



HAL
open science

Systems Engineering Approach to Identify Requirements for Digital Twins Development

Ali Gharaei, Jinzhi Lu, Oliver Stoll, Xiaochen Zheng, Shaun West, Dimitris Kiritsis

► **To cite this version:**

Ali Gharaei, Jinzhi Lu, Oliver Stoll, Xiaochen Zheng, Shaun West, et al.. Systems Engineering Approach to Identify Requirements for Digital Twins Development. IFIP International Conference on Advances in Production Management Systems (APMS), Aug 2020, Novi Sad, Serbia. pp.82-90, 10.1007/978-3-030-57993-7_10 . hal-03630920

HAL Id: hal-03630920

<https://inria.hal.science/hal-03630920>

Submitted on 5 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Systems Engineering Approach to Identify Requirements for Digital Twins Development

Ali Gharaei¹ [0000-0002-9644-2318], Jinzhi Lu¹ * [0000-0001-5044-2921], Oliver Stoll² [0000-0002-1918-3842], Xiaochen Zheng¹ [0000-0003-1506-3314], Shaun West² † [0000-0002-8042-1459] and Dimitris Kiritsis¹ [0000-0003-3660-9187]

¹ EPFL, SCI STI DK ME A1 Station 9, CH-1015 Lausanne

² HSLU, Technikumstrasse 21, CH-6048 Horw

* jinzhi.lu@epfl.ch

† shaun.west@hslu.ch

Abstract. Digital Twins (DT) are proposed in industries to support the entire lifecycle of services with different perspectives. Lack of systematic analysis of DT concepts leads to various definitions and services which challenges the DT developers for data integration and integrated service delivery. In this paper, a systems engineering approach is proposed to identify the requirements of DT in order to formalize the DT concepts from a systematic perspective. The conceptual architecture of DT is defined based on ISO standard 42010. Several concepts are captured to recognize DT, to define related terminologies, and to identify concerns and viewpoints in order to provide cues for delivering DT services to industry. This approach is evaluated by multiple industrial use-cases under the Innosuisse IMPULSE project, from which one-use case is selected for further elaboration. It contributes to the development of DT associated to the use-case by addressing the requirements of DT using a semi-formal approach.

Keywords: Digital Twin, Systems Engineering, Architecture Design.

1 Introduction

With the rapid advancement of Industry 4.0, Digital Twins (DT) are increasingly gaining attention for their potential to support intelligent decision-making for real-time cyber-physical systems. As an emergent technique, many industries from different domains are developing different DT concepts aimed at establishing cost-effective solutions for product development and maintenance. However, the DT concepts are separately defined by different domains, which leads to challenges defining a unified DT concept. Without unified definitions, stakeholders of DT find it difficult to communicate with the same language and thought structure. This negatively impacts the proper delivery of DT services and decision-making identifications.

The DT is a digital duplication of entities, with real-time two-way communication enabled between the cyber and physical spaces [1]. As a DT is developed, four main perspectives need to be considered: i. DT domains; ii., DT organizations; iii., DT models; and, iv., DT decision-making. DT domains refers to the disciplines the DT is used for. DT organizations refer to the group and enterprises providing DT concepts. DT

models refer to models constructing virtual entities of DT. DT decision-making to the purposes. These perspectives are difficult for DT stakeholders to identify because the systematic view of the DT requires more domain-specific knowledge. Thus, a general architecture description of DT is useful for stakeholders for DT development, implementation and maintenance.

The contribution of this paper is to illustrate a systems engineering approach to formalize architecture descriptions of DT for different domains. Based on ISO standard 42010 as a reference model, DT is defined as a “system” whose architectural entities are formalized including stakeholder concerns and viewpoints. Through these entities, requirements are identified which are used for making decisions about DT service delivery. Finally, an ontology model is developed to visualize the architectural entities from the selected use-case in an Innosuisse project IMPULSE, which aims to provide a unified architecture description of DT.

The rest of the paper is organized as follows. We discuss the related work in Section 2 and introduce the systems engineering approach to identify the architecture description of DT in Section 3. In Section 4, the use case is elaborated to evaluate the approach to identify each requirement of DT development. Finally, we discuss findings and offer our conclusions with a summary in Section 5.

