Business context-based approach for Digital Twin services integration

Elena Jelisic[*,***], Marija Jankovic[**], Nenad Ivezic[***], Boonserm Kulvatunyou[***], Dionysios Kehagias[**], Zoran Marjanovic[*]

[*] Faculty of organizational sciences, Belgrade, Serbia
[**] Centre for Research and Technology Hellas, Thessaloniki, Greece
[***] National Institute of Standards and Technology, Gaithersburg, USA
elena.jelisic@nist.gov, jankovicm@iti.gr, nenad.ivezic@nist.gov, boonserm.kulvatunyou@nist.gov, diok@iti.gr, zoran.marjanovic@fon.bg.ac.rs

***Abstract*—Digital Twins (DTs) are among the most popular and quickly evolving technologies, particularly within Industry 4.0. The rapid expansion of a new generation of information technologies, such as the Internet of Things (IoT), artificial intelligence (AI), and cloud computing, results in new requirements for DT modeling. Consequently, a five-dimensional DT model is emerging, focusing on servitization, but with limited consideration of standardization for integration of the services. In this paper, we propose a new, business context-based approach that enables accurate and fast integration of DT services, contributing to the standardization of DT integration specifications. Furthermore, we present an initial validation result from the aerospace domain.**

***Keywords*—digital twin; services integration; business context.**

I. Introduction

These days, digital twins are one of the most sought-after technologies. An accurate digital twin (DT) model is a critical element for a successful DT implementation. Initially, three-dimensional digital twin models were proposed [1]. However, new requirements continue to emerge as we move forward with the ongoing technological developments and the introduction of a new generation of information technologies, such as the Internet of Things (IoT), artificial intelligence (AI), edge computing, and cloud computing. To fulfill these new requirements, the authors in [2] proposed a five-dimensional DT model that emphasizes the importance of services and data. As part of this new design, servitization plays a prominent role. By offering the potential for achieving interoperable services, coupled with their platform independence, servitization is an attractive approach for integrating innovative manufacturing applications and collaborative processes. However, there may be differences in data models, interfaces, and communication protocols underlying various physical and virtual services within DT, resulting in issues with data exchange, synchronization, and module integration [3]. Therefore, it is imperative that standardization efforts are aimed at resolving the above-mentioned issues. This paper describes a novel business context-based approach for DT services integration to help in achieving the missing standardization capabilities. For that purpose, concepts derived from Core Components Technical Specification (CCTS) [4] are employed, as well as the Enhanced UN/CEFACT [5] business context model. The paper demonstrates usage of the proposed approach in the domain of the aerospace industry and presents an initial validation result. The remaining parts of the paper are structured as follows. Section II introduces the needed background. Section III describes the proposed approach. Section IV offers an initial validation example. Section V contains a discussion of the results and future work. Section VI completes the paper with concluding thoughts.

II. Background

A. Digital Twin services

The smart manufacturing industry is increasingly paying attention to digital twins due to their enormous growth potential [10]. In his presentation on product lifecycle management at the University of Michigan in 2003, Michael Grieves introduced the concept of the traditional three-dimensional digital twin for the first time [1]. Originally described by Grieves, each *three-dimensional DT* consists of three components: (i) a *physical entity* situated in a physical space; (ii) a *virtual entity* situated in a virtual space; and (3) a *data and information link* connecting the two entities [1]. A novel five-dimensional DT model is introduced in [2], which adds DT data and services to keep up with continued technological advances in the IT domain. Based on the novel model, DT can be viewed in five dimensions: *physical entities (PE)*, *virtual entities (VE)*, *services* for PE and VE, *DT data*, and the *connections* that link all these entities together [2]. Compared to the previous three-dimensional DT model, the five-dimensional DT model benefits from data fusion, on-demand access to data, and a wide array of applications.

Servitization and standardization of DT. The manufacturing industry has undergone several changes that have contributed to servitization over the years [6]. Nowadays, manufacturing resources can be virtualized during the entire product lifecycle and then delivered as a service [6]. Researchers in [7] argue that it is essential to incorporate DT functions into standard services to promote easy and convenient collaboration across distributed enterprise systems. Additionally, they stress the need for increased standardization efforts to address the problems caused by the diversity of interfaces, data exchange standards, and implementation technologies.

According to Tao et al. servitization is an indispensable step in DT modeling since standardized services can provide easy and convenient access to advanced manufacturing functions such as simulation, data integration and fusion, production, optimization, etc. [3]. In [8], the authors discuss in detail the servitization of physical entities, virtual entities, and data in DT. In a study on data servitization, Tao et al. point out that data are often difficult to obtain and understand, because of various standards and user interfaces [9]. A uniform way of encapsulating services is essential to ensure effective DT [10]. According to Qi et al., the most important step in service encapsulation is to establish a unified information template that describes important aspects such as underlying physical objects, QoS, capacities, inputs, and outputs [10]. The research on DT encapsulation and standards-based representation is still at an early stage.

B. *CCTS*

Core Components Technical Specification (CCTS) is an ISO-approved meta-model standard that provides a basis for Data Exchange Standard (DES) development and usage. CCTS defines two building component types. The first, is a Core Component (CC) type that can be used to develop the structure of the DES. To support different granularity levels of the DES, different types of CCs are offered—*Aggregate CC (ACC)*, *Basic CC (BCC)*, and *Association CC (ACC)* type. Second, a *Business Information Entity (BIE)* type provides support for a DES usage specification. There is a corresponding BIE type for each CC type. The BIE can be understood as a profile created over the underlying CC which is restricted for the integration use case at hand. Each CC and BIE has an associated list of attributes—definition, data type, constraints, etc. Further, these attributes will be denoted as *component specification*. To define the integration use case for which the BIE is created, each BIE has an associated business context expression. This process is called BIE *contextualization*. Business context is specified by a list of categories that describe a specific aspect of the integration use case (e.g., business process, business process role, industry, geopolitical location). Each category has one or more associated schemes (e.g., ISIC [11] for industry) that provides a source of possible values for that category. A list of identified CCs and BIEs is denoted as *Core Components Library* (*CCL*), while business context schemes and their values make the *Business context knowledge base*. An enhancement of the CCTS contextualization mechanism was proposed in [12] which introduces new types of business context expressions. The first is *Assigned business context*, which identifies an integration use case for which the corresponding BIE is created. The second—*Overall business context*—is calculated, and it determines the cumulative business context for which an aggregate or association BIE is valid. The third—*Effective business context* —is also calculated and it enables filtering those BIEs that are valid for some requested integration use case at hand (i.e., *Requested business context*). The *Effective business context* is calculated as an intersection between an *Overall* and a *Requested* business context. A BIE will be considered relevant for a *Requested business context* if such an intersection results in a nonempty set.

C. *E-UCM business context model*

In literature there are different models for representation and usage of the business context knowledge base [5,13,14]. For demonstration purposes, Enhanced UN/CEFACT (E-UCM) [5] business context model is employed. According to E-UCM, a business context knowledge base is represented as an acyclic directed graph (DAG). There, each business context scheme is presented as a hierarchical subgraph; its root node is the scheme itself, while descendent nodes are values defined in that scheme. The usage of the business context knowledge base is supported by a list of predicates and operators to construct business context expressions. The E-UCM adopted all predicates and operators from the original UN/CEFACT business-context model (UCM) and introduced two new operators.

III. PROPOSED APPROACH

The proposed approach contains two main modules—CCL management and Mapping management—that enable 1) a DT services specification standardization and 2) a standards-based usage specification leading to a specific integration solution. Details for each of these modules are provided in following sections.

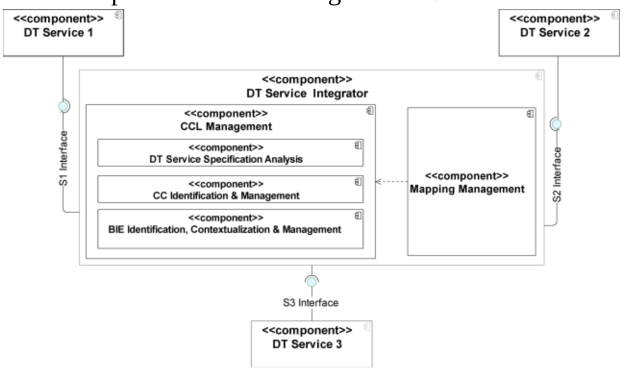

Figure 1. Digital Twin service integrator

A. *CCL management*

The purpose of the ***CCL management*** module is to manage the identified CCs and BIEs based on the service interface definition that includes the message specification definition. The module should assure component reuse, appropriate component specification and contextualization. The CCL management has three important functions: DT service specification analysis, CC identification and management, and BIE identification, contextualization, and management.

The ***DT service specification analysis*** function enables identification of important requirements for a specific DT service. These requirements reveal the constraints (e.g., data types, data semantics, formats, a list of possible values) that are applied on a specific message type that is sent by the DT service.

The ***CC identification and management*** function abstracts previously identified service requirements to construct the list of CCs. This function consists of two subfunctions—CC identification and CC management. The CC identification subfunction ensures that no duplicate CCs are stored in the CCL. For that purpose, the specification of a *candidate CC* component (to be inserted or updated) is analyzed. If the CC identification subfunction concludes that there is no such CC with the same specification as candidate CC, then the CC management subfunction will insert a candidate CC. Otherwise, if existing CC has been identified, the CC management subfunction will reuse it and update its specification if necessary (i.e., additional component specification is identified that will be important for future (re)uses).

The ***BIE identification, contextualization, and management*** function creates BIEs for previously identified CCs. These BIEs reflect all collected (not abstracted) DT service requirements. This function has three BIE subfunctions—identification, contextualization, and management. The first and the third are similar to those previously defined for CC identification and management. The BIE identification assures that there are no duplicates stored in the CCL. Similarly, the specification of a *candidate BIE* (to be inserted or updated) is analyzed. If the BIE identification subfunction concludes that there is no such BIE with the same specification as candidate BIE, then the BIE contextualization subfunction will contextualize a candidate BIE, and the BIE management subfunction will insert that BIE in the CCL. Otherwise, if an existing BIE has been identified, the BIE contextualization subfunction will recontextualize the BIE, and the BIE management subfunction will reuse the BIE and update its specification if necessary. Recontextualization means that the business context expression has to be recalculated to cover the newly identified integration use case for which the BIE is applicable.

B. Mapping management

The purpose of the ***Mapping management*** module is to provide a standardized basis for DT services integration. The mapping activity is inevitable because different DT services have different specifications and, consequently, different representations and interpretations of exchanged messages. However, these mappings are usually created in ad-hoc, manual, and non-standardized manner which has a negative impact on DT services integration and data exchange. By introducing CCs as a common basis for the DT services specifications, the proposed approach enables standardized and decentralized mapping specifications. The primary function of this module is to define the BIE → CC mapping specification, which occurs at design time. This function is key to achieving services integration via a common component specification. Every time a new BIE is created by the CCL management module, a corresponding BIE → CC mapping specification is defined.

