inspected data. As-executed data is stored as a list of start-stop timestamp pairs. Data generated from a CNC machine is related to the shape aspect occurrence within these time intervals. For each characteristic, the as-inspected data contains a pass-fail boolean and characteristic value from the QIF Results for the shape aspect and a reference to the QIF results file.

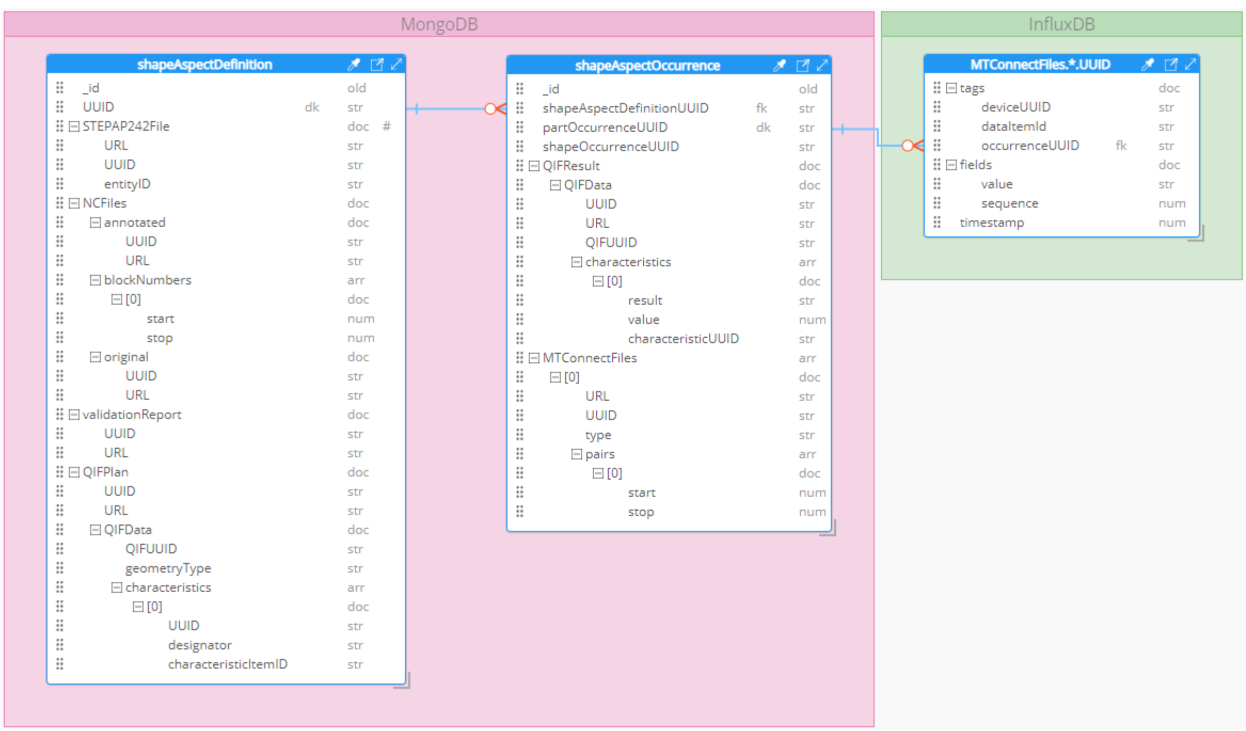

Figure 1: Entity Relationship Diagram

3.4. Digital Thread Architecture

The spatially aligned standards-based digital thread developed in this project targets four phases in the manufacturing lifecycle: as-designed, as-planned, as-executed, and as-inspected. In each stage, data alignment takes advantage of a standardized format for data within the stage, creating a stacked set of linkages across otherwise disparate data. Moreover, the same architecture for data linking would apply to any other system component that supports the named standards.

The digital thread uses a STEP AP242 file with shape aspects to align the as-designed data. Shape aspects are, for this work, synonymous with geometric features within the STEP AP242 file. NC files ("part programs") are universally used to convey instructions to computer-controlled machine tools and, in this case, used for the as-planned data. MTConnect ties as-planned to as-executed data. MTConnect supports the "Block" data item, facilitating simple alignment of MTConnect formatted data to NC Plan lines. However, it should be noted that new versions of the MTConnect standard deprecate the "Block" data item in favor of the "Line Number." Lastly, the as-inspected phase is aligned using QIF 3.0 formatted data. QIF 3.0 utilizes features and characteristics for the definition and inspection of geometries. QIF features, in turn, map one-to-one with STEP AP242 shape aspects.

The digital thread begins with the STEP AP 242 file. Within the STEP AP242 file, the shape aspects are extracted and assigned UUIDs. While no currently standardized and accepted method of embedding UUIDs in the STEP AP242 file exists, ISO 10303 working group 12 is defining exactly such a method. This work assigns a UUID to each shape to address the lack of unique identifiers within present STEP AP242 files. After the generation of the respective NC file, the motion driven by that part program is simulated in 3D space along with the shape aspects to correlate lines of NC code and shape aspects. Next, the part is machined, resulting in an MTConnect data stream. Using the Block data item from MTConnect, the as-executed data is aligned to the NC plan data. Lastly, the QIF Features are extracted from the QIF plan file and rendered along with the shape aspects extracted from the STEP AP242 file. Pairs of correlated QIF Features and STEP AP242 shape aspects are created based on the geometry types and location according to their respective files.

3.4.1. As-designed

Nothing has been aligned yet at the as-designed stage of the manufacturing lifecycle. Rather, this stage aims to initialize the shape aspects for alignment in subsequent phases and begin the thread. Each shape aspect within the STEP AP242 file is assigned a v5 RFC 4122 UUID using the entity number and file name. After extraction, a ShapeAspectDefinition is generated for each shape aspect within the STEP AP242 file. The stage concludes by storing a reference to the STEP AP242 file where the shape aspect was found.

3.4.2. As-planned

The second stage aligns each line of the generated NC plan file to any affected shape aspects in the STEP AP242 file. The as-designed geometry, the stock part, and each shape aspect are rendered in 3D space. First, the nominal geometry is subtracted from the stock part to create disparate subtraction geometries in space. Next, each shape aspect is assigned to an adjacent subtraction geometry based on the direction of the shape aspect's geometry and the geometry's normal. Lastly, the machining is simulated using the NC file and machining parameters such as tool radius. When the tool touches a subtraction geometry, the UUIDs of the shape aspects assigned to that geometry are assigned to the NC line. At the end of this procedure, an annotated NC file is generated that ends each line with a comma-delimited list of shape aspect UUIDs using NC file comments. This stage is concluded by storing the original and annotated NC file in the ShapeAspectDefinition.

