National Physical Laboratory

NPL REPORT MS 51

AN OVERVIEW OF DIGITAL THREAD TESTING AND SOFTWARE INTEROPERABILITY FOR DIGITAL SUPPLY CHAINS

JOAO GREGORIO & PAUL DUNCAN

DECEMBER 2023

An Overview of Digital Thread Testing and Software Interoperability for Digital Supply Chains

Joao Gregorio & Paul Duncan
Informatics, Data Science, National Physical Laboratory

ISSN 1754-2960

DOI ADDRESS: https://doi.org/10.47120/npl.MS51

National Physical Laboratory
Hampton Road, Teddington, Middlesex, TW11 0LW

Approved on behalf of NPLML by
Thomas Byrne, Group Leader.

CONTENTS

GLOSSARY/ABBREVIATIONS

TERM	DEFINITION
2D	2-dimensional
3D	3-dimensional
AMRC	Advanced manufacturing Research Centre
ASME	American Society of Mechanical Engineers
BAE	British Aerospace
CAD	Computer-aided design
CAE	Computer-aided engineering
CAM	Computer-aided manufacturing
CMM	Coordinate-measuring machine
DMS	Digital Minimum Standards
DQP	Digital Quality Planning
ERP	Enterprise Resource Planning
ISO	International Organization for Standardization
MBE	Model Based Enterprise
MBEE	Model Based Enterprise Environment
MES	Manufacturing Execution System
NCC	National Composites Centre
PLM	Product Lifecycle Management
PMI	Product Manufacturing Information
Prime	Primary manufacturer, supply chain centre
SHA-1	Secure Hashing Algorithm 1
SPC	Statistical Process Control

EXECUTIVE SUMMARY

This paper acknowledges the widely accepted definition of a digital thread as a traceable flow of data and information, pertaining to the whole life cycle of a product, which spans across several network boundaries owned by different organisations. The use of model-centric systems, which store data and information in queryable models rather than conventional digital documents, is widely recognised as a more efficient and reliable approach for implementing and deploying a digital thread. However, the development and deployment of such systems need significantly more resources compared to conventional document-based solutions. Although these solutions have the potential to construct resilient digital threads, more investment in infrastructure is necessary to allow the seamless transfer of data and metadata across network boundaries, while ensuring the preservation of the information when being shared.

This research investigates the use of hashing techniques for file content verification. It examines how this method can help maintain the original design intention of technical data packs when shared amongst stakeholders over different network boundaries. In the context of this report, network boundaries are defined as the limits of a digital network owned by different organisations, delineating the extent of a network's reach and control (defined by technical constraints, security protocols, and organisational policies). Furthermore, the notion of software interoperability is also regarded as crucial for the establishment of a taxonomical vocabulary aimed at facilitating communication among various stakeholders.

The combined use of these two concepts offers a robust framework for the examination and authentication of data transmission pathways within the digital testbed offered by the Connected MBEE Testbed. This framework aims to accurately simulate the real-world context of supply chain integration involving multiple organisations. The aforementioned objective is accomplished by employing the file content verification technique, which enables a more efficient evaluation of the pathways through which data is transferred. Additionally, the examination of a software interoperability matrix showcases its potential for integration with a Digital Minimum Standards assessment to aid with the classification of third-party suppliers based on their capabilities, facilitating their integration with supply chains.

In summary, this report outlines two primary characteristics that facilitate the establishment of a document-centric digital thread throughout numerous organisations: data transfer and software interoperability. Additionally, it offers guidance on the creation of a tool that facilitates the active examination and evaluation of data transfers, as well as the preservation of design intent. Furthermore, it establishes the foundation for a methodology to address the problem of software interoperability within the framework of supplier classification.

INTRODUCTION

There are various interpretations regarding the concept of a Digital Thread and its components across different application areas. Nevertheless, it is widely acknowledged that the term refers to the traceable flow of data and information pertaining to the life cycle of a product, encompassing its design, manufacturing, and even post-deployment evaluation of performance (1).

Within the framework of the Connected Model Based Enterprise Environment (MBEE) Flagship (2), the concept of a Digital Thread holds significance as it offers the potential to describe a network of interconnected primes and their corresponding third-party suppliers. Here the term “primes” is used to refer to the main organisations in a supply chain network, which are typically larger companies that have significant influence over the supply chain itself due to their size, resources, or strategic positions.