Based on the output of this paper an ontology-based approach will be used as a continuation of this works to formalize the architecture entities of DT and visualize a DT concept. This, however, remains beyond the scope of this paper.

2 Literature review

The concept of digital twins is an emergent technique aiming at providing a rapid and effective solution for product development [2]. The DT concept consists of physical products in real space, virtual products in virtual space and communications between these two spaces [3]. Different domain specialists provide different definitions and technologies to construct digital twins [4]. This brings a new challenge to define and formalize the architecture of DT so stakeholders can make appreciate decisions when developing DT [1].

In systems engineering aspects, an international standard ISO/IEC/IEEE 42010 - Systems and software engineering – architecture description – is officially recommended for describing the system architectures as a reference practice [5]. As R. Capilla et al., [6] explained, the ISO 42010 has provided meta-models for describing systems and software architectures. The specification concluded that most of the key concepts and meta-models in different architectural knowledge management (AKM) tools are consistent. E. Kavakli et al., [7] use the proposed architecture model to produce specifications for a smart system for disruption management. They proposed informational, physical and logical viewpoints to frame the stakeholders’ concerns. P. Obergfell et al., [8] used the reference architecture description to specify functional capabilities of modern automobiles from various aspects including services, software, electronics and electrics. They have applied their so-called Electric/Electronic-architecture (E/E-architecture) description method for identifying the Adaptive Cruise Control (ACC) features.

From the literature review, we found DT has been widely proposed by different domains with specific definitions and concepts [1, 9]. A general architecture concept for

DT is needed to identify domain-specific requirements for DT development. We found ISO 42010 has been widely used to formalize the architectures of complex systems and can potentially support DT architecture descriptions from different perspectives.

3 Systems engineering approach to identify the requirements of a digital twin

In this paper, a systems engineering approach is proposed to formalize the requirements used for making decisions on DT development. DT developers make use of ISO 42010 to identify the architecture of DT and propose questions to represent views of DT. Then based on an Easy Approach to Requirements Syntax (EARS) [10], requirements are formalized referring to answers to the proposed questions. Fig. 1 shows how the requirements are captured in order to make DT development decisions.

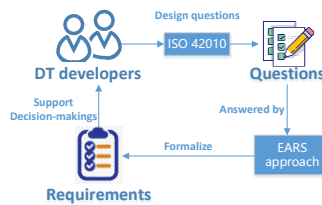


Fig. 1. Workflow of the proposed systems engineering approach

3.1 Identify questions using ISO 42010

From literature reviews, we used the international standard ISO/IEC/IEEE 42010 as the reference model, aiming at defining the key architectural concepts and system boundaries for DT from domain-specific perspectives, particularly for the Swiss Innovation Project IMPULSE.

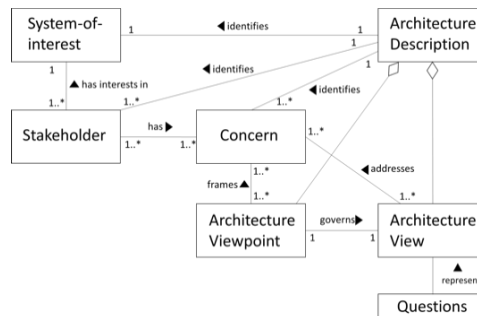


Fig. 2. Reference Architecture in ISO 42010 [11]

To focus on the DT requirements, we simplified the model to investigate the core conceptual elements of the DT architecture; System-of-interest, Stakeholder, Concern, Architecture Description, Architecture Viewpoint, and View – as shown in Fig. 2.

The System-of-interest refers to the collection of physical and virtual entities in a DT. Stakeholders interested in the DT (System-of-interest) each have their concerns, for example, DT system developers want to obtain the structure of the physical entities. The architecture description identifies the system-of-interest, the stakeholders and their concerns. It is comprised of more than one architecture view addressing a set of stakeholders' concerns and more than one architecture viewpoint. An architecture viewpoint is a work product that provides a framework for a set of concerns by presenting conventions for constructing, interpreting and using architecture views. An architecture view is a product addressing a (set of) concern(s) raised by the identified stakeholders.