3.4.3. As-executed

The as-executed stage begins by creating a ShapeAspectOccurance for each ShapeAspectDefinition. Next, The MTConnect stream is extracted from the CNC machine in situ or post-operation. Afterward, the as-executed data is aligned based on the Block and Program_running data items. The SF API monitors the MTConnect stream, and when there is a matching annotated NC file for the program running on the CNC, the SF API begins to align the as-executed data based on the UUIDs within the annotated NC file. Within the ShapeAspectOccurances, lists of timestamp pairs are stored where the first timestamp begins a data segment for the time stamp, and the second timestamp ends the data segment non-inclusively.

3.4.4. As-inspected

The inspection data in the QIF result file correlates to a matching QIF Plan file. The QIF Plan file contains the features and characteristics for inspection. Each feature defined within the QIF Plan file has an RFC 4122 UUID. Each QIF Feature and STEP AP242 shape aspect is rendered to determine the correlated pairs based on location and geometry type. The UUID assigned to each QIF Feature is saved to the matching shape aspect definition, and any QIF Characteristics associated with the QIF Feature are saved to the shape aspect definition. Within the QIF Result file, the array QIF Characteristic's pass/fail and a value for each characteristic are extracted and assigned to the respective shape aspect, concluding the construction of the digital thread.

3.5. Digital Workflow

The workflow for constructing a digital thread is split into six phases, each divided by a hand-off of the driving STEP AP242 file. Five actors execute the workflow for creating and updating the digital thread: The Network Engineer responsible for initializing the SF API, the Design Engineer responsible for creating and validating the as-designed STEP AP242, the QA engineer responsible for planning and executing the inspection, the manufacturing engineer responsible for creating the manufacturing plan, and the machinist responsible for setting up the CNC machine and machining the part. At the end of the presented BPMN diagram, a spatiotemporally aligned digital thread exists in the databases managed by the SF API.

3.5.1. Phase 1 – Network Engineering

The process begins with the network engineer. The network engineer has two responsibilities at the beginning of the workflow. (1) Ensure that the SF API is operational, and (2) configure the MTConnect data streams so that the in-situ CNC and CMM data can be collected. Configuration occurs when the MTConnect URLs of the CNC and CMM are passed to the SF API. Afterward, the streams are monitored for changes to the “Block” and “Program_running” data items in the CNC stream by the SF API.

3.5.2. Phase 2 – Design Engineering

While the network engineer configures the SF API, the design engineer can create the as-designed part. To create the part, Siemens NX 1988 was used to generate the STEP AP242 file. However, any CAD system or program capable of generating STEP AP242 files is synonymous with NX 1988 for the presented workflow. After creating the STEP AP242 file from some initial product specifications and ensuring it includes GD&T authored as semantic PMI rather than graphical PMI, as shown in Figure 2, the design engineer runs the STEP AP242 file through the NIST STEP File Analyzer. Specifically, version 5.01 of the STEP file analyzer was used, and a validation report was generated from the STEP File Analyzer. If the validation fails, the STEP AP242 file is updated and reanalyzed until it passes validation. When the Validation passes, the STEP AP242 file is handed off to the quality engineer for inspection planning, and the STEP AP242 file is uploaded to the SF API with the corresponding validation report generated from the SF API. This consequently creates the shape aspect definitions in MongoDB.

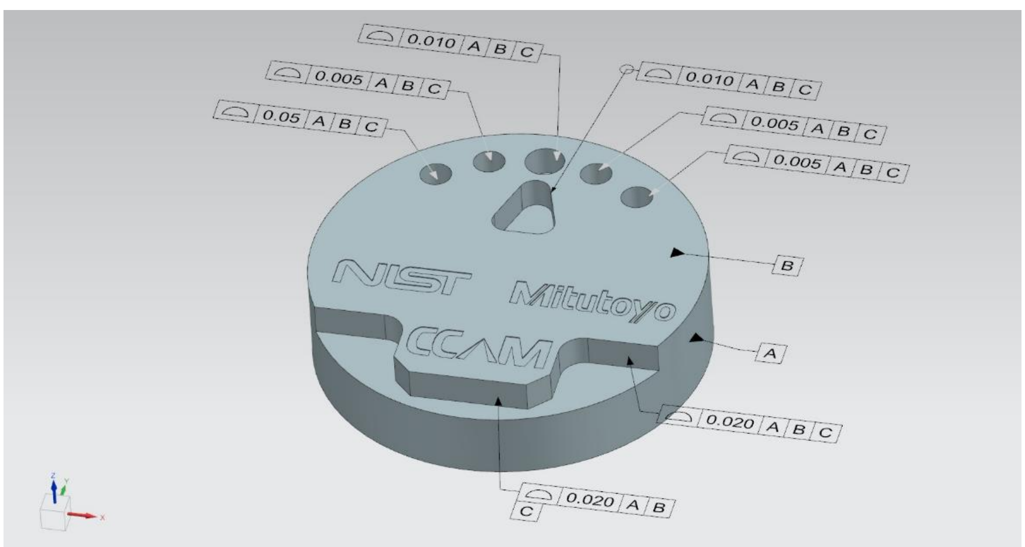

Figure 2: As-designed STEP AP242 file in NX 1988

3.5.3. Phase 3 – Quality Engineering

Phase three begins immediately after phase 2 and is executed by the quality engineer. In this phase, the quality engineer is responsible for first using Mitutoyo MiCAT Planner version 2.0 to define the measurement rules for inspecting the physical part. Next, the inspection and QIF Plan files are generated for the part using MiCAT Planner version 2.0. The measurement rules for this part use model-based GD&T (Profile) with touch point contact, resulting in an array of points for inspection, as shown in Figure 3. This process is concluded by uploading the QIF plan file to the SF API, which is then assigned to each shape aspect definition defined by the STEP AP242 file.

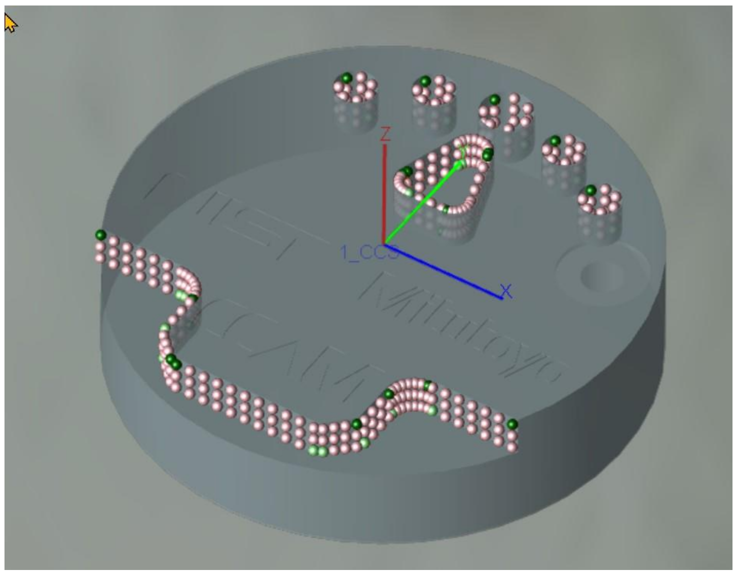

Figure 3: Nominal measure points from GD&T.

3.5.4. Phase 4 – Manufacturing Engineering

After the quality engineer generates the CMM program and QIF Plan files, the STEP AP242 file is handed off to the manufacturing engineer responsible for generating the NC plan file. After generation, the NC Plan file is uploaded to the SF API. An annotated copy of the NC file is created by simulating a machining operation using a stock part, the AP242 file, and the NC file, as shown in Figure 4. However, this annotated file is not used during the machining process. The manufacturing engineer's role is completed after the original NC file is uploaded and handed off to the machinist for the machining of the stock part.

```
N414 G01 X26.088 Y-20.158 Z.064
N415 G01 X26.069 Y-19.963 Z.054
N416 G01 X26.012 Y-19.776 Z.043
N417 G01 X25.92 Y-19.603 Z.033
N418 G01 X25.796 Y-19.451 Z.022
N419 G01 X25.644 Y-19.327 Z.012
N420 G01 X25.471 Y-19.234 Z.001
N421 G01 X25.284 Y-19.177 Z-.01
N422 G01 X25.088 Y-19.158 Z-.02
N423 G01 X24.893 Y-19.177 Z-.031
N424 G01 X24.706 Y-19.234 Z-.041
N425 G01 X24.533 Y-19.327 Z-.052
N426 G01 X24.381 Y-19.451 Z-.062
N427 G01 X24.257 Y-19.603 Z-.073
N428 G01 X24.165 Y-19.776 Z-.083
N429 G01 X24.108 Y-19.963 Z-.094
N430 G01 X24.088 Y-20.158 Z-.104
N431 G01 X24.108 Y-20.353 Z-.115
N432 G01 X24.165 Y-20.541 Z-.125
N433 G01 X24.257 Y-20.714 Z-.136
N434 G01 X24.381 Y-20.865 Z-.146
N435 G01 X24.533 Y-20.99 Z-.157
N436 G01 X24.706 Y-21.082 Z-.167
N437 G01 X24.893 Y-21.139 Z-.178
```

```
N414 G01 X26.088 Y-20.158 Z.064
N415 G01 X26.069 Y-19.963 Z.054
N416 G01 X26.012 Y-19.776 Z.043
N417 G01 X25.92 Y-19.603 Z.033
N418 G01 X25.796 Y-19.451 Z.022
N419 G01 X25.644 Y-19.327 Z.012
N420 G01 X25.471 Y-19.234 Z.001
N421 G01 X25.284 Y-19.177 Z-.01
N422 G01 X25.088 Y-19.158 Z-.02 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N423 G01 X24.893 Y-19.177 Z-.031 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N424 G01 X24.706 Y-19.234 Z-.041 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N425 G01 X24.533 Y-19.327 Z-.052 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N426 G01 X24.381 Y-19.451 Z-.062 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N427 G01 X24.257 Y-19.603 Z-.073 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N428 G01 X24.165 Y-19.776 Z-.083 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N429 G01 X24.108 Y-19.963 Z-.094 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N430 G01 X24.088 Y-20.158 Z-.104 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N431 G01 X24.108 Y-20.353 Z-.115 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N432 G01 X24.165 Y-20.541 Z-.125 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N433 G01 X24.257 Y-20.714 Z-.136 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N434 G01 X24.381 Y-20.865 Z-.146 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N435 G01 X24.533 Y-20.99 Z-.157 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N436 G01 X24.706 Y-21.082 Z-.167 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
N437 G01 X24.893 Y-21.139 Z-.178 ; 5937e472-7583-5bdd-9fa6-535ac29985f6
```

Figure 4: Original NC File (left) and annotated NC file (right).

3.5.5. Phase 5 – Machining

During the machinist's processing of the stock part, there are no extra steps to be performed during the machining operation. The SF API, given the annotated NC file and the CNC machine's MTConnect URL, will extract and spatially align machining data. When the original NC file is loaded into the machine, the MTConnect data item "Program_running" is set to the original NC file's name. When the SF API observes this change in the MTConnect stream, a ShapeAspectOccurance is created for each ShapeAspectDefinition. The created shapeAspectOccurances are assigned V5 UUIDs based on the machine name and ISO 8061 time. The raw data is stored in InfluxDB, a time series database. The timestamp pairs extracted from changes in the "Block" data item are added to the document database under the ShapeAspectOccurance. When machining concludes, the quality engineer concludes the workflow by running the CMM inspection program on the part.

3.5.6. Phase 6 – Inspection

Phase six is the final phase of the workflow and is executed by the quality engineer. During this phase, the machined part is collected from the machinist and aligned to the CMM's coordinate plane using MCOSMOS 5.0 and the alignment plan generated in Phase 3. The SF API monitors the "MachineControllerProgram" MTConnect data item to determine whether the CMM runs an alignment or inspection operation. The "MachineState" data item determines when programs start and end. The MTConnect streams are extracted from the CMM and saved to the time series database and the shape aspect occurrence. When the inspection concludes, as shown in Figure 5, the QIF Result file is generated using Origin International's CMM Results Translator and uploaded using the SF API.

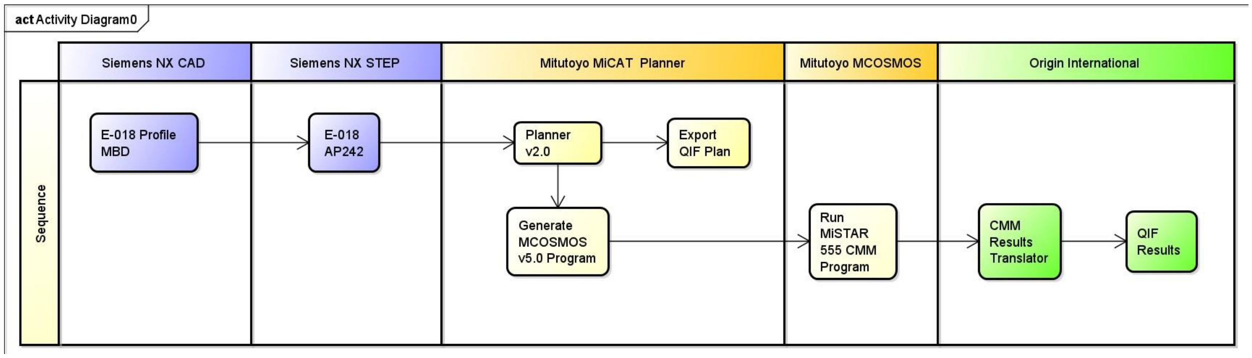

Figure 5: Activity Diagram for Inspection planning and execution.

After all the phases were concluded, a second inspection took place using traditional GD&T to inspect the part. This shows the versatility of the spatiotemporal digital thread in linking both types of inspection to geometric features. The plan is generated using traditional GD&T from PMI within the STEP file and follows the activity diagram shown in Figure 6.

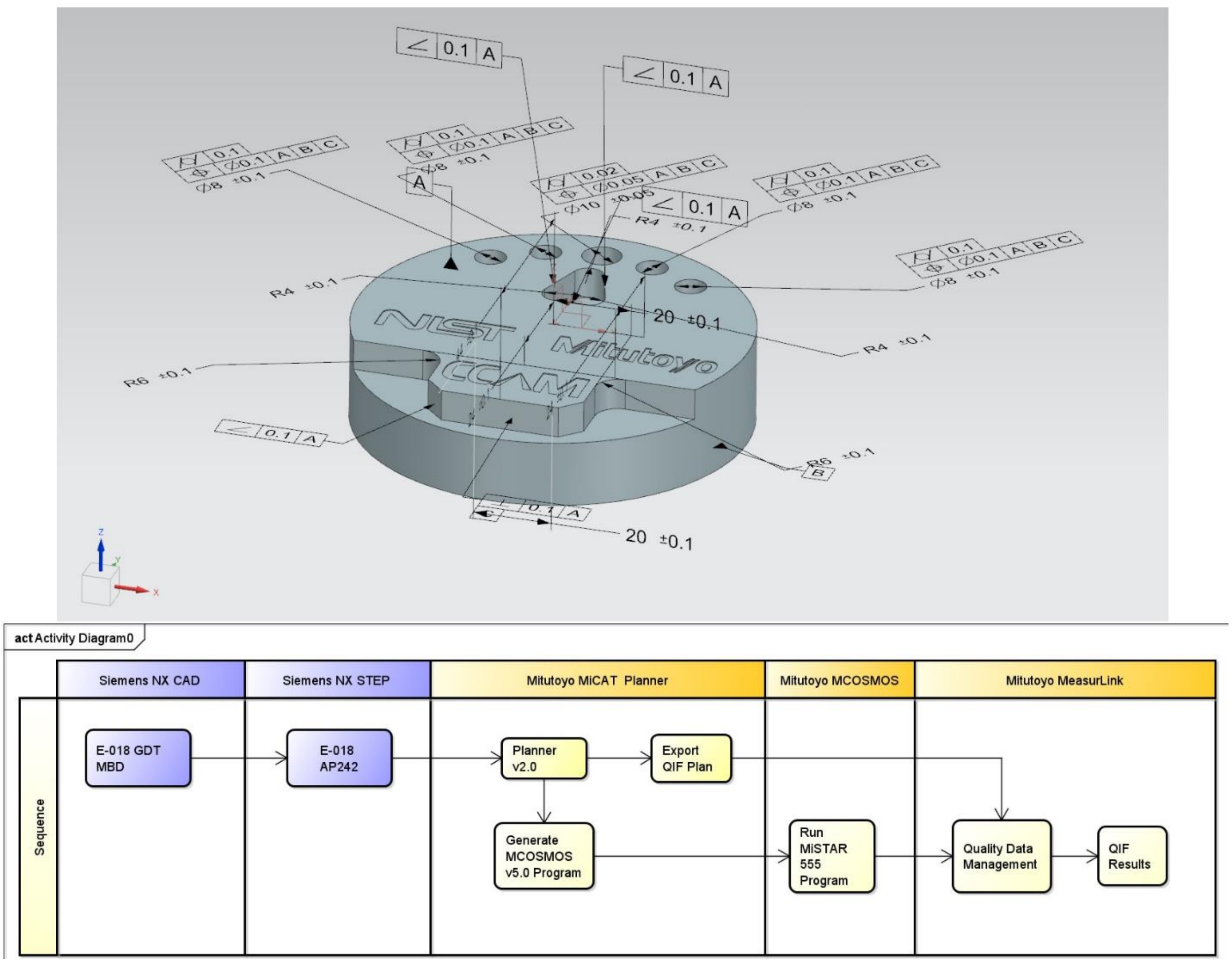

Figure 6: Part PMI and activity diagram for the traditional GD&T inspection.

3.6. Watertight Digital Thread

In addition to the process of assembling a digital thread detailed in Section 3.4, our partners, Metalogi and nVariate, propose an alternative methodology. They use a watertight model generated from the STEP AP242 file consisting of precise continuous surfaces using a PTC Creo Parametric extension module. Next, features of interest were identified and given arbitrarily chosen human-readable names, as shown in Figure 7. However, it should be noted that in the final run, these names are substituted with the UUIDs assigned to each feature.

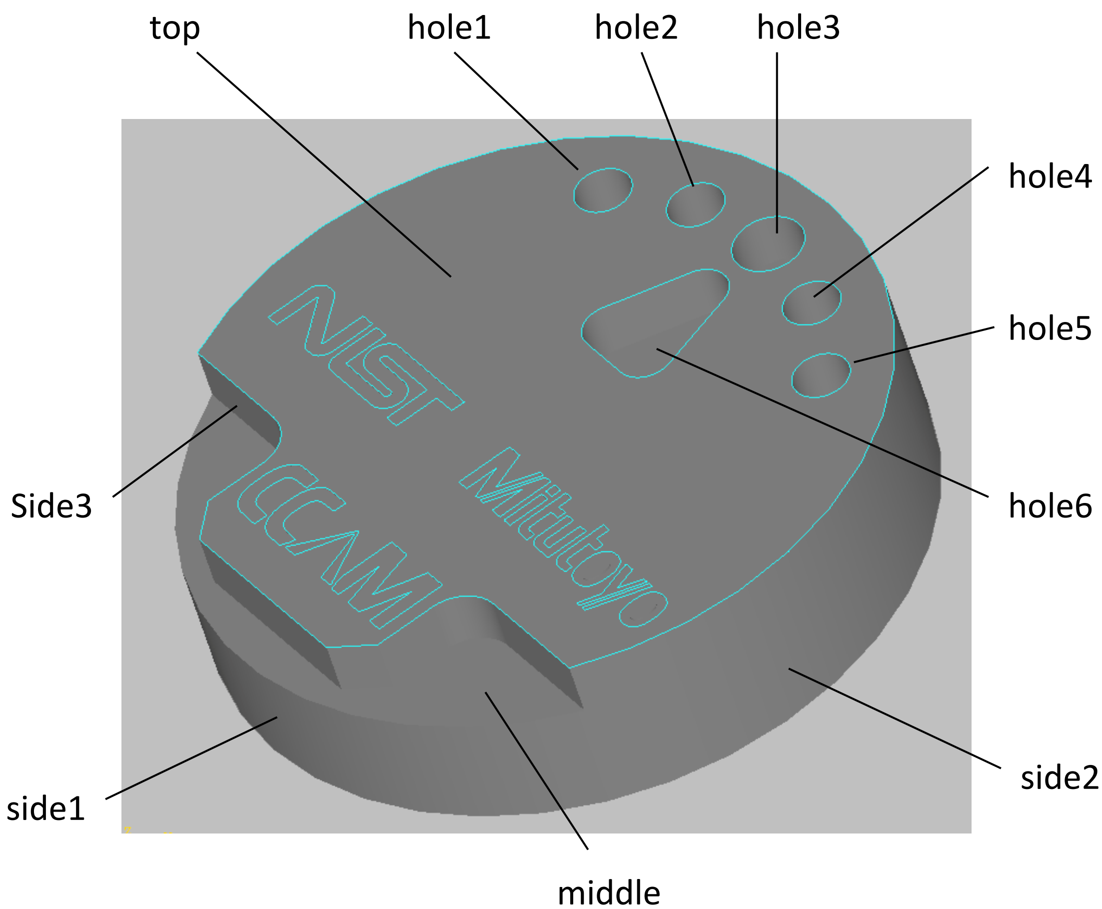

Figure 7 Watertight model with human readable names for features of interest.

To link the design data, currently being processed as the watertight model, to the execution data, Metalogi's GeometryStudio reads the MTConnect data and watertight model to perform real-time alignment of execution data to features of interest based on the cutting tool's interaction with material near the features. In Figure 8, a visualization of Metalogi's GeometryStudio where features being interacted with are displayed in red text.

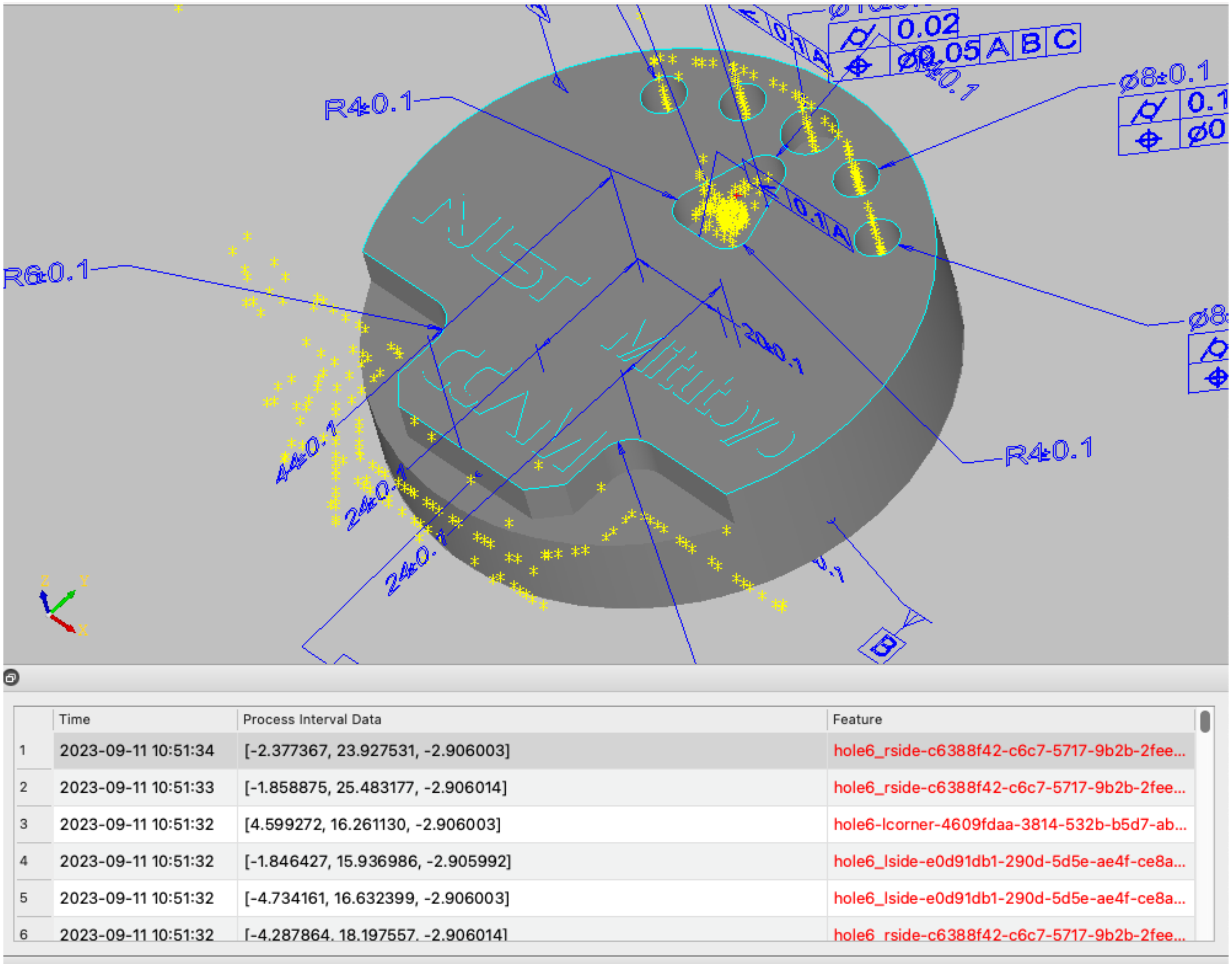

Figure 8 Alignment of execution data to features in the watertight model using Metalogi's GeometryStudio.

When the path position is unavailable in the MTConnect stream, the X, Y, and Z positions can replace the missing data item. This is commonly the case with older MTConnect implementations. However, there is a key problem with this approach. Namely, the timestamps on the X, Y, and Z position data items can be milliseconds apart making it difficult to determine instantaneous position.

This problem is fixed by introducing backfilling of position observations. The project used a three-axis machine with an older MTConnect implementation that reported each axis independently. The scan cycles were not consistent, which required us to perform the following transform to assemble a set of valid positions characteristic of the machine's motions.

The scan frequency for the MTConnect adapter was between 10-15 milliseconds, with some variability. The three-space position was considered valid and reported if each axis reported within ten milliseconds of the other. The following example illustrates this scenario:

```
2023-01-03T19:09:39.0897142Z|X_Position|381.635706
2023-01-03T19:09:39.0897142Z|Y_Position|186.3539058
2023-01-03T19:09:39.0897142Z|Z_Position|-76.84782697
```

The previous example represents the Path Position [381.635706, 186.3539058, -76.84782697] (in machine coordinates). When reported, the transform adjusts for workpiece offset, tool length, and part offset to make them align with the design geometry.

Suppose an axis does not report for more than one scan cycle. In that case, it can be for the following reasons: the MTConnect adapter has suppressed a duplicate reading, only publishing changes as it should, or the adapter skipped a measurement. The first case is the usual reason, meaning the following can be assumed: If the controller has not reported a change in value, the axis has been static, and when it reports, we can assume it has moved recently. In addition, when it reports a slight movement (< 0.001 mm), this is due to axial drift and is not significant.

Therefore, from the last point reported to this current position, we can assume that the other axes move in two-dimensional or one-dimensional space. It is safe to backfill the previous positions with the current value and send any completed three-space positions.

The following example illustrates motion only along the Z-axis:

```
2023-01-03T19:09:36.8165842Z|X_Position|381.6356968
2023-01-03T19:09:37.1756047Z|Y_Position|186.3538967
```

Only Z axis motion

```
2023-01-03T19:09:37.1776048Z|Z_Position|-114.9003502
2023-01-03T19:09:37.3126126Z|Z_Position|-112.8449619
2023-01-03T19:09:37.5396256Z|Z_Position|-108.7203834
2023-01-03T19:09:37.6536321Z|Z_Position|-106.6602343
2023-01-03T19:09:37.7606382Z|Z_Position|-104.6052351
2023-01-03T19:09:37.8716445Z|Z_Position|-100.4754995
```

Backfill X-Axis

X axis positions are <= 0.0001mm motion

Backfill to 2023-01-03T19:09:36.8165842Z

```
2023-01-03T19:09:37.9796507Z|X_Position|381.635706
2023-01-03T19:09:37.9806508Z|Z_Position|-98.42008084
```

Backfill X-Axis with previous Z-axis positions since <= 0.0001mm motion

```
2023-01-03T19:09:38.0886570Z|X_Position|381.6356968
2023-01-03T19:09:38.0886570Z|Z_Position|-96.35552926
```

Backfill X-Axis with previous Z-axis positions since <= 0.0001mm motion

```
2023-01-03T19:09:38.2026635Z|X_Position|381.635706
2023-01-03T19:09:38.2046636Z|Z_Position|-94.30084305
```

Backfill X-Axis with previous Z-axis positions since <= 0.0001mm motion
```
2023-01-03T19:09:38.3196702Z|X_Position|381.6356968
2023-01-03T19:09:38.3226703Z|Z_Position|-93.27053004
```

Backfill X-Axis with previous Z-axis positions since <= 0.0001mm motion
```
2023-01-03T19:09:38.4326766Z|X_Position|381.635706
2023-01-03T19:09:38.4336767Z|Z_Position|-91.2101329
```

Backfill X-Axis with previous Z-axis positions since <= 0.0001mm motion
```
2023-01-03T19:09:38.5416829Z|X_Position|381.6356968
```

Only Z-axis motion
```
2023-01-03T19:09:38.5426829Z|Z_Position|-89.14583311
2023-01-03T19:09:38.6486890Z|Z_Position|-87.08574498
2023-01-03T19:09:38.7556951Z|Z_Position|-85.03053231
2023-01-03T19:09:38.8667015Z|Z_Position|-82.96547716
2023-01-03T19:09:38.9797079Z|Z_Position|-78.85029082
```

Synchronous point

Backfill X position back to 2023-01-03T19:09:38.5416829Z

Backfill Y position back to 2023-01-03T19:09:37.1756047Z

Start new list
```
2023-01-03T19:09:39.0897142Z|X_Position|381.635706
2023-01-03T19:09:39.0897142Z|Y_Position|186.3539058
2023-01-03T19:09:39.0897142Z|Z_Position|-76.84782697
```

Creates the following set of points
```
[381.6356968, 186.3538967, -114.9003502]
[381.6356968, 186.3538967, -112.8449619]
[381.6356968, 186.3538967, -108.7203834]
[381.6356968, 186.3538967, -106.6602343]
[381.6356968, 186.3538967, -104.6052351]
[381.6356968, 186.3538967, -100.4754995]
[381.635706, 186.3538967, -98.42008084]
[381.6356968, 186.3538967, -93.27053004]
```

This resulted in minor data loss, but this removed the erroneous positions reported by the native MTConnect Agent. Additional enhancements could be made by introducing interpolation and verification based on assumptions such as the latest reported position being no more than 15ms old.