The primes operate within independent digital networks, each delineated by its own set of "network boundaries." These boundaries serve to define the reach and authority of a network, typically established based on technical limitations, security measures, and organisational rules. The transmission of data and information across network borders is a continuous process that occurs during the whole lifespan of a product, creating a cohesive entity referred to as a "Digital Thread". The hindrance of this flow might arise from disparities in data formats, software systems, and security protocols across diverse networks.

The description and comprehension of such systems enables a smooth integration of numerous autonomous supply chains, perhaps incorporating a mechanism for automatically documenting the reliability, variability, and quality of shared data and information. This feature enables a streamlined approach for efficiently and verifiably updating modifications on files that are shared among multiple individuals facilitating the process of version controlling documentation. Moreover, it can be utilised for the purpose of problem identification, such as the detection of data degradation sources, and facilitating their prompt resolution.

Despite the advantages offered by a well-established Digital Thread, whose conceptual representation is shown in Figure 1, it frequently embodies an idealised framework that interconnects numerous entities, posing significant challenges in terms of feasibility, given its predominantly model-centric design. The term "model-centric" in this particular context refers to the concept that data and information are contained within a model, allowing for querying and retrieval. This stands in contrast to traditional platforms that primarily rely on file and document exchange. The primary advantage is in the immediate propagation of updates through the model, hence eliminating any concerns around inadvertent use of outdated or deprecated data. Furthermore, the process of segregating information becomes significantly more manageable, facilitating the obfuscation of sensitive and confidential data while enabling the dissemination of other information fragments. This task is often intricate when sharing documents through conventional methods, as it is complicated by the challenges posed by redaction procedures.

The realisation of such a system necessitates substantial investments of time and money, as well as significant operational changes deriving from the change from file-centric to model-centric systems, for its successful creation and execution.