In this paper, the system-of-interest is defined as the collection of physical and virtual entities of the DT which is being developed. The boundary is identified by considering the main stages of the DT system's lifecycle: concept, development, implementation, utilization, support, and retirement [12]. Stakeholders are identified within that boundary while envisioning the phases of the DT's lifecycle. The term 'actor' (with defined roles) is used within Service Dominant logic [13]) and 'beneficiary' identifies where value accrues; these terms are complementary to the Systems Engineering term of stakeholder: stakeholders are actors directly involved with, or one-step-removed from the focus. Stakeholders are identified by the classification of different actors (roles). They provide concerns in order to identify the viewpoints and views. The onion diagram of product stakeholders [14] clearly details the taxonomy of stakeholders and their relationships. The concerns of the Digital Twin's stakeholders can be categorized based on various viewpoints such as performance, asset safety, etc. These viewpoints and views construct an architecture description of DT. Finally, questions for DT development are followed by answers (given by the use case owner) in a semi-formal format and proposed as views to DT developers to identify requirements for DT development.

3.2 Define Requirements using an EARS approach

The EARS approach is proposed to formalize the requirement syntax using a semi-formal approach. The requirement is defined as a statement as following:

1. What a system must do (Functional Requirements).
2. A known limitation or constraint on resources or design (non-Functional Req.).
3. How well the system must do what it does (non-Functional Requirements).

The functional requirement is defined as a general syntax for functional requirements:

$$\text{ReqF}=\{[\text{Trigger}], [\text{Precondition}], \text{System}, \text{System behavior}, [\text{Object}]\} \quad (1)$$

Where square brackets (e.g., []) refers to the elements which can be removed. Trigger refers to when a System or others implement actions. Precondition refers to the conditions when a System or others implement actions. System refers to the target system implementing some actions (System behavior) to Objects.

In the EARS approach, requirements are defined as:

1. Ubiquitous refers to the system behavior always occurring.
2. Event-driven refers to the requirements for certain system behaviors.

3. Unwanted behaviors refer to the system behaviors under unwanted conditions.
4. State-driven refers to the system behaviors while in its own states.
5. Optional features refer to the system behaviors under specific conditions.
6. Complex refer to requirements constructed by collections of the above.

Table 1. EARS approach for formalizing requirements

Requirement type	Format rules
Ubiquitous	The <system or component> shall <system behavior> ([Object]).
Event-driven	WHEN <trigger> <optional precondition> the <system> shall <system behavior> ([Object]).
Unwanted behaviors	IF <unwanted condition or event>, THEN the < system or component > shall < system behavior> ([Object]).
State-driven	WHILE <system statement>, the < system or component > shall < system behavior> ([Object]).
Optional features	WHERE <feature is included>, the < system or component > shall < system behavior> ([Object]).
Complex	Combination of the above requirements using first order logic[1].

4 Use case

In order to evaluate the proposed approach, an industrial use case is investigated. First, stakeholders across the lifecycle of Digital Twin (from Developer, Implementer, User, and Maintainer/Supporter) are identified. By interviewing them, their concerns are collected. Then, four main architecture viewpoints are proposed to frame those concerns.

4.1 Use Case Overview

Company A seeks to add value to its business by offering new asset condition monitoring with control capabilities to its customer (Company C). Company C owns critical rooms and assets including server equipment where their temperature has to be monitored and controlled with special care. Company A has two partner companies: Company B and Company D; the former supports Company A in developing, implementing, and maintaining the DT and the latter in the support phase of the DT lifecycle.

The DT is designed for constant monitoring of temperature profiles in parts of the room as well as the status of specific artifacts such as servers, door/window (being open/closed), HVAC systems, etc. The DT will also be coupled with simulation capabilities to run what-if scenarios and offer potential solutions including control command suggestions to secure proper functioning, and optimal post-failure reaction processes.

4.2 Systems Engineering Analysis

Using the systems engineering approach, we interviewed the stakeholders related to the use case. Fig. 3 illustrates the dependencies from stakeholders (blocks in blue) to viewpoints (blocks in green). Blocks in yellow represent categories of stakeholders, and the term ‘stakeholder’ refers to all the main people/teams involved across the DT lifecycle.