After the execution data is linked to the geometric features, the Inspection data from the model-based GD&T is used to morph the watertight model introduced from the STEP AP242 file into a digital twin created from the inspection data stream. Usable for evaluating local geometric artifacts.

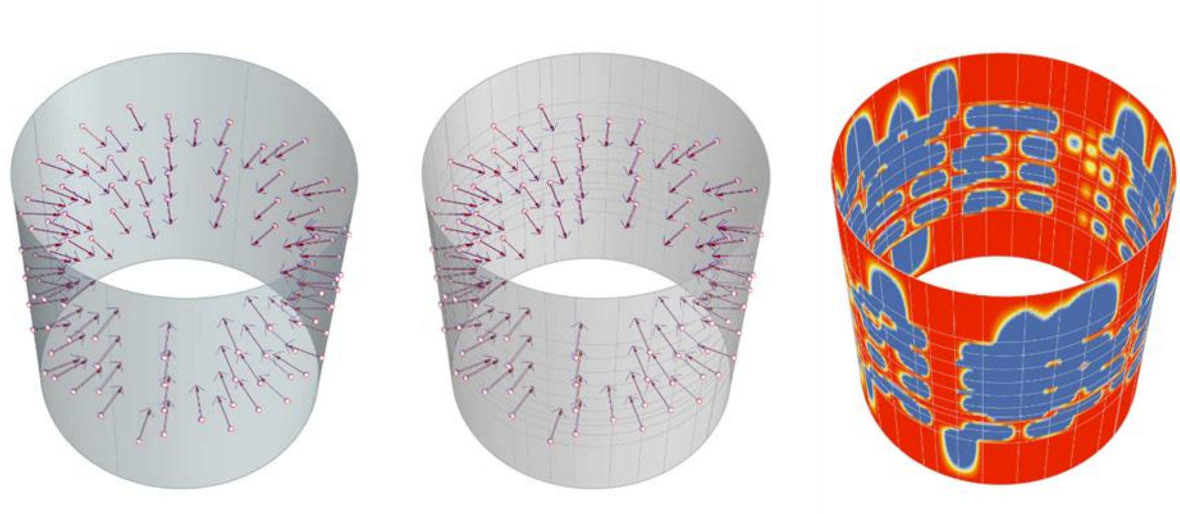

Figure 9 Nominal (as-designed) spline form, morphed (as-inspected) spline form, and the resulting distortion plot.

4. Findings and Results

The project delivered positive results for building a digital thread and utilizing established standards for various stages of lifecycle data. It is proven that a digital thread can be constructed across phases of the lifecycle, and a standards layer is a sound approach for spatiotemporal data alignment. The tech stack blends off-the-shelf data collection and data management tools with a custom API built specifically to handle spatiotemporal features. Implementation challenges remain for adapting time-sensitive networking to real-world equipment already in place, constraining how much a digital thread can be populated with disparate data sources before cost and custom coding requirements become impractical.

4.1. Task 1 – IEEE 802.1AS Time Synchronization and Time Sensitive Networking (TSN)

This work followed a prior demonstration of TSN published by National Instruments https://download.ni.com/pub/iot/IIoT_Lab_Flexible_Manufacturing_With_Time_Sensitive_Networking.pdf in which representative hardware was used for network infrastructure, image capture/vision, industrial process control, human-machine interface (HMI), and edge gateways. As with all testbeds, taking proven capability to additional sites or divergent hardware and software configurations gets easier with repetition.

The preferred approach for this project was to use commercial off-the-shelf networking hardware or to apply software updates or upgrades to the switch gear, CNC controller, and National Instruments controller already in place. In practice, software-only upgrades to support the required TSN features were not an option. Replacing the network switch and other hardware with new equipment was a straightforward path to get the required TSN features, but this alternative was cost-prohibitive for this project.

The next option was an alternative, software-based solution using only the hardware already in place. Without replacing any equipment in the manufacturing cell, the team considered implementing Precision Time Protocol (IEEE 802.1AS is based on PTP) on the CNC machine controller and the NI data acquisition controller with the Cisco IE 4000 ethernet switch configured as the PTP master clock. In this case, the conditions of the National Instruments testbed diverged significantly from norms for CNC machine tools. The VM10i controller at CCAM is built on an older version of the Windows operating system, which does not support PTP natively. This means adding TSN support or even simply adding PTP would require installing third-party software on vendor-supplied equipment. Understandably, this was also dependent on negotiating software rights management and warranty support for the equipment with the machinery supplier.

After careful consideration, this project ultimately prioritized the standards-based digital thread alignment work over the time synchronization work. Broad support from vendors in discrete manufacturing for the target standards other than 802.1AS means the impact from a successful digital thread built with those has a much higher chance of commercial replication. In fact, each of those standards once occupied a space where support for legacy equipment or end-of-life operating systems was unsupported; over time, the market adapted and expanded support from new machinery to include older devices. We hope that we can revisit the topic of IEEE TSN and 802.1AS time synchronization for industrial automation and synchronization of disparate data sources in a future project.

4.2. Task 2 – Machine learning-based alignment

Data alignment based on features in the data can be made with techniques such as change point detection (CPD) and dynamic time warping (DTW). In this case, data from the spindle-mounted dynamometer and acoustic emission sensor in the machine work envelope should be able to map to data sourced from the machine control. However, frequency mismatches in the sensor data versus control data left these techniques yielding unsatisfactory results. The machine learning approach was deemed insufficient for alignment as a standalone solution and instead should be considered only in tandem with network-based high-speed data synchronization.

4.3. Task 3 – Standards-based digital thread

Bespoke digital threads are well-established in discrete parts manufacturing, often appearing as offshoots or extensions to factory historian software. There is, however, still a need to demonstrate standards-based approaches that can scale beyond a few devices. In the case of STEP AP242, NC, MTConnect, and QIF, each standard is well established in design, planning, machining, and quality, and except NC, are developed with at least some interoperability with each other and of other standards up and down the manufacturing and data collection tech stack. Digital thread is a use case that tests these standards and their ability to fit an integrated system outside of their silos, the relative ease with which off-the-shelf hardware, open-source software for data collection, and off-the-shelf database tooling could be combined with the purpose-built SF API indicates there is a good path to commercial options following a standards-based approach.

4.4. How to develop a standards-based spatiotemporally aligned digital thread.

Generalizing from the specific target standards in this project, there are three key characteristics to consider when following this approach to digital thread. First, what manufacturing stage(s) are addressed by the standard? Second, how well does the standard communicate with standards in other manufacturing stages? Third, how well does the standard communicate with standards in the same manufacturing stages?

In answering the first question, standards can be isolated to manufacturing stages to aid in the future selection of the optimal minimal set of standards. In answering the second, competing standards can be removed based on whether they can link to future or previous stages. The value in the final question comes into play when increasing the “thickness” of the thread. For example, while the STEP AP242 file strongly supports model-based design and engineering, a future competing design standard may be better in some capacity. Should this occur, a new methodology of linking the design stages would not be necessary so long as one aligns the new design standard to the STEP AP242 format used in this work.

Ultimately, the standards selected to form the skeleton of the digital thread were the STEP AP242, NC file, MTConnect, and QIF. The STEP AP242 format support for shape aspects allowed easy identification of features of interest within the file. The NC file format's popularity in common manufacturing settings solidified its choice as the as-planned format. MTConnect, on the other hand, seamlessly linked between the as-executed data and the NC file through its "Program" and "Line" data items. Lastly, the QIF format's support for features defined by the STEP AP242 format made it an excellent choice, considering the previous set of standards within the chain.

4.5. Achievements

The major achievements in the project included:

1. Constructing a spatiotemporally aligned standards-based digital thread.
2. Layering multiple industry standards (STEP AP242, NC, MTConnect, and QIF) to describe disparate geometries on a part seamlessly.
3. Fitting as-planned and as-executed data into the digital thread.

5. Conclusions and Recommendations

In this project, CCAM developed a standards-based spatiotemporally aligned digital thread composed of common industry standards for representing the design, plan, execution, and inspection stages—namely, STEP AP242, NC, MTConnect, and QIF 3.0, respectively. The shape aspects definitions were defined using the design, plan, and inspection plan standards. Upon acquiring the process execution or inspection data, the shape aspect occurrences are generated using their respective shape aspect definitions as a template.

Future work should include a dedicated effort at using time-sensitive networks (TSN) to either utilize networking hardware that includes built-in support or build a bespoke solution on existing equipment that is representative of a typical manufacturing environment and includes older machines and controls. Another focus area should be generalizing the SF API from a bespoke solution for this project into best practices for such an API or developing a general API that supports more than the standards studied here.

6. References

[1] "Global Horizons: Final Report," United States. Air Force, AF/ST TR 13-01; Air Force/Small Business Technology Transer 13-01, Jun. 2013. Accessed: Nov. 05, 2021. [Online]. Available: https://www.hsdl.org/?abstract&did=741377

[2] T. Feldhausen, K. Saleeby, and T. Kurfess, "Spinning the digital thread with hybrid manufacturing," *Manuf. Lett.*, vol. 29, pp. 15–18, 2021, doi: 10.1016/j.mfglet.2021.05.003.

[3] V. Singh and K. E. Willcox, "Engineering design with digital thread," *AIAA J.*, vol. 56, no. 11, pp. 4515–4528, 2018, doi: 10.2514/1.J057255.

[4] T. Y. Pang, J. D. Pelaez Restrepo, C.-T. Cheng, A. Yasin, H. Lim, and M. Miletic, "Developing a Digital Twin and Digital Thread Framework for an 'Industry 4.0' Shipyard," *Appl. Sci.*, vol. 11, no. 3, Art. no. 3, Jan. 2021, doi: 10.3390/app11031097.

[5] ISO/TC 184/SC 4, "ISO 10303-242:2022," ISO. Accessed: Mar. 29, 2024. [Online]. Available: https://www.iso.org/standard/84667.html

[6] "MTConnect Standard," MTConnect. Accessed: Mar. 29, 2024. [Online]. Available: https://www.mtconnect.org/standard-download20181

[7] ISO/TC 184/SC 4, "ISO 23952:2020," ISO. Accessed: Mar. 29, 2024. [Online]. Available: https://www.iso.org/standard/77461.html

[8] R. Bonnard, J. Hascoet, P. Mognol, E. Zancul, and A. Alvares, "Hierarchical object-oriented model (HOOM) for additive manufacturing digital thread," vol. 50, pp. 36–52, Jan. 2019, doi: 10.1016/j.jmsy.2018.11.003.

[9] R. Bonnard, J. Hascoet, P. Mognol, and I. Stroud, "STEP-NC digital thread for additive manufacturing: data model, implementation and validation," *Int. J. Comput. Integr. Manuf.*, vol. 31, pp. 1141–1160, Aug. 2018, doi: 10.1080/0951192X.2018.1509130.

[10] S. Kwon, L. V. Monnier, R. Barbau, and W. Z. Bernstein, "Enriching standards-based digital thread by fusing as-designed and as-inspected data using knowledge graphs," *Adv. Eng. Inform.*, vol. 46, p. 101102, Oct. 2020, doi: 10.1016/j.aei.2020.101102.

[11] IEEE, "IEEE 802.1 AV Bridging Task Group." Accessed: Mar. 29, 2024. [Online]. Available: https://ieee802.org/1/pages/avbridges.html

[12] National Instruments, "IIoT Lab Flexible Manufacturing With Time Sensitive Networking," 2018.

[13] A. S. Polunchenko and A. G. Tartakovsky, "State-of-the-Art in Sequential Change-Point Detection," *Methodol. Comput. Appl. Probab.*, vol. 14, no. 3, pp. 649–684, Sep. 2012, doi: 10.1007/s11009-011-9256-5.

[14] M. Müller, Ed., "Dynamic Time Warping," in *Information Retrieval for Music and Motion*, Berlin, Heidelberg: Springer, 2007, pp. 69–84. doi: 10.1007/978-3-540-74048-3_4.