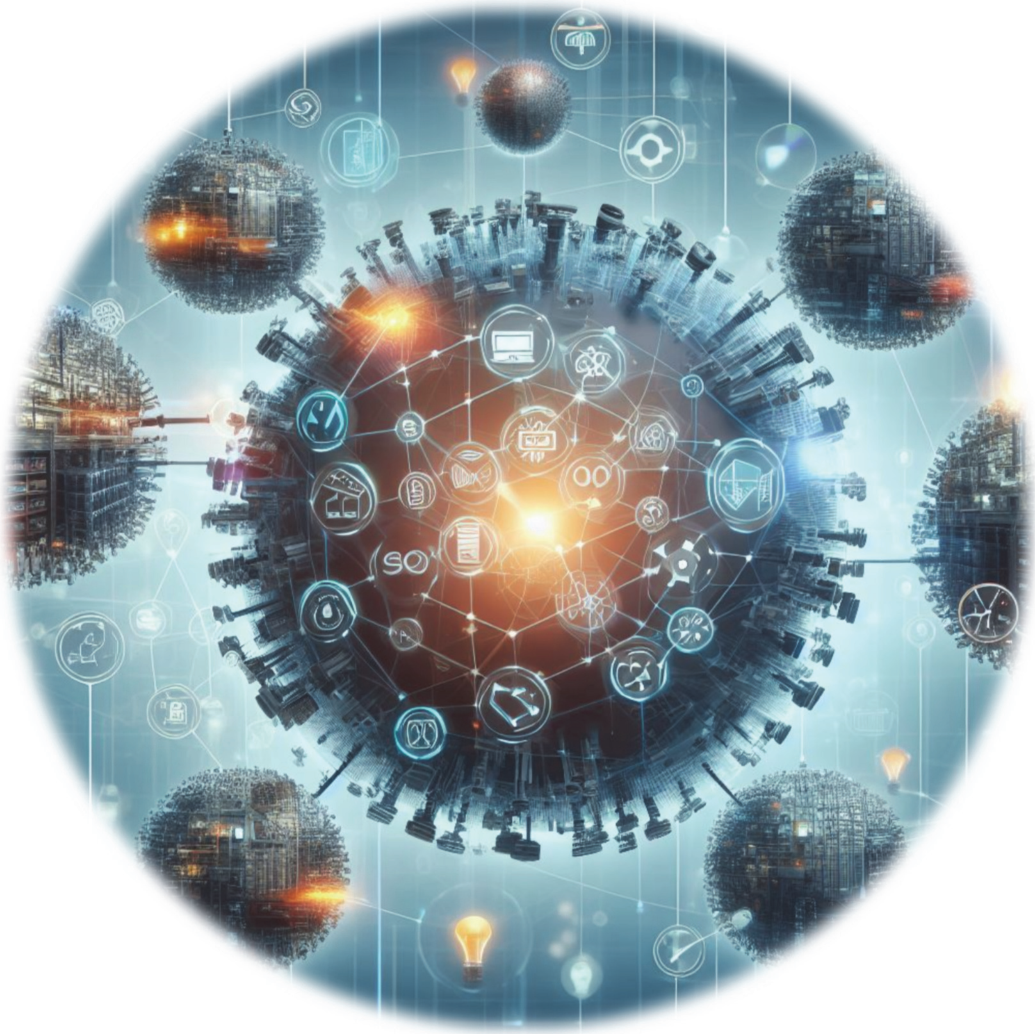

Figure 1 - Conceptual representation of a Digital Thread: Each sphere represents its own organisation and the confines of its network boundaries. Interconnectivity between spheres shows communication pathways enabaled by digital thread. This overlap enables information and data sharing to be seamless and metadata to be traceable over different networks.

Taking into consideration the aforementioned factors, the concept of a regulated file-oriented data transfer platform, employing readily available technologies, which facilitates the interconnection of numerous systems, presents an attractive solution due to its ability to also maintain process records without the need for such drastic changes. These records possess the capability to retain metadata pertaining to the integrity, variability, and quality of data and files. This metadata can be utilised to monitor alterations, identify errors, and determine the origins of data degradation, and in turn be used to facilitate the prompt dissemination of updated versions of files across various supply chains or rectify any identified error sources.

Nevertheless, it is important to note that any system, including the one mentioned in the previous paragraph, has its own set of prerequisites, and necessitates specific validation procedures. These procedures are crucial in guaranteeing that the data and information remain identifiable and maintain their integrity during the process of being transferred across network boundaries, which serve as the separation between the networks of the involved companies. The focal points addressed in this work are aimed at answering the questions surrounding the validation of data integrity and content during transfers across network boundaries, as well as the assurance of universal accessibility and editing capabilities of relevant file types for all organisations involved.

This work discusses the development and implementation of a validation tool that enables the verification of file contents. Specifically, the tool allows for the comparison of an original file with a duplicate that has been transmitted across a network boundary, in order to assess any alterations to their integrity. Additionally, this work gives a compilation of findings pertaining to file interoperability, specifically focusing on the capacities of three distinct Computer-Aided Design (CAD) software packages to successfully open and read specific design features. Furthermore, a comparative analysis will be conducted between these factors and the corresponding criteria outlined in the ISO and ASME Standards for compliance checking. The ability to verify, and potentially automate the verification of, file changes across network boundaries, along with a reference guide for software interoperability, are two essential tools for ensuring the successful implementation of a file-centric Digital Thread.

OVERVIEW OF THE TESTBED

Prior to delving into the procedure established for validating changes to file contents or the software interoperability matrix, it is fundamental to provide an overview of the testbed in which the procedure will be implemented.

Figure 2 illustrates the implementation and use of the Digital Thread Testbed within the framework of the Connected MBEE hub and has been adapted from work developed within the scope of the flagship. The system consists of two prime networks, one of which includes an additional controlled access network, as well as three supplier networks. Design files can be originated from any of the networks and are processed using different CAE software.

These files are then transmitted to other networks over the described pathways, where they are opened, and maybe further modified, using CAE software that differs from the one first employed for their creation. Subsequently, the files and data are subjected to a reverse conversion procedure, restoring them to their initial forms, and returned to their origins for the purpose of verification and validation. The aim is to ensure that the original design intent remains unaltered throughout the entirety of the process. Furthermore, various Process Lifecycle Management (PLM) systems are used within the different networks to conduct additional assessments on process interoperability.