Each stakeholder has their own concern(s). The developed viewpoints frame multiple concerns which are raised by one or more stakeholders. By considering all stages of the Digital Twin from concept to retirement, main stakeholders in the use case are identified, followed by the various concerns they have – as shown in Fig. 3.

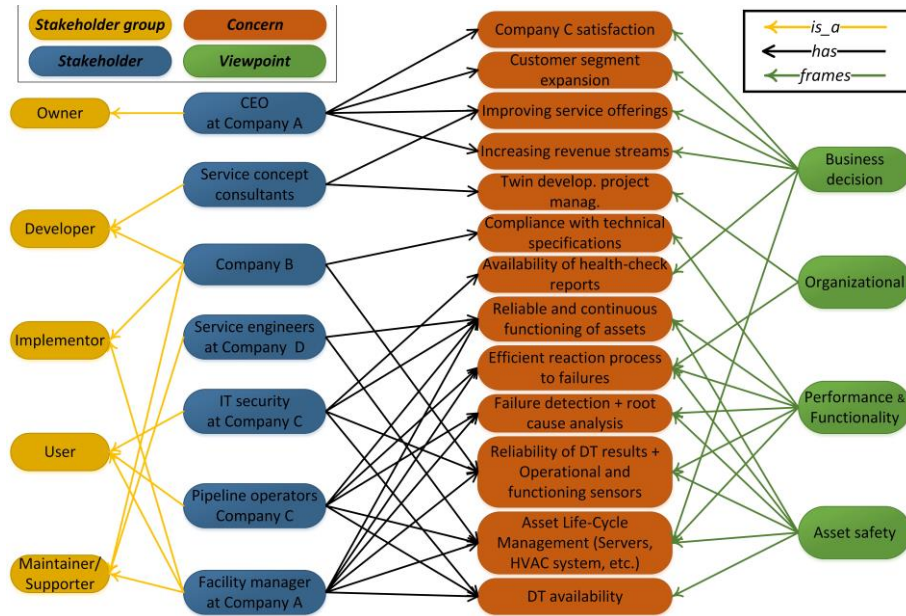


Fig. 3. Dependencies between actors, stakeholders, concerns and viewpoints

Based on the concerns in each viewpoint, questions are proposed to identify views, then the answers are formalized using the ERAS approach. Proposed examples of questions and answers are shown in Table 2. They are used to capture the requirements of the DT development in the use case.

4.3 Limitations

In the use case, only one service scenario for one customer (Company C) is explored. The stakeholders are identified from actors related to the DT. More stakeholders can be defined in order to capture more complete requirements. However, based on the scope of the IMPULSE project, several key stakeholders were selected to evaluate the approach given in this paper. The whole analysis process demonstrates the systems engineering analysis for the DT required by the use case. More detailed requirements will be analyzed in the future.

This paper is limited to the requirements identifications as one of the earliest efforts prior to initiating the DT development phase. Nevertheless, based on the output of this paper, the work is continued further on to identify all the DT's entities in each use case, classify those entities in relation with each other using ontology languages and finally propose a reference architecture for DTs.

Table 2. Questions used for formalizing views

Viewpoints	Examples of questions for representing views
Business decision	<p>1. Would the DT raise customer satisfaction? <i>WHEN <the DT is operational> the <DT> shall < improve system reliability through failure detection and improving reactions> so that < operations of company C can carry out their core businesses to a high degree of satisfaction>.</i></p> <p>2. Would the DT help Company A reach new customers? <i>WHEN <DT solution service is offered by the owner> the <DT> shall <provide further benefits to a previously unidentified group of customers, in a way that they are willing to benefit from Company A in exchange>.</i></p>
Organizational	<p>1. How can twin project development and management lead to the execution of a successful DT solution? <i>When <service concept consultants and project managers> guide <the project work, to focus on obtaining the highest value from their activities> the DT solution <shall meet and fulfil the needs of all actors in the ecosystem so that they can execute their own business goals successfully>.</i></p> <p>2. In case of a failure and detection by the DT, how should responsibilities be distributed among various actors? <i>When <the DT is operational> various failure alerts shall guide <facility managers through the reaction process> so that <they can deliver their service level agreement to the pipeline operators>.</i></p>
Performance & functionality	<p>1. Would the DT be able to meet the predefined specifications? <i>The <sensors> shall <remain continuously operational and functional>.</i> <i>The <DT> shall <produce automatic periodic reports>.</i> <i>The <DT results> shall <continuously be displaying monitoring data>.</i> <i>The <DT> shall <alert when data exceeds boundaries and assist with that specific reaction process>.</i></p> <p>2. How can we be sure the critical assets work continuously and reliably? <i>The <Digital Twin> shall <constantly provide asset operational data to Company A, Company C and facility managers; additionally, routine physical inspections are carried out.></i></p>
Safety	<p>1. How can this DT system improve asset safety? <i>WHEN <the DT is operational> it shall <improve multiple process of MEBAG, various departments of Company C and Company D> so that <all equipment operates as should that supports the other critical reliant equipment in this system to improve overall asset safety>.</i></p>

5 Conclusion

In this paper, we proposed a Systems Engineering approach to identify requirements, in order to support decision-makings of DT development. The systems engineering approach uses ISO 42010 to identify the architectures of DT and provide questions for capturing the requirements. Based on such questions, requirements are formalized using an EARS approach. Finally, a use case from the IMPULSE project is used to evaluate the approach. From the results, we find the requirements are identified to support decision-making for developing the DT with features to monitor and control the temperature in critical areas.

Acknowledgement

The work presented in this paper is supported by the EU H2020 project (869951) FACTLOG-Energy-aware Factory Analytics for Process Industries and EU H2020 project (825030) QU4LITY Digital Reality in Zero Defect Manufacturing and the Inno-suisse (Project 35258.1 IP-SBM) IMPULSE project on Digital Twins. The authors also would like to thank the Lucerne University of Applied Sciences and Arts and the Zurich University of Applied Sciences for supporting this work.

References

1. J. Lu, X. Zheng, A. Gharaei, K. Kalaboukas, and D. Kiritsis, "Cognitive Twins for Supporting Decision-Makings of Internet of Things Systems," in *Proceedings of 5th International Conference on the Industry 4.0 Model for Advanced Manufactory*, 2019, pp. 1 – 11.
2. M. Bricogne, J. Le Duigou, and B. Eynard, "Design Processes of Mechatronic Systems," in *Mechatronic Futures*, Cham: Springer International Publishing, 2016, pp. 75–89.
3. F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Trans. Ind. Informatics*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.
- K. Y. H. Lim, P. Zheng, and C. Chen, "A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives," *J. Intell. Manuf.*, no. November, Nov. 2019.
4. IEEE, "INTERNATIONAL STANDARD ISO / IEC / IEEE Systems and software engineering — Architecture description," 2011.
5. R. Capilla, A. Jansen, A. Tang, P. Avgeriou, and M. A. Babar, "10 years of software architecture knowledge management: Practice and future," *J. Syst. Softw.*, vol. 116, pp. 191–205, 2016.
6. E. Kavakli, J. Buenabad-Chavez, V. Tountopoulos, P. Loucopoulos, and R. Sakellariou, "WiP: An architecture for disruption management in smart manufacturing," *Proc. - 2018 IEEE Int. Conf. Smart Comput. SMARTCOMP 2018*, pp. 279–281, 2018.
7. P. Obergfell, F. Oszwald, M. Traub, and E. Sax, "Viewpoint-Based Methodology for Adaption of Automotive E/E-Architectures," *Proc. - 2018 IEEE 15th Int. Conf. Softw. Archit. Companion, ICSA-C 2018*, pp. 128–135, 2018.
8. J. Terzakis, "EARS : The Easy Approach to Requirements Syntax," 2013.
9. Tao, F., Zhang, M., & Nee, A. Y. C. (2019). *Digital twin driven smart manufacturing*. London, United Kingdom: Academic Press.
10. ISO/IEC/IEEE© Std. 4201:2011, "INTERