Digital Threads via Knowledge-Based Engineering Systems

Joe David*†, Eeva Järvenpää*, Andrei Lobov†
*Tampere University, Tampere, Finland
†Norwegian University of Science and Technology, Trondheim, Norway
joe.david@tuni.fi, joesd@stud.ntnu.no
eeva.jarvenpaa@tuni.fi, andrei.lobov@ntnu.no

***Abstract*—Digitalization of manufacturing is led by the notion of a digital thread, a framework that allows for integrating authoritative and trustable data of a product throughout its lifecycle. Under the model-based enterprise (MBE) vision, digital three-dimensional (3D) model of a product serve as this authoritative source of information across the product lifecycle. A class of systems called knowledge-based engineering (KBE) systems represents the evolution of knowledge-based systems towards designing these models and this study presents a case for using such systems to realize the digital thread in a model-based enterprise. The paper proposes four directions and potential enabling technologies at the disposal of KBE systems for this purpose are identified. Further, a case study that demonstrates such use of a KBE system to realize the digital thread for a manufacturing assembly process is presented.**

I. INTRODUCTION

The manufacturing industry has come a long way. Traditionally, 2D drawings were used for design and manufacturing environments [1]. However, such an approach made routine activities such as maintainability, synchronicity, and checking for completeness and accuracy, cumbersome and prone to errors [2]. Further, the increasing complexity of today's products with increasing number of constituent components surpassed the ability of 2D drawings to represent information in a compact and meaningful manner [3] [4]. Manufacturing soon saw the emergence of an additional dimension to models used that, to a large extent, overcame the problems associated with 2D Models. The benefits with these 3D model-based definitions (MBD) were significant enough that they were used as authoritative sources of information for the lifecycle of the product in the model based enterprise (MBE) paradigm [3]. Conceptually the aggregation of model-based manufacturing (MBM) that uses MBDs to realize the value of a model-based enterprise is accomplished by the digital thread (Fig.1) [1]. The digital thread, in essence, is a framework that helps integrate product lifecycle data to streamline processes along all lifecycle stages. One of the earliest phases of the product lifecycle is the product design stage where MBD of the product (CAD model) is created. The digital thread in a MBE is purposed to provide a framework where this CAD model serves to aggregate all information associated with the product in order to turn them into actionable insights to optimize processes of the enterprise for all stakeholders [1].

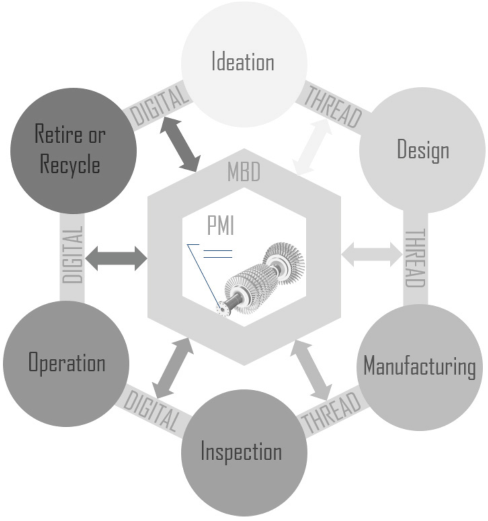


Fig. 1. Conceptual diagram depicting the role of model-based definition (MBD) in realizing the digital thread in a model-based enterprise (MBE)

Meanwhile, knowledge-based engineering lies at the juncture of these model-based definitions (CAD), object-oriented programming (OOP) and Artificial Intelligence (AI) [5]. The main goal of KBE systems is to capture product and process related information in order to "to capture and systematically reuse product and process engineering knowledge ..." [6]. Realizing digital threads in a manufacturing enterprise necessitates the capture of product information and delivering the right information to the right place at the right time [7]. While knowledge-based engineering systems were developed aimed at the capture and re-use of engineering design knowledge, its flexibility makes an appealing platform to realize digital thread frameworks in model-based enterprises. Also, capturing engineering knowledge can be considered an integral part realizing digital threads in many contexts [8] [9].

This study abductively reasons that if knowledge-based engineering systems allows for the capture of process and product engineering in model-based definitions, then it should be extendable to be an important enabler of the digital thread vision, which in itself exploits MBDs to manage the entire lifecycle of the product. The subsequent Background section introduces the concepts of digital thread and KBE systems.

The proposed four ways in which KBE systems could help realize digital thread frameworks are discussed in Section III. Section IV identifies important technology enablers that can be exploited by KBE systems to help achieve the digital thread vision. Finally, Section V presents a case study where a KBE system realizes the digital thread by exposing and augmenting a product design model with semantic descriptions for a robot to enable intelligent manufacturing. The paper is summarized in Section VI.

II. BACKGROUND

The concept of digital thread notionally integrates information and knowledge from traditionally siloed systems across different phases of the product lifecycle. What information is integrated and across what systems is largely dependant on what value the digital thread is purposed to bring in the context or domain it is implemented in. For example, realizing digital threads in the aerospace domain could be for affordability based analysis for trading performance with manufacturability, iterative design decisions under uncertainty or for reducing cost and time at the aircraft detailed design stage [9] [8] while in an additive manufacturing context it might be for part producibility, process repeatability, and part-to-part reproducibility [10] or supply-chain visibility in a logistical context [11].

Model-based enterprise envisions the use of (3D) models as the authoritative sources of information along the product lifecycle, i.e. as the digital thread (Fig. 1). Initially, these model-based definitions were annotated with product and manufacturing information (PMI) to serve downstream processes such as manufacturing and inspection and testing of the product. The PMI included geometric dimensions and tolerances (GD&T), material specifications, component lists, process specifications, and inspection requirements. While PMI has demonstrated value and extends the use of MBDs beyond the design stage, and thus can be considered necessary, it alone does not hold sufficient in downstream processes in many contexts. PMI lacks beyond geometry concepts that is one up the data-information-knowledge-wisdom (DIKW) hierarchy [12] in terms of knowledge about the product such as design intent and other functional, resource and process knowledge. Moreover, in many cases it is necessary to facilitate an upward information flow from downstream processes. This is particularly important in cases where operational data of the product needs to be fed back to design to foster better design decisions for the next generation of the product or for the creation of digital twins which constantly feeds on operational data. A digital thread implemented in such environments need real-time access to high-fidelity model definitions.

KBE systems, on the other hand, are essentially the merger of two class of systems that have been existing fairly independently prior to its rise. These are knowledge based systems (KBS) and engineering or conventional CAD systems [6]. To understand what KBE systems are it is important to understand knowledge-based systems (KBS) first. Knowledge-based systems came into being to relieve humans from making decisions in part. It does so by running reasoning procedures on known facts and domain knowledge in a knowledge base by an inference engine. However, they never found their way to the field of engineering design due to three main problems: formalizing design knowledge into rules for use by inference engines, their inability to perform geometry manipulations and their inability to perform complex computations [6]. Meanwhile CAD systems where adept at the latter two problems through specialized CAD kernels. Thus KBE systems came into being that combined the strengths of KBS (reasoning about facts and inferring knowledge) and that of engineering CAD systems (geometry manipulation and analysis) thereby realizing a system that allows for geometry manipulation and analysis while maintaining a formalized body of knowledge that relieves designers from repetitive design activities (Fig. 2). Conceptually, it took a simple step on the part of CAD vendors to provide such a functionality. The CAD kernel that drove the design process was exposed by means of an API that could be manipulated at will by the designer by a KBE language. *Knowledge Fusion* [13] is an example of one such language of a KBE system *Siemens NX*. Modern KBE vendors such as *Siemens NX* also further this flexibility by providing bindings in popular general purpose programming languages such as Java, C++, Python, .NET.

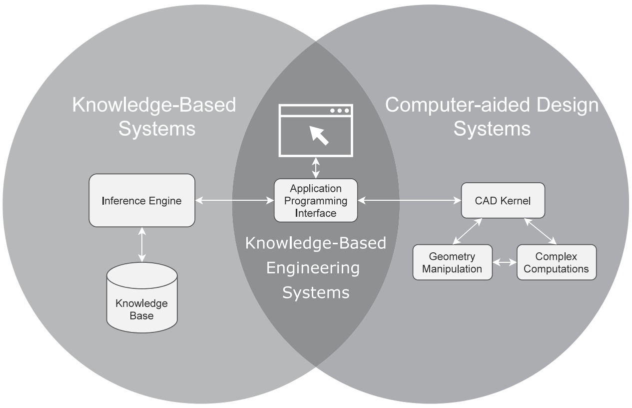


Fig. 2. Knowledge-based engineering systems as a merger of two class of systems

Knowledge-based engineering systems have evolved over the years and is headed towards maturity to have features capable of realizing digital threads in many contexts that serve a model-based enterprise. This study proposes that knowledge-based engineering systems, originally built to aid engineering the product, can be extended to serve as enablers the digital thread vision. As such this paper advocates its use as an integrator of federated knowledge throughout the product's value/supply chain through appropriate integration mechanisms made available by the flexibility of state of the art KBE systems. The next section proposes four ways how KBE systems can contribute to realize a manufacturing digital thread.

III. ENGAGING KBE SYSTEMS FOR REALIZING THE DIGITAL THREAD

What appeals most to the cause of digital threads in a KBE system is its ability to drive a CAD engine programatically via a general purpose programming language. Coupled with its mechanisms of knowledge representation and reasoning, there are theoretically no limits to how this exposes a MBD to a world of opportunities. This section proposes four directions how KBE tools may help contribute in realizing the digital thread in a manufacturing enterprise.

A. Knowledge Representation

In a model-based enterprise, model-based definitions act notionally as the digital thread of the product and different information are of interest by consumers of the model. These consumers may be humans, softwares or machines and interested information depends on the role of the consumer in the enterprise. Currently what information is represented throughout the lifecycle and how is governed by standards. For example, STEP is one such standard (ISO 10303) used for the exchange of product information throughout a product's lifecycle. It may include product and manufacturing information (PMI) such as geometric dimensioning and tolerancing (GD&T) information. However there may be many use cases where the information to be represented may go beyond those identified by the standards despite the comprehensive approaches taken by standardizing bodies. These could be contextual information that is not explicitly mentioned as dimensions or tolerances [14] and exist as a result of specific practices followed by an enterprise that may not be representative of a vast majority of them to reach the standardizing bodies or is not shared as they are considered trade secrets which gives them a competitive edge [4]. These (implicit) information may be important with respect to specific workflows adopted in these (minority) enterprises [14]. Further, it may be impossible to standardize representational schemas of knowledge that the workforce accumulates that can take different forms even with enterprises operating in the same domains.

All of these reasons calls for the need to be able to go beyond standards to represent and communicate non-standard product models at will of stakeholders to realize a true information thread addressing the need of specific enterprises. KBE systems allow just this and provides an environment for representation of geometry and beyond geometry user-defined knowledge. As an example, OntoSTEP [15] has been an approach towards developing semantically enhanced MBDs based on Web Ontology Language of the Semantic Web. With KBE systems, it would be possible to realize an application that integrates OntoSTEP at the design level and realize a digital thread to those functions of the enterprise that require these semantically enriched product models as a standalone run-time application. This paper provides such an example as a case study in Section V. OntoSTEP happens to be just one semantic product model and semantic models are just one example of class of models besides the conventional CAD design model but the case for the KBE systems here is that they allow for customer or need-specific beyond standard knowledge representation formalisms for exchange of product data with only availability of APIs being the limit.

B. Implementation of Trust Mechanisms

Employing digital threads garners more exposure for data for an otherwise “closed” data which can be subject of cyber attacks. In some cases, this may lead to revealing proprietary information and trade secrets that give one company a competitive edge over another. Data confidentiality is one of the major trust-related concerns of manufacturers envisioning the digital thread [16]. The nature of complex parts is such that many of its constituent parts are manufactured by vendors that specialize in manufacturing these sub-parts. A breach of trust in information relating to any of the sub-parts may provide valuable insights to the part it fits into and vice versa. [16].This also means that manufacturers must trust each other's trust mechanisms for protecting intellectual property. Further, design information are transmitted along the factory floor without any digital signatures. Thus, there is no guarantee that the ‘as designed’ plans have not been tampered with or even stolen by eavesdroppers. Its integration with the physical asset in the cyber physical space make it vulnerable to cyber threats that could directly translate to defective produced goods [17] or expose an entire industry considering the breadth of the digital thread across the supply chain [18] [19]. Thus, organizations must aim to strike a fair balance between data access and data confidentiality through appropriate trust mechanisms.

A digital thread realized using a KBE system can filter request for information based on identified actors it considers authentic and share geometry or part specific data of the product model accordingly. It may also choose to serve reduced order models [20] compliant with the enterprise data governance policies based on where the request for product information comes from. It also provides the ability to integrate a model-based definition to an enterprise-wide blockchain that can be used to trace the development of the product data along its lifecycle. Such mechanisms are implementable only by systems such as the KBE systems that allows programmatic manipulation of product model geometry and features.

C. Modelling and Analysis

Often is the case that complex models are created by a team rather than an individual and an environment to collaborate on such activities is not present for mainstream use. Lack of such environments that allow translation to a suitable computer representation poses a challenge to enable effective collaboration on such early or conceptual models. Lack of commercial solutions that offer cloud-based simulations has also been identified in the digital thread context [7]. The complex nature of the aerospace domain has garnered the attention of multidisciplinary design optimization (MDO) to realize a digital thread that supports decision making [21] [8]. Reuse of models must be advocated as a lot of time and expertise is involved in creating one. Mechanisms that support reuse, that aid in discovery and use of existing models need to develop further [7]. Further, myriad models exist in a digital thread framework and they can be of many types such as physics-based, mathematical, etc. The digital thread vision necessitates that these models co-exist to have a certain degree of interoperability to integrate data [7]. Sometimes, different properties of the same model may be of interest in different kinds of analysis. For example, in different properties of the same geometry in finite element models. The FEMs must be solved separately in an interoperable manner that shares results between them [22]. Further, these models are of varying degrees of fidelity that poses a challenge of its own to integration [23].

A digital thread application built using a KBE system would be capable of providing such design environments that allows for maintaining a repository of models. Mechanisms for finding different available models is essentially the matter of searching available local or remote repositories, a functionality common in any basic application. Where a KBE application has an advantage is the ability to load these models and serve them as integrated and interoperable models through user-defined endpoints and APIs. Programmatically KBEs have the ability to load different FEMs of the same part dynamically at run-time with varying degrees of fidelity. This important flexibility allows digital thread applications to maintain real-time constraints imposed in several contexts. Real-time'ness depends on the context. KBE systems could run such analyses on isolated parts of complex components by developing faster algorithms, code optimization and parallel computation [24] using powerful and efficient compiled languages such as C++ that modern KBE systems provide [6]. KBE systems have also been developed in the context of multidisciplinary design optimization where it has supported the integration of different analysis tools by generating disciplinary abstractions in an automated manner [25].

D. Knowledge capture and reuse

In the digital thread context, domain interoperability (between design, manufacturing, quality, etc.) suffers from the lack of robust mechanisms for contextualization [4]. Different domains involve different actors that perform activities in the capacity of different roles in domain specific environments. Data needs to be presented with respective contexts. These challenges are currently being addressed by file and format-based interoperability that are necessary but not sufficient. Semantic interoperability is not being sufficiently probed from a digital thread standpoint [4]. Further, autonomy is key to realize scalable systems, or in other words, the lifecycle cannot afford the human capital required to keep create and maintain knowledge bases. They need to attain certain degree of autonomy that discovers relationships between knowledge constructs and links in lifecycle data [26].

A digital thread realized with KBE systems can utilize libraries that help implement graph-based knowledge constructs such as Apache Jena for OWL [27] while generating design models from requirements or manufacturing process plans from design models as an example. Graph theory has been explored as a viable solution to enable contextual viewpoints along the lifecycle [26]. Sub-graphs of the graph can aid role-based viewpoints while graphs in the form of trees can aid decision making. KBE systems can be developed to treat knowledge-elements based on such contexts embracing a connectivist approach to knowledge management [28]. Using such approaches allow for viewing the same artefact, for example a user-defined feature in a CAD model, as a logical element, geometrical element or an element in a CNC process [28]. Dynamically generating these connections in order to prevent its laborious manual creation is reported as a future research direction [29] and is something digital threads realized with KBE systems can solve and is presented in part as a case study in this paper. Knowledge-reuse can also take the form of repetitive operations that need to take place. KBE systems can automate the monotonous repetitive tasks by design automation that KBE allows and generate dynamic knowledge bases by dynamically linking engineering, manufacturing and quality functions of the manufacturing enterprise to realize the digital thread [4].

IV. ENABLING TECHNOLOGIES

Several enabling technologies are identified in this section that knowledge-based engineering systems can be programatically integrated with to help realize digital threads in an enterprise. This section aims to provide a brief overview of these digital thread enablers and is neither intended to be a comprehensive explanation nor an exhaustive list. Example use of the technology in literature in a digital thread context is provided.

A. Handle System

The handle system [30] is a distributed information system that provides a mechanism to locate and access resources distributed over a network (for example, the internet). The resolution mechanism behind the handle system allows for the resources themselves to be changed in terms of its location and other state information while still being able to interact with it in a secured manner with both client and server authorization with support for data confidentiality and integrity. Each handle is globally unique and can refer to many instances of a resource that changes its location with time. Each handle can also refer to multiple attributes of a resource that could be used as entry points to services a resource has to offer.

From literature, the Lifecycle Information Framework and Technology (LIFT) framework uses the handle system to resolve location of digital artefacts with unknown locations [26]. LIFT furthers the technology to include connections to the physical world. For this, apart from the concepts of global handle registry and the local handle services that it borrows from the Handle System, it comprises also of agent-based adapters composed of microservices that interfaces on client support systems. These adapters are purposed to track activities within these systems and store its associated handles in a local graph database. The KBE system may be integrated with these agent-based adapters using any library that implements RESTful HTTP communication (`requests` [31] for python).

B. Blockchain, Cloud and the Industrial Internet of Things

Blockchain is a distributed ledger in a peer-to-peer network that essentially is a chain of blocks containing data or transactions [32]. Each block contains a cryptographic hash of the block in the chain which makes them immutable. Data in any of the blocks cannot be altered which requires consensus of the network majority thus providing a decentralized consensus mechanism. There are several properties of blockchain that makes it an intriguing prospect. It follows a decentralized architecture in that information is controlled by data owners rather than third parties and can be validated by a consensus mechanism. This eliminates intermediary costs, promotes data ownership and automates business rules and mining of transactions. Immutable transactions on the blockchain means the data across the network cannot be manipulated without the knowledge of associated stakeholders thus enforcing the idea of a single source of truth. Public-key encryption can keep transactions secure and its decentralized nature means there can be no single point of failure. Blockchain technology is expected to be a key technology to facilitate data trust and can be integrated with KBE systems to maintain data integrity and traceability issue in distributed and collaborative model environments.

Recent advances in communication technology strengthens the case for cloud infrastructures as well. Essentially, cloud refers to the availability of data storage, computation power and / or analytics on-demand [33]. Digital thread data, needless to say, builds very quickly and cloud infrastructures can be one way to go as it is built to scale. This removes the burden of digital thread data management away from manufacturers and outsource it to those adept at the task. This can help realize business models such as Infrastructure, platform, software and function-as-a-service models for the digital thread [34]. However, opponents of cloud technologies cite data security and privacy issues when integrating cloud-based architectures that need to be heavily researched on before opting for the same.

The Industrial Internet of things primarily helps extend the digital thread during the operational phases of the lifecycle giving insights into the as-used data. The internet of things as a technology in the digital thread context plays the role of ‘transport’ by getting the information from location A to location B (from sensors to a middleware, for example).

Examples of theses technologies employed in literature include the work of Li et.al [11] that provides a framework architecture based on a dynamic hybrid peer-to-peer network and a private/public blockchain data model that leverages internet of things to enable real-time visibility of the supply chain in the distribution phase. It is is based on a hybrid peer-to-peer network and a combination of private and semi-public blockchain ledgers. The hybrid P2P network consists of a central index server that indexes information regarding active peers while the blockchain ledgers record significant events that take place and serve for timely delivery of ground truth information to associated stakeholders. Adhikari proposes a solution for capturing and storing factory floor data based on blockchain and cloud technologies [35]. Hedberg et. al. use blockchain registered transactions for promoting data ownership [36].

C. Semantic Web Standards

The semantic web (SW) came into being with a vision of incorporating “more machine oriented semantic information, allowing sophisticated processing” [37] to the World Wide Web to transform it from a document-based Web focussed on people to a Web of structured data with formal semantics enabling computers as well to understand to work in cooperation with people. The first versions of resource description framework were published circa 2000s. Twenty years later a lot of progress has been seen with the semantic web and that has allowed it to find applications in many areas within the manufacturing context such as education [38] and resource descriptions [39].

The semantic web suite of tools, languages and standards has potential to be an important enabler of the digital thread vision. The semantic web provides a means for knowledge representation and reasoning. While knowledge representation formalisms include knowledge graphs using RDF/RDFS [40], OWL [41], reasoning mechanisms allow to derive implicit knowledge constructs from represented knowledge using rules by rule languages such as Semantic Web Rule Language (SWRL) [42]. Further, consistency of represented knowledge to check if they conform to expected schemas may be done by SHACL [43] and SPIN [44] standards. The semantic web also provides query languages for knowledge retrieval. SPARQL is a query language that can be used to retrieve information from RDF (or OWL) graphs and is a W3C recommendation. SQWRL [45] is another query language based on semantic foundation of SWRL that is purposed for querying OWL graphs where SPARQL [46] may fall short.

While semantic web languages can be used to represent knowledge and reason with it it can also be used to form shared conceptualizations of domain knowledge known as ontologies; which is essentially a formal description of knowledge and relationships of terms, concepts, resources within a domain. Such ontologies have the potential to form the cornerstone of semantic mediation and integration for the digital thread. Further, ontology matching and aligning techniques is an active research area can aid in forming an authoritative digital thread representation. From the digital thread standpoint, such ontologies can bootstrap the processes of knowledge discovery and integration. R2RML [47], a W3C recommendation is another language to help define mappings from relational databases to RDF which could be used across the digital thread to integrate legacy systems.

There has been an enormous amount of research involved over the last two decades and while a few languages and tools have been mentioned here, we have only discussed the

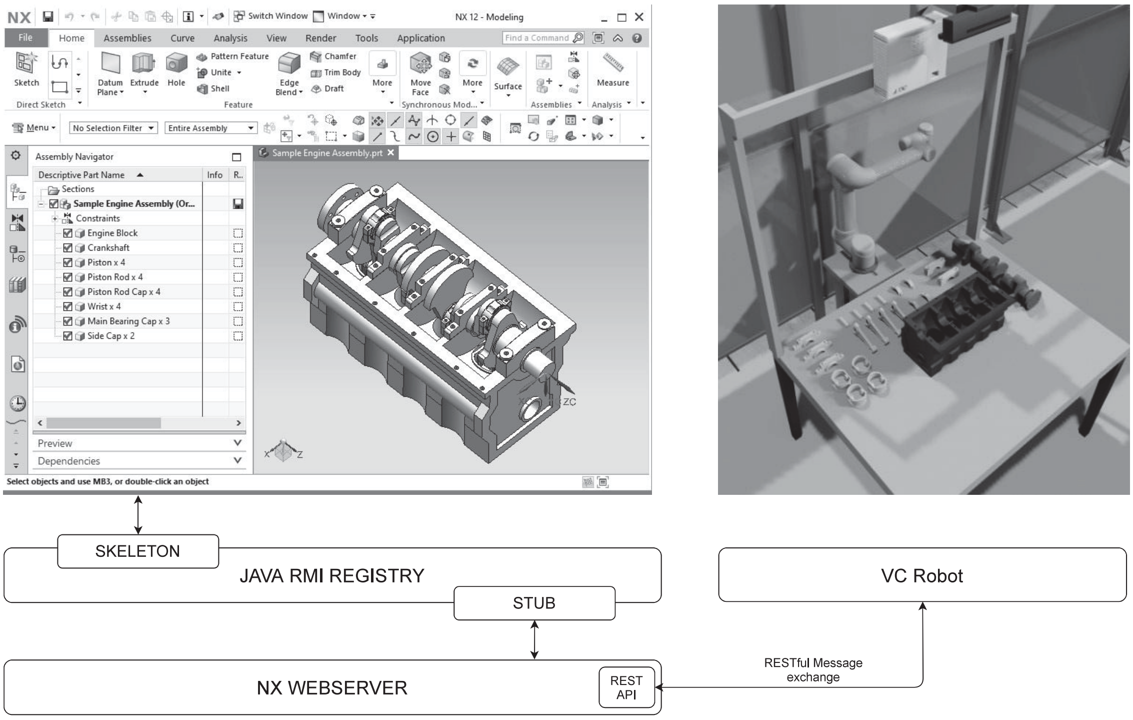


Fig. 3. Architecture of case study

proverbial tip of the iceberg. While some have reached to a W3C recommendation status, others remain as proposals and research topics but however may be used in implementations that advocates its use. Nevertheless, these standards are resourceful when it comes to enabling knowledge management and representation functionalities of KBE systems discussed in the previous section. Examples from literature include the Interoperability and Integration Framework (IoIF) framework [48] that aggregates data from Systems Modelling Language (SysML) models and transforms them to Web Ontology Language (OWL) format for checking consistency and completeness by aligning the ontology with a decision ontology. Shani et. al. [49] develop a semantic mediation container that mediates between OWL ontology models of model-based systems engineering tools.

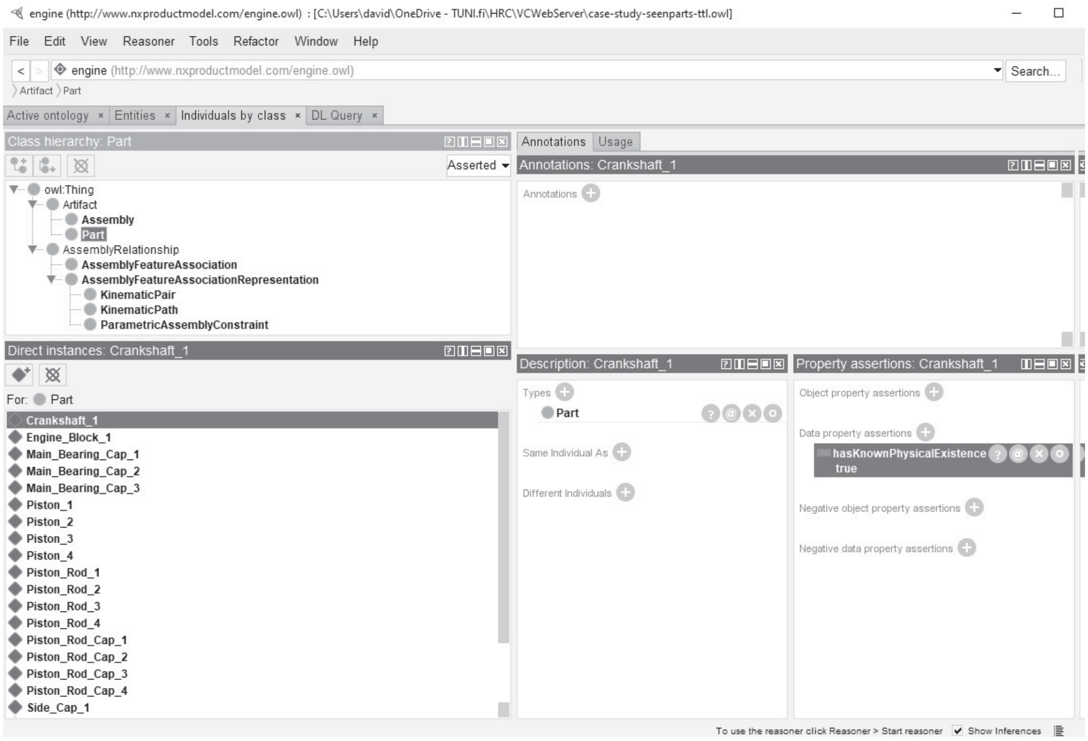
D. Visualization

A digital thread application realized via a KBE system could integrate product design information with external visualization techniques that emphasizes the semantics or context specific data adapted to tailored use-cases. Since data could be presented to both an expert audience (typically domain-specific) and the layman, it could allow for a more granular view of data that facilitates data-driven exploration and discovery. The challenge here is to present the context of an entity based on domain-specific roles. A wide variety of visualization techniques exist today that KBEs can programatically exploit that can be either web-based (using JavaScript libraries directly such as d3.js [50]) or tool based such as orange [51]. Some suite of tools work both as applications and the web, PowerBI [52] or Tableau [53] for example.

Example works in literature include that of Crowell et. al. [54] that builds a suite of tools using Jupyter notebooks to aid parsing, visualizing and analyzing build time sensor data. Pre-scripted Tableau visualizations are used to view the effect of changes in the IoIF framework [48]. Kallou et. al. [55] use web development technologies to aid decision making by use of analysis tools and visualizations. In a web-based open architecture lies the future of the digital thread visualization and it has already been proposed for Computer-aided technologies (CAx) [56]. Web apps work seamlessly cross-platform when designed responsive. Further, an open architecture allows the possibility to integrate with other tools needed to realize the vision of the digital thread. For example, the KBE system could exploit myriad file formats developed for the web that have not achieved mainstream adoption today. The system could achieve the model-based enterprise vision exploiting JavaScript libraries for 3D models on the web such as WebGL [57] or frameworks built on it. Support for immersive technologies on the Web by means of Javascript libraries such as WebXR [58] introduce augmented and virtual reality solutions to the enterprise as well.

V. CASE STUDY

This section presents a case study wherein a KBE system is used to realize a digital thread to make a robot aware of the product in a manufacturing process. The specific process is a manufacturing assembly process of an engine and the robot



Class hierarchy: Part

- owl:Thing
  - Artifact
    - Assembly
    - Part
  - AssemblyRelationship
    - AssemblyFeatureAssociation
    - AssemblyFeatureAssociationRepresentation
      - KinematicPair
      - KinematicPath
      - ParametricAssemblyConstraint

Annotations: Crankshaft_1

Direct instances: Crankshaft_1

For: Part

- Crankshaft_1
- Engine_Block_1
- Main_Bearing_Cap_1
- Main_Bearing_Cap_2
- Main_Bearing_Cap_3
- Piston_1
- Piston_2
- Piston_3
- Piston_4
- Piston_Rod_1
- Piston_Rod_2
- Piston_Rod_3
- Piston_Rod_4
- Piston_Rod_Cap_1
- Piston_Rod_Cap_2
- Piston_Rod_Cap_3
- Piston_Rod_Cap_4
- Side_Cap_1

Description: Crankshaft_1

Types: Part

Property assertions: Crankshaft_1

Data property assertions: hasKnownPhysicalExistence true

Fig. 4. OAM Ontology as viewed from Protégé GUI

exists in a virtual environment (Visual Components). However, the case study can be realized on a real robot as well. Fig. 3 depicts the overall architecture. The KBE system is Siemens NX and exposes the product assembly via a WebServer that wraps the JAVA stub of an *NXSession* Object exported via JAVA RMI. The robot communicates with the KBE system via a REST interface of web server.

The KBE system enriches the product CAD Model using semantic descriptions. Specifically we use a subset of the Open Assembly Model (OAM) developed at NIST [59] to automatically populate the sub-assembly parts using Siemens NXOpen API. The Robot that shares the OAM as part of its beliefs of the product uses Shapes Constraint Language (SHACL) for validating its beliefs against a pre-defined constraint that allows it to start with the manufacturing assembly process only if all the sub-assembly parts are present.

Fig. 4 shows the classes of the subset of the NIST Ontology re-used in the *Class Hierarchy* Window in the Protégé ontology editor GUI[1]. The Robot requests this semantic assembly model from the KBE system which updates the ontology with individuals of the `Part` class based on the product design model and sends it to the robot. The individuals are shown in the `Part` Class *Direct Instances* Window. As can be seen, the names match with the individual components in the Assembly Navigator Window of the NX software shown in Fig. 3.

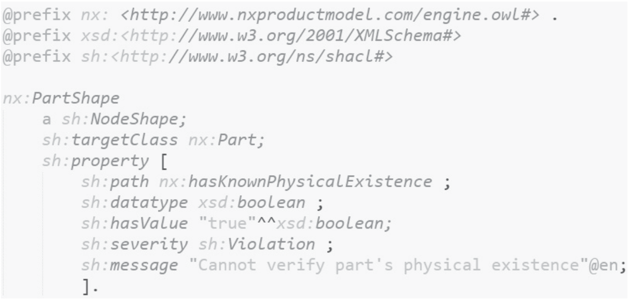
The robots perception capabilities allow it to identify the parts that are available to it on top of the table. While for the purposes of the case study this is done programatically via an API in the virtual environment, in a real-world scenario this would be by recognizing a visual marker (e.g. barcode) or through matching point cloud data with a suitable sensor. The robot then updates its beliefs about what all it sees. This is done here by setting the OWL Data Property `nx:hasKnownPhysicalExistence` to `True`. This is shown as a data property assertion in the *Property assertions* Window in Fig. 4.

The robot validates its beliefs based on its understanding of the world. This is accomplished by SHACL graphs. The shapes graph associated with the validation in this study is shown in Fig. 5. The target for the node shape `nx:PartShape` is the set of all SHACL instances of the class `nx:Part` which is specified using the property `sh:targetClass`. During validation, these (`Part`) instances become focus nodes for the shape and are validated against the constraints set by the blank node property shape (in square paranthesis []) declared by `sh:property`.

The property shape defines a constraint on the property `nx:hasKnownPhysicalExistence` (indicated with `sh:Path`). Two constraints are specifically imposed by this propery shape. First, the value is of type `xsd:boolean` (indicated by `sh:datatype` predicate) and that the value is `true`

---

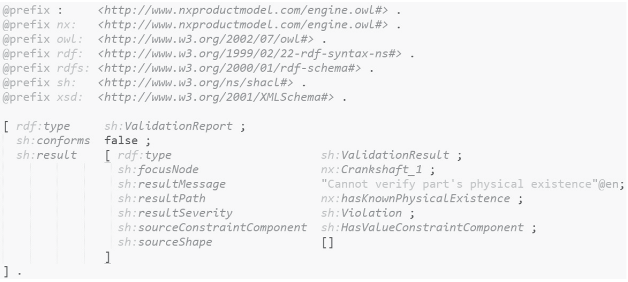
[1] Note that the Protege GUI represents the ontology axioms at the end of all the above mentioned communication. At no point is it needed to use protege during run-time and is only used here as a familiar way to visualize the ontology. All interactions above are accomplished dynamically at run-time with Apache Jena [27].



```
@prefix nx: <http://www.nxproductmodel.com/engine.owl#> .
@prefix xsd:<http://www.w3.org/2001/XMLSchema#>
@prefix sh:<http://www.w3.org/ns/shacl#>

nx:PartShape
    a sh:NodeShape;
    sh:targetClass nx:Part;
    sh:property [
        sh:path nx:hasKnownPhysicalExistence ;
        sh:datatype xsd:boolean ;
        sh:hasValue "true"^^xsd:boolean;
        sh:severity sh:Violation ;
        sh:message "Cannot verify part's physical existence"@en;
        ].
```

Fig. 5. SHACL Node Shape defining Property Constraints



```
@prefix :     <http://www.nxproductmodel.com/engine.owl#> .
@prefix nx:   <http://www.nxproductmodel.com/engine.owl#> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .

[ rdf:type     sh:ValidationReport ;
  sh:conforms  false ;
  sh:result    [ rdf:type                      sh:ValidationResult ;
                 sh:focusNode                  nx:Crankshaft_1 ;
                 sh:resultMessage              "Cannot verify part's physical existence"@en;
                 sh:resultPath                 nx:hasKnownPhysicalExistence ;
                 sh:resultSeverity             sh:Violation ;
                 sh:sourceConstraintComponent  sh:HasValueConstraintComponent ;
                 sh:sourceShape                []
               ]
] .
```

Fig. 6. A negative validation Result after the Robot validates its beliefs with the shapes graph

(indicated by `sh:hasValue` predicate). In other words, if the robot does not acknowledge the physical existence of the any of the parts in its beliefs constructs, a violation is raised along with the message specified by `sh:message`. If all parts are present, the beliefs conforms to the shapes graph. The validation report serialized in turtle showing these cases is shown in Fig. 6 and Fig. 7.

This simple case study demonstrates how knowledge residing in a product design model can be exploited by real-world agents by a manufacturing digital thread realized by the flexibility offered by modern KBE softwares. Programmatic access to CAD is powerful as seen, can be coupled with many 3rd party libraries to streamline manufacturing processes. For example, the case study uses Apache Jena library [27] with for semantic annotation of the product model. Specifically, the KBE system traverses the assembly tree and populates the OWL ontology automatically which is recognized by the robot. The case study is oversimplified to demonstrate mechanisms of knowledge capture and representation while in real-word scenarios, a lot of other factors would also have to be accounted for. For example, it might be okay for the robot not to take physical cognizance of certain sub-assembly parts that it does not interact with during the assembly or some part may become available only at the time it is required. Nevertheless, it demonstrates how information residing in product design models can be exposed to enable intelligent manufacturing by using knowledge representation and validation mechanisms provided by state of the art KBE tools.

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .

[ rdf:type     sh:ValidationReport ;
  sh:conforms  true
] .
```

Fig. 7. A positive validation Result after the Robot validates its beliefs with the shapes graph

VI. CONCLUSION AND FUTURE WORK

While the goal of KBE systems is to support, automate and optimize design activities, this paper builds on an initial abductive reasoning that their programmatic flexibility that allows them to do so could also make them a lucrative prospect as an integral point for multi-disciplinary product life-cycle knowledge for realizing manufacturing digital threads. To support this, four directions are proposed as to how this may be realized and the study identifies important supportive enabling technologies. These technologies act as useful tools in deploying KBE systems in a model-based enterprise to integrate lifecycle information realizing digital threads. The case study builds further to demonstrate the use of one such technology, i.e. semantic web technology, with a design model of a product in a KBE system at the design stage integrated with the processes in its subsequent life cycle stage, i.e. manufacturing. What would otherwise be a CAD model residing in a closed product data management system (PDM), the digital thread realized by the KBE system makes the robot product-aware and intelligent, and help with its decisions in a manufacturing assembly process use-case. Although, case studies involving all technologies discussed are not presented, the conclusion we arrive to from the initial retroduction is that KBE systems can indeed be useful for realizing digital threads in manufacturing. The study is precursor to research that employs a KBE system as an integral part of a digital thread framework in a collaborative assembly environment between a human and robot which remains as future work.

ACKNOWLEDGMENT

The research leading to these results has been funded from the Doctoral School of Tampere University, Finland to whom the authors are grateful.

REFERENCES

[1] T. Hedberg, J. Lubell, L. Fischer, L. Maggiano, and A. B. Feeney, “Testing the digital thread in support of model-based manufacturing and inspection,” *Journal of Computing and Information Science in Engineering*, vol. 16, no. 2, pp. 1–10, 2016.

[2] J. Lubell, K. Chen, J. Horst, S. Frechette, and P. Huang, “Model Based Enterprise / Technical Data Package Summit Report,” Tech. Rep., 2012. [Online]. Available: http://dx.doi.org/10.6028/NIST.TN.1753http://nvlpubs.nist.gov/nistpubs/TechnicalNotes/NIST.TN.1753.pdf

[3] S. P. Frechette, “Model Based Enterprise for Manufacturing,” *44th CIRP Conference on Manufacturing Systems*, no. December, 2011.

[4] T. D. Hedberg, N. W. Hartman, P. Rosche, and K. Fischer, “Identified research directions for using manufacturing knowledge earlier in the product life cycle,” *International Journal of Production Research*, vol. 55, no. 3, pp. 819–827, feb 2017.

[5] C. B. Chapman and M. Pinfold, “Design engineering - a need to rethink the solution using knowledge based engineering,” *Knowledge-Based Systems*, vol. 12, no. 5-6, pp. 257–267, 1999.

[6] G. L. Rocca, “Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design,” *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 159–179, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.aei.2012.02.002

[7] G. Shao and D. Kibira, “Digital manufacturing: Requirements and challenges for implementing digital surrogates,” in *Proceedings - Winter Simulation Conference*, vol. 2018-Decem. Institute of Electrical and Electronics Engineers Inc., jan 2019, pp. 1226–1237.

[8] A. Gharbi, D. Sarojini, E. Kallou, D. Harper, V. Petitgenet, D. Rancourt, S. Briceno, and D. Mavris, “Standd: A single digital thread approach to detailed design,” in *AIAA SciTech Forum - 55th AIAA Aerospace Sciences Meeting*, 2017.

[9] V. Singh and K. E. Willcox, “Engineering design with digital thread,” *AIAA Journal*, vol. 56, no. 11, pp. 4515–4528, 2018.

[10] D. Kim, P. Witherell, Y. Lu, and S. Feng, “Toward a digital thread and data package for metals-additive manufacturing,” *Smart and Sustainable Manufacturing Systems*, vol. 1, no. 1, 2017.

[11] Z. Li, H. Wu, B. King, Z. Miled, J. Wassick, and J. Tazelaar, “On the Integration of Event-Based and Transaction-Based Architectures for Supply Chains,” in *Proceedings - IEEE 37th International Conference on Distributed Computing Systems Workshops, ICDCSW 2017*, 2017, pp. 376–382.

[12] A. Lobov, “Smart manufacturing systems: Climbing the dikw pyramid,” in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 4730–4735.

[13] “Siemens Documentation: Introduction to KBE.” [Online]. Available: https://docs.plm.automation.siemens.com/tdoc/nx/12/nx_api#uid:index_fusion:id1395376:intro_kbe_ov

[14] A. M. D. Miller, N. W. Hartman, T. Hedberg, A. B. Feeney, and J. Zahner, “Towards identifying the elements of a Minimum Information Model for use in a Model-Based Definition,” *ASME 2017 12th International Manufacturing Science and Engineering Conference, MSEC 2017 collocated with the JSME/ASME 2017 6th International Conference on Materials and Processing*, vol. 3, no. June, 2017.

[15] R. Barbau, S. Krima, X. Fiorentini, S. Rachuri, A. Narayanan, S. Foufou, and R. D. Sriram, “OntoSTEP: Ontology for product model data,” *IEEE Transactions on Knowledge and Data Engineering*, no. Iso 10303, 2010.

[16] A. Adhikari, A. Hojjati, J. Shen, J.-T. Hsu, W. King, and M. Winslett, “Trust Issues for Big Data about High-Value Manufactured Parts,” in *Proceedings - 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, 2016, pp. 24–29.

[17] N. Gupta, A. Tiwari, S. T. Bukkapatnam, and R. Karri, “Additive Manufacturing Cyber-Physical System: Supply Chain Cybersecurity and Risks,” *IEEE Access*, vol. 8, pp. 47 322–47 333, 2020.

[18] M. L. Aller, T. Franta, and H. V. Helldorff-Garn, *AMTech CAPE Turbine Materials Standards Road Map: An Industry-Led Road Map Supporting the Next Generation of Power and Propulsion Manufacturing, Maintenance and Repair*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2018-1711

[19] J. Mantle, J. Rammohan, E. Romantseva, J. Welch, L. Kauffman, J. McCarthy, J. Schiel, J. Baker, E. Strychalski, K. Rogers, K. Rogers, and K. Lee, “Cyberbiosecurity for biopharmaceutical products,” *Frontiers in Bioengineering and Biotechnology*, vol. 7, no. MAY, 2019.

[20] E. M. Kraft, “Approach to the development and application of a digital thread/digital twin authoritative truth source,” in *2018 Aviation Technology, Integration, and Operations Conference*, 2018.

[21] D. Siedlak, O. Pinon, P. Schlais, T. Schmidt, and D. Mavris, “A digital thread approach to support manufacturing-influenced conceptual aircraft design,” *Research in Engineering Design*, vol. 29, no. 2, pp. 285–308, 2018.

[22] E. J. Tuegel, “The airframe digital twin: Some challenges to realization,” *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, no. April, pp. 1–8, 2012.

[23] A. Gharbi, S. Briceno, and D. N. Mavris, “Enabling the digital thread in commercial aircraft companies,” in *2018 Aviation Technology, Integration, and Operations Conference*, 2018.

[24] D. Marinkovic and M. Zehn, “Survey of finite element method-based real-time simulations,” *Applied Sciences*, vol. 9, no. 14, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/14/2775

[25] J. Sobieszczanski-Sobieski, A. Morris, M. J. van Tooren, G. La Rocca, and W. Yao, *Multidisciplinary Design Optimization Supported by Knowledge Based Engineering*, 2015.

[26] T. D. Hedberg, M. Bajaj, and J. A. Camelio, “Using graphs to link data across the product lifecycle for enabling smart manufacturing digital threads,” *Journal of Computing and Information Science in Engineering*, pp. 1–29, 2019.

[27] “Siemens Documentation: Introduction to KBE.” [Online]. Available: https://docs.plm.automation.siemens.com/tdoc/nx/12/nx_api#uid:index_fusion:id1395376:intro_kbe_ov

[28] J. Johansson, M. Contero, P. Company, and F. Elgh, “Supporting connectivism in knowledge based engineering with graph theory, filtering techniques and model quality assurance,” *Advanced Engineering Informatics*, vol. 38, no. October 2017, pp. 252–263, 2018. [Online]. Available: https://doi.org/10.1016/j.aei.2018.07.005

[29] T. Hedberg Jr, “Enabling Connections in the Product Lifecycle using the Digital Thread,” Ph.D. dissertation, 2018.

[30] “HANDLE.NET (version 9),” Corporation for National Research Initiatives, Tech. Rep., 2018. [Online]. Available: http://hdl.handle.net/20.1000/112

[31] “Requests: HTTP for Humans™ — Requests 2.26.0 documentation.” [Online]. Available: https://docs.python-requests.org/en/master/

[32] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, “Blockchain,” *Business and Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, 2017. [Online]. Available: www.proofofexistence.

[33] B. Hayes, “Cloud computing,” *Commun. ACM*, vol. 51, no. 7, p. 9–11, Jul. 2008. [Online]. Available: https://doi.org/10.1145/1364782.1364786

[34] J. Wang, C. Xu, J. Zhang, J. Bao, and R. Zhong, “A collaborative architecture of the industrial internet platform for manufacturing systems,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, no. April 2019, 2020.

[35] A. Adhikari and M. Winslett, “A Hybrid Architecture for Secure Management of Manufacturing Data in Industry 4.0,” in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2019*, 2019, pp. 973–978.

[36] T. D. Hedberg, S. Krima, and J. A. Camelio, “Method for Enabling a Root of Trust in Support of Product Data Certification and Traceability,” *Journal of Computing and Information Science in Engineering*, vol. 19, no. 4, 2019.

[37] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret, “The world-wide web,” *Commun. ACM*, vol. 37, no. 8, p. 76–82, Aug. 1994. [Online]. Available: https://doi.org/10.1145/179606.179671

[38] J. S. David, “Development of a digital twin of a flexible manufacturing system for assisted learning,” Master’s thesis, 2018.

[39] E. Järvenpää, N. Siltala, O. Hylli, and M. Lanz, “The development of an ontology for describing the capabilities of manufacturing resources,” *Journal of Intelligent Manufacturing*, vol. 30, 02 2019.

[40] “RDF - Semantic Web Standards.” [Online]. Available: https://www.w3.org/RDF/

[41] “OWL - Semantic Web Standards.” [Online]. Available: https://www.w3.org/OWL/

[42] “SWRL: A Semantic Web Rule Language Combining OWL and RuleML.” [Online]. Available: https://www.w3.org/Submission/SWRL/

[43] “Shapes Constraint Language (SHACL).” [Online]. Available: https://www.w3.org/TR/shacl/

[44] “SPIN - Overview and Motivation.” [Online]. Available: https://www.w3.org/Submission/spin-overview/

[45] M. O’Connor and A. Das, “Sqwrl: a query language for owl,” 01 2009.

[46] “R2RML: RDB to RDF Mapping Language.” [Online]. Available: https://www.w3.org/TR/r2rml/

[47] “R2RML: RDB to RDF Mapping Language.” [Online]. Available: https://www.w3.org/TR/r2rml/

[48] M. Bone, M. Blackburn, B. Kruse, J. Dzielski, T. Hagedorn, and I. Grosse, “Toward an Interoperability and Integration Framework to Enable Digital Thread,” *Systems*, vol. 6, no. 4, p. 46, dec 2018. [Online]. Available: http://www.mdpi.com/2079-8954/6/4/46

[49] U. Shani, S. Jacobs, N. Wengrowicz, D. Dori, and N. Wengrowiz, “Engaging ontologies to break MBSE tools boundaries through semantic mediation,” *Conference on Systems Engineering Research*, pp. 1–10, 2016. [Online]. Available: http://esml.iem.technion.ac.il/

[50] “D3.js - Data-Driven Documents.” [Online]. Available: https://d3js.org/

[51] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan, "Orange: Data Mining Toolbox in Python," *Journal of Machine Learning Research*, vol. 14, pp. 2349–2353, 2013. [Online]. Available: http://jmlr.org/papers/v14/demsar13a.html

[52] "Data Visualization — Microsoft Power BI." [Online]. Available: https://powerbi.microsoft.com/en-us/

[53] "Tableau." [Online]. Available: https://www.tableau.com/

[54] A. M. Crowell, S. A. Cramer, J. P. Lopez, E. K. Rashka, R. C. Reuss, V. K. Wordsworth, and G. C. Lewin, "Development of a quality assurance tool for additive manufacturing," *2018 Systems and Information Engineering Design Symposium, SIEDS 2018*, pp. 76–81, 2018.

[55] E. Kallou, A. Gharbi, S. Briceno, and D. Mavris, "Application of data science visualization techniques to enhance structural failure analysis," in *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018*, no. 210049, 2018.

[56] A. Lobov, A. Martikkala, and P. Biswas, "Towards a web-based framework for computer-aided manufacturing," 12 2019, pp. 59–65.

[57] "WebGL Overview - The Khronos Group Inc." [Online]. Available: https://www.khronos.org/webgl/

[58] "WebXR Device API." [Online]. Available: https://www.w3.org/TR/webxr/

[59] M. M. Baysal, U. Roy, R. Sudarsan, R. D. Sriram, and K. W. Lyons, "THE OPEN ASSEMBLY MODEL FOR THE EXCHANGE OF ASSEMBLY AND TOLERANCE INFORMATION: OVERVIEW AND EXAMPLE M." pp. 1–12, 2004.