IV. DEMONSTRATION

A. Use case description

For demonstration purposes, a use case from an aircraft engine maintenance domain will be employed. In order to predict any necessary maintenance of an aircraft engine or to anticipate its outage, the physical model of the aircraft engine is equipped with different sensors that should collect key data that give information about the engine's operating status, such as oil pressure, oil temperature, fuel starvation, detected smoke, leaks, etc. A report with all detected data is sent to the aircraft engine virtual model that employs machine learning algorithms to predict future engine behavior. The structure of one such report can be expressed using XML Schema as presented in Fig. 2.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema ...>
    <xs:element name="AircraftEngineReport">
        <xs:complexType>
            <xs:sequence>
                ...
                <xs:element name="EngineIdentification">
                    <xs:complexType>
                        <xs:sequence>
                            ...
                            <xs:element name="LastMaintenanceDate" type="EUDateFormat"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Measures">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="OilPressure" type="xs:string"/>
                            <xs:element name="OilTemperature" type="Celsius"/>
                            ...
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:simpleType name="Celsius">...</xs:simpleType>
    <xs:simpleType name="EUDateFormat">...</xs:simpleType>
</xs:schema>
```

Figure 2. An excerpt from an aircraft engine service specification (physical model)

If any problem is predicted by the virtual model, the state of the engine will be changed accordingly (e.g., state 'out of service' in case that outage is predicted). It is important to emphasize that the structure of the report can differ depending on the engine type, manufacturer, embedded sensors, etc. Also, different DT services provide different message types with different structures.

B. Business context knowledge base

In order to provide a basis for the integration of different DT services, a business context knowledge base has to be constructed. This knowledge base enables identification of the source of the collected data by associating them to a specific business context expression. For demonstration purposes, only two business context categories are chosen—Business Process and Business Process Role. Also, two corresponding schemes will be assigned to these categories to identify activities inside the aircraft engine maintenance collaborative business process, as well as the system roles that exchange messages within that business process. A portion of the business context knowledge base is presented in Fig. 3.

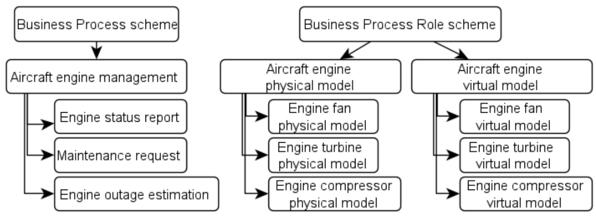

Figure 3. A portion of business context knowledge base

Using the provided business context knowledge base, a report previously presented in Fig. 2 can be described using the following values: (1) *Engine status report* for Business Process category; and (2) *Aircraft engine physical model* for Business Process Role category. For further reference, this business context expression is denoted as BC_A.

C. CCL management

In this section, the main results of the CCL management module are presented. First, the DT service specification (see Fig. 2) is analyzed. Second, the information obtained from the analyzed DT service specification is used to construct the list of CCs and BIEs that capture that service specification. For contextualization of BIEs, the previously created business context knowledge base is employed (see Fig. 3). The assumption is that the CCL is empty at the beginning (which does not have to be the case in reality). The CCL structure is presented in Fig. 4.

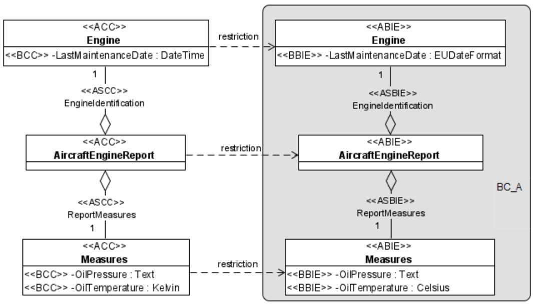

Figure 4. CCL – CC and BIE components

The list of CCs is presented on the left-hand side, while BIEs are presented on the right-hand side. CCs are created by abstracting all specificities identified in the DT service specification (e.g., data patterns). For demonstration purposes, for the *OilTemperature* BCC, Kelvin is chosen as data type, which is the conventional temperature unit according to the International System of Units (SI system) [15]. Since only one DT service specification was analyzed, all BIEs have the same business context expression which was previously denoted as BC_A (see Section IV.B).

D. Mapping management

This section presents the main result of the Mapping management module. As emphasized earlier, BIE → CC mapping specification is created every time when a new BIE is identified. In Table I two examples of mapping specifications are presented. The first specification demonstrates the needed activity to map the data from EUDateFormat to DateTime format. The second specification demonstrates needed activity to map the data from Celsius to Kelvin temperature unit.