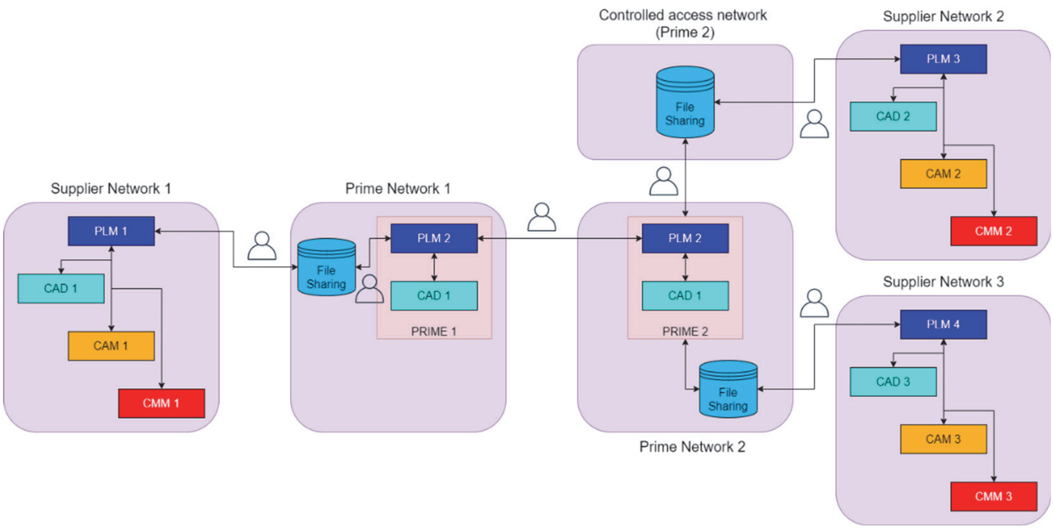

Figure 2 – Schematic overview of the Digital Thread Tesbed used in the Connected MBEE hub.

Various data formats are encompassed under the scope of Digital Thread testing. These are design-specific file types, such as Siemens NX, Catia, SolidWorks, STEP and JT. Additionally, there are more conventional file types that form the accompanying documentation, such as Word, Excel, CSV, and PDF. The technical data pertaining to a specific design is commonly transferred collectively as a package of multiples of these files and is often compressed for the purpose.

The testbed incorporates many networks that use distinct PLM and CAE software. This configuration aims to replicate a practical cooperation setting encompassing diverse supply chains. Therefore, it is crucial to assess the interoperability capabilities and constraints of the software used, in addition to the capacity for data exchange and management of the testbed itself. This evaluation should extend beyond merely identifying changes in file contents, hence the need for an interoperability matrix such as the one discussed in this work.

FILE CONTENT VERIFICATION

In the process of uploading or transferring files across network boundaries, it is common for the method of transfer to be obscured. In fact, several platforms, and PLM systems, employ a procedure where they extract the contents of files and reconstruct the files on the receiving end on the other side of the network boundary. This process often results in the loss of some, or even all, metadata attributes attached to the files. Consequently, these metadata cannot always be used for validation purposes. Nevertheless, it is important to note that design intent is primarily associated with the contents of a file rather than its metadata. Therefore, this approach of file verification is considered to be more resilient compared to a straightforward verification of metadata properties.

A simple software program was devised in order to authenticate changes made to the contents of transferred files. The provided solution uses the widely adopted Secure Hash Algorithm 1 (SHA-1) implemented as a standard library of the Python programming language (3; 4). This algorithm generates a hash value of 160 bits by processing the contents of a file. The resulting hash value is then shown to the user in the form of a string of 40 hexadecimal digits. The value of this string is dependent upon the contents of the file, meaning that files with identical contents will generate the same string regardless of their associated metadata attributes, such as authorship and dates of creation and modification.

Figure 3 illustrates the validation method for the proposed verification algorithm. In order to verify the efficacy of the approach, a technical technical data pack, containing a design file and its corresponding documentation was duplicated and thereafter subjected to deliberate alterations: a modification to the design file itself as well as to each of the accompanying documentation files; the removal of a file; and the addition of a file.

The conducted tests have provided evidence that the algorithm had the capability to identify changes properly and correctly in the content of files, irrespective of the specific file format being processed. Furthermore, a comparative analysis conducted using an unaltered version of the original technical data pack yielded no discernible changes, aligning with the anticipated outcome. Finally, the software was designed to generate a log file that includes timestamps and descriptions of identified changes arising from comparison operations, serving the purpose of maintaining a comprehensive testing record.

The suggested procedure demonstrates an efficient, straightforward, and cost-effective approach to accelerate the verification process of data transmission pathways on the previously illustrated testbed (Figure 2). This method is then capable of identifying any authorised or unauthorised changes to file contents that may impact the original design intent. Moreover, the ability to execute such an activity programmatically also presents opportunities for greater automation in the future.

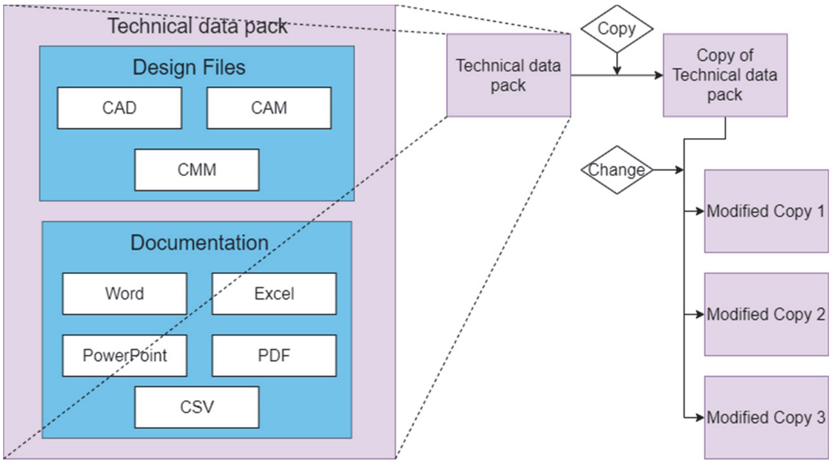

Figure 3 - Validation framework for the file checking algorithm. Pre-established files are inserted into a technical data pack, which is copied and selectively modified to replicate data changes, either degradation or intentional, over transfers. The algorithm is then applied to detect all changes on replicated files.

SOFTWARE INTEROPERABILITY

Ensuring the accessibility of data and information to all relevant actors is also a crucial element in constructing a continuous digital thread. Within the framework of supply chain integration, it is primarily imperative to ensure the preservation of the design intent of transferred files. Hence, ensuring seamless integration across multiple supply chains requires demonstrating the capabilities of commonly used software packages, as it is certain that players involved possess different software packages for opening and handling 3D design files. Moreover, it is important to consider the element of compliance. As stated in our earlier report titled "Model Based Definition – Recommendations for Supply Chain Integration," standards have a significant impact on determining optimal practices within the industry, and adherence to their guidelines frequently enhances interoperability between different systems (5).

The Connected MBEE technical team, under the leadership of the National Composites Centre (NCC), conducted an assessment to compare the feature creation capabilities of three widely used software packages, these being CATIA V5R30, NX1973, and SolidWorks 2020, with each other and with the requirements defined by standards created by the two major standards bodies in this area (ISO and ASME). The exercise also considered the dichotomy between machine readability and human readability of the features.

In this exercise, a comprehensive collection of 84 Project Manufacturing Information (PMI) elements was considered. Among these, 58 features are included as per ISO standards and 56 features as per ASME standards in 3D designs in order to adhere to the regulations set forth by the respective standard bodies. Among the three software packages that were examined, CATIA demonstrated the ability to generate 80% of the features, NX exhibited the ability to generate 92% of the features, and SolidWorks displayed the ability to generate 57% of the features. Lastly, it is worth noting that among the complete range of features, a total of 61 may be processed by machines, but the remaining features can only be comprehended by humans. The full list of tables, along with the results can be found attached at the end of the document as supporting information in Table 1 in the Appendix.

Although this existing iteration of the interoperability matrix is somewhat restricted, as it solely focuses on selected software packages and versions, it presents an opportunity for the creation and establishment of an extensive vocabulary. This vocabulary could be used to promptly evaluate supplier capabilities, specifically in terms of their accessible tools and processes, thereby promptly determining their potential for integration within the supply chain.

DIGITAL MINIMUM STANDARDS

A Digital Minimum Standards (DMS) framework has undergone thorough validation and has been employed to assess the capabilities of their suppliers (6). The use of this interoperability matrix, in conjunction with the DMS, can establish a solid foundation for the development of a systematic taxonomy that enables the evaluation and categorisation of supplier capabilities. For instance, using the aforementioned software packages and their respective versions as an example, one can categorise them based on their distinct design capabilities such as CAD and CAM. Additionally, these software packages can also be classified according to their design types, such as 3D or 2D, to provide further specificity. The implementation of a structured framework, as illustrated by Figure 4, would facilitate the efficient evaluation of capabilities and the categorisation of suppliers based on those capabilities. Furthermore, this framework has the potential to serve as a valuable tool in the field of cybersecurity, enabling the quick addressing of software and version specific vulnerabilities found.

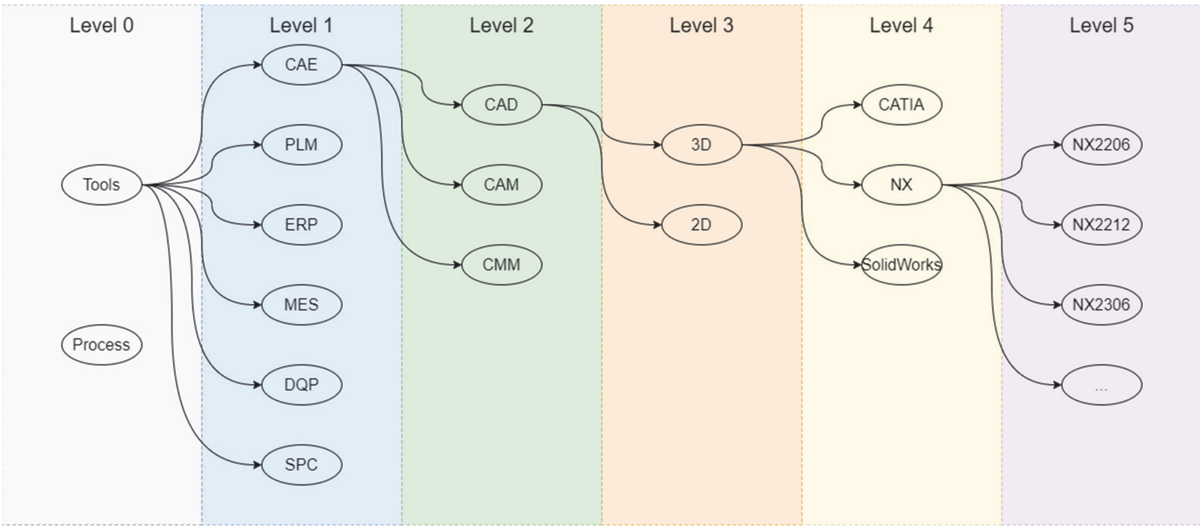

Figure 4 - Example structure of a taxonomical vocabulary for the quick identification and assessment of supplier capabilities and classification.

It is important to acknowledge that the taxonomic structure that has been discussed thus far consists of work in development and that the illustration in Figure 4 is merely representative of such a system. Such a framework has the potential to drive the creation of a data model for the identification of supplier capabilities, facilitating the integration of third-party suppliers into a supply chain.

CONCLUSIONS

This report examines the necessary criteria for implementing a file-centric digital thread-enabled supply chain system. There are two crucial factors that are essential for the successful execution of this undertaking. Firstly, it is critical to ensure the integrity of file transmission pathways in order to preserve the design intent during the transmission process across interconnected supply chains, without any loss or deterioration. Secondly, it is necessary to evaluate the software interoperability to ensure effective communication of the design intent among all relevant stakeholders.

In order to tackle the first of these aspects, this work presents a straightforward software solution that relies on the SHA-1 procedure, as implemented in the Python programming language, to authenticate that there have been no changes in file contents during the transfer process. These changes can occur either deliberately, as a result of planned design adjustments, or inadvertently, which may result in a misinterpretation of the design intent.

The investigation of the second part involved a collaborative exercise that sought to evaluate the capacities and constraints of widely used design software packages in generating PMI characteristics. The generated interoperability matrix offers an initial assessment of software interoperability and its ability to adhere to standards, such as those created by ISO and ASME. However, its primary value lies in its potential to contribute to the creation of an organised vocabulary for efficiently evaluating and categorising supplier capabilities.

FUTURE WORK

The main focus of future work should primarily be the extensive incorporation of the software interoperability matrix with the DMS assessment, that was briefly outlined in this report. The use of a structured vocabulary offers advantages not just in terms of promptly evaluating and categorising suppliers, but also in terms of its potential to be applied across other sectors. For instance, the ability to promptly identify all suppliers using certain software versions facilitates the potential mitigation of cyber-security threats in the event of vulnerabilities that are software- and version-specific being detected. Moreover, use of a controlled vocabulary allows for the streamlined evaluation of data and information obtained from DMS assessments. This, in turn, facilitates the development of a comprehensive data model, which enhances the understanding of the necessary requirements for achieving complete supply chain integration.

ACKNOWLEDGEMENTS

This work was funded by the UK Government Department for Science, Innovation and Technology through the UK's National Measurement System. We would also like to thank all our partners from the NCC, AMRC, Rolls-Royce, BAE Systems, and Siemens who worked jointly on the Connected MBEE Hub under the Made Smarter Innovation Hubs programme.

REFERENCES

1. *The digital thread in industry 4.0.* Tiziana Margaria, Alexander Schieweck. Bergen : Springer International Publishing, 2019.

2. Digital Catapult. Connected MBEE. *Digital Supply Chain Hub UK.* [Online] 2023. [Cited: 12 10 2023.] https://digitalsupplychainhub.uk/showcase/connected-mbee/.

3. NIST. Secure Hashing. *Computer Security Resource Center.* [Online] NIST, 12 4 2011. [Cited: 12 10 2023.] https://web.archive.org/web/20110625054822/http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html.

4. Hashlib. *Python Documentation.* [Online] 12 10 2023. [Cited: 12 10 2023.] https://docs.python.org/3/library/hashlib.html#module-hashlib.

5. Joao Gregorio, Jennifer Hughes, Chris Lorch. *Model Based Definition--Recommendations for Supply Chain Integration.* s.l. : NPL Publications, 2023.

6. Rolls-Royce. Digital Supply Chain. *Digital Supply Chain.* [Online] Rolls-Royce, 2023. [Cited: 12 10 2023.] https://www.rolls-royce.com/supplychain/#/home.

APPENDIX

Table 1 - PMI features comparison matrix for common CATIA, NX and SolidWorks versions as developed and assessed in the Connected MBEE Hub.

FEATURE	STANDARDS		CAD CREATION CAPABILITY		
	ISO	ASME	CATIA V5R30	NX1973	SolidWorks 2020
3D GEOMETRY	Y	Y	Y	Y	Y
LINEAR DIMENSION	Y	Y	Y	Y	Y
RADIAL DIMENSION	Y	Y	Y	Y	Y
ANGULAR DIMENSION	Y	Y	Y	Y	Y
HOLE & THREAD CALLOUT	Y	Y	Y	Y	Y
CHAMFER DIMENSION	Y	Y	Y	Y	N
ORDINATE DIMENSION	N	Y	Y	Y	N
THICKNESS DIMENSION	N	Y	N	Y	N
ARC LENGTH DIMENSION	Y	Y	Y	Y	N
NOTE ANNOTATION	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME ANNOTATION	Y	Y	Y	Y	Y
DATUM FEATURE SYMBOL ANNOTATION	Y	Y	Y	Y	Y
SURFACE FINISH ANNOTATION	Y	Y	Y	Y	Y
DATUM TARGET ANNOTATION	Y	Y	Y	Y	Y
WELD SYMBOL ANNOTATION	Y	Y	N	Y	Y
BALLOON ANNOTATION	Y	N	Y	Y	Y
SUPPLEMENTAL REGION	N	N	Y	Y	N
SUPPLEMENTAL CENTRE MARK	Y	N	N	Y	Y
SUPPLEMENTAL CENTRELINE	Y	N	Y	Y	Y
SUPPLEMENTAL BOLT CIRCLE CENTRELINE	Y	N	Y	Y	N
SUPPLEMENTAL SECTION VIEW	Y	N	Y	Y	Y
SUPPLEMENTAL TABLE	Y	Y	Y	Y	Y
CUSTOM SYMBOL (I.E DELTA/FLAG NOTE)	Y	N	Y	Y	Y
COMPOSITES PLY BOUNDARIES	N/A	N/A	Y	Y	N
COMPOSITES LAYUP SEQUENCE	N/A	N/A	Y	Y	N