TABLE I.
MAPPING SPECIFICATION

BIE name	BIE data type	CC name	CC data type	Mapping activity
Last maintenance date	EUDateFormat	Last maintenance date	DateTime	Date format mapping
Oil temperature	Celsius	Oil temperature	Kelvin	C to K mapping

V. DISCUSSION AND FUTURE WORK

The initial validation of the proposed approach shows promising results, but certain aspects are simplified for demonstration purposes. First, in this paper, a DT service specification is expressed as an XML Schema. However, there are different ways to express the specification that should be supported. This challenge would require implementation of some form of an adapter pattern and a representation of the DT service specification in a uniform way. For that purpose, Open API specification will be considered.

Second, the paper discusses a design-time mapping specification only. However, there is a need to define mapping specification at run-time when data exchange occurs. This would require a mapping between different BIEs that represent specifications of two DT services that exchange messages. To achieve this, BIE → CC mapping specification can be employed which was previously created at design-time. Since there is a possibility that these BIEs do not restrict the same CC, the CC → CC mapping should also be addressed.

Third, for demonstration purposes, only two business context categories are employed. In general case, all aspects that can affect the message structure should be analyzed and potentially included as categories. This task can be demanding and requires detailed analysis of a business process. Identified challenges and further validation will be addressed in future work.

VI. CONCLUSION

The paper proposes a business context-based approach for DT services integration. A novel conceptual architecture is presented as well as an initial validation in the aerospace industry domain. The results are promising and show that the proposed approach can be used for standardization of DT services. The paper also identifies needed improvements that will be targeted through future work.

DISCLAIMER

Any mention of commercial products is for information only; it does not imply recommendation or endorsement by NIST.

REFERENCES

[1] M. Grieves, Digital Twin : Manufacturing Excellence through Virtual Factory Replication - A Whitepaper by Dr . Michael Grieves, White Pap. (2014) 1–7.

[2] F. Tao, M. Zhang, Y. Liu, A.Y.C. Nee, Digital twin driven prognostics and health management for complex equipment, CIRP Ann. 67 (2018) 169–172. https://doi.org/10.1016/j.cirp.2018.04.055.

[3] F. Tao, M. Zhang, A.Y.C. Nee, Digital Twin Driven Smart Manufacturing, 1st editio, Academic Press, 2019.

[4] Core Components Technical Specification CCTS, version 3.0., (2009) 162. https://unece.org/DAM/cefact/codesfortrade/CCTS/CCTS-Version3.pdf (accessed March 10, 2021).

[5] D. Novakovic, C. Huemer, Business Context Sensitive Business Documents : Business Context Aware Core Components Modeling Using the E-UCM Model, in: 11th IEEE Int. Conf. Ind. Informatics, IEEE, Bochum, Germany, 2013: pp. 523–528.

[6] F. Tao, Q. Qi, New IT driven service-oriented smart manufacturing: Framework and characteristics, IEEE Trans. Syst. Man, Cybern. Syst. 49 (2019) 81–91. https://doi.org/10.1109/TSMC.2017.2723764.

[7] F. Tao, Y. Cheng, L. Zhang, A.Y.C. Nee, Advanced manufacturing systems: socialization characteristics and trends, J. Intell. Manuf. 28 (2017) 1079–1094. https://doi.org/10.1007/s10845-015-1042-8.

[8] S. Phumbua, B. Tjahjono, Towards product-service systems modelling: A quest for dynamic behaviour and model parameters, Int. J. Prod. Res. 50 (2012) 425–442. https://doi.org/10.1080/00207543.2010.539279.

[9] F. Tao, Q. Qi, A. Liu, A. Kusiak, Data-driven smart manufacturing, J. Manuf. Syst. 48 (2018) 157–169. https://doi.org/10.1016/j.jmsy.2018.01.006.

[10] Q. Qi, F. Tao, Y. Zuo, D. Zhao, Digital Twin Service towards Smart Manufacturing, Procedia CIRP. 72 (2018) 237–242. https://doi.org/10.1016/j.procir.2018.03.103.

[11] International Standard Industrial Classification of All Economic Activities (ISIC), (2008). https://unstats.un.org/unsd/classifications/Econ/Download/In

Text/ISIC_Rev_4_publication_English.pdf (accessed June 19, 2021).

[12] D. Novakovic, Business Context Aware Core Components Modeling Vienna PhD School of Informatics, Vienna University of Technology, 2014. https://publik.tuwien.ac.at/files/PubDat_234005.pdf.

[13] Unified Context Methodology Technical Specification, (2010). https://studylib.net/doc/7241720/un-cefact-unified-context-methodology-technical (accessed May 15, 2021).

[14] D. Novakovic, C. Huemer, Business Context Sensitive Business Documents: An Ontology Based Business Context Model for Core Components, Proc. Tenth Int. Conf. Informatics Inf. Technol. (2013).

[15] International System of Units, (n.d.). https://www.britannica.com/science/International-System-of-Units (accessed February 10, 2022).