COMPOSITES ORIENTATIONS (IE PLY ORIENTATION)	N/A	N/A	Y	Y	N
COMPOSITES MATERIAL NAME	N/A	N/A	Y	Y	N
COMPOSITES MATERIAL THICKNESS	N/A	N/A	Y	Y	N
COMPOSITES MATERIAL ROLL WIDTH	N/A	N/A	Y	Y	N
COMPOSITES ROSETTE	N/A	N/A	Y	Y	N
COMPOSITES CORE	N/A	N/A	Y	Y	N
FEATURE CONTROL FRAME-FORM-STRAIGHTNESS	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-FORM-FLATNESS	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-FORM-CIRCULARITY	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-FORM-CYLINDRICITY	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-PROFILE-PROFILE OF A LINE	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-PROFILE-PROFILE OF A SURFACE	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-ORIENTATION-ANGULARITY	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-ORIENTATION-PERPENDICULARITY	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-ORIENTATION-PARALLELISM	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-LOCATION-POSITION	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-LOCATION-CONCENTRICITY	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-LOCATION-SYMMETRY	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-RUNOUT-CIRCULAR RUNOUT	Y	Y	Y	Y	Y

FEATURE CONTROL FRAME-RUNOUT-TOTAL RUNOUT	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-MAXIMUM MATERIAL BOUNDARY/CONDITION	Y	Y	Y	Y	y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-LEAST MATERIAL BOUNDARY/CONDITION	Y	Y	Y	Y	y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-TRANSLATION	N	Y	N	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-PROJECTED TOLERANCE ZONE	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-FREE STATE	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-TANGENT PLANE	N	Y	Y	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-UNEQUALLY DISPOSED PROFILE	N	Y	N	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-INDEPENDENCY	N	Y	N	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-STATISTICAL TOLERANCE	N	Y	Y	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-CONTINUOUS FEATURE	Y	Y	N	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-DIAMETER	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-SPHERICAL DIAMETER	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-RADIUS	Y	Y	Y	Y	N
FEATURE CONTROL FRAME-MODIFYING	Y	Y	Y	N	N

SYMBOL-SPHERICAL RADIUS					
FEATURE CONTROL FRAME-MODIFYING SYMBOL-CONTROLLED RADIUS	Y	Y	Y	N	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-SQUARE	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-REFERENCE	N	Y	Y	N	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-ARC LENGTH	Y	Y	N	N	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-DIMENSION ORIGIN	Y	Y	N	N	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-BETWEEN	Y	Y	Y	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-ALL AROUND	N	Y	N	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-ALL OVER	N	Y	N	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-DEGREE	N	N	Y	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-RECIPROCITY	N	N	Y	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-MODIFIER A	N	N	Y	N	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-FOR ORIENTATION CONSTRAINT ONLY	N	N	Y	N	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-MINIMAX FEATURE	Y	N	Y	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-LEAST SQUARES FEATURE	Y	N	Y	Y	N
FEATURE CONTROL FRAME-MODIFYING	Y	N	Y	Y	N

SYMBOL-MINIMUM CIRCUMSCRIBED FEATURE					
FEATURE CONTROL FRAME-MODIFYING SYMBOL-MAXIMUM INSCRIBED FEATURE	Y	N	Y	Y	N
FEATURE CONTROL FRAME-AXIS INTERSECTION	N	N	N	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-REGARDLESS OF FEATURE SIZE	Y	Y	N	Y	Y
FEATURE CONTROL FRAME-MODIFYING SYMBOL-FLAG	N	N	N	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-DATUM	Y	N	N	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-DOT TERMINATED	N	N	N	Y	N
FEATURE CONTROL FRAME-MODIFYING SYMBOL-ENVELOPE	Y	N	N	Y	N
BASIC (BOXED) DIMENSIONS	Y	Y	Y	Y	Y
LINEAR FEATURE TOLERANCE (BILATERAL TOLERANCE)	Y	Y	Y	Y	Y
LINEAR FEATURE TOLERANCE (UNILATERAL TOLERANCE)	Y	Y	Y	Y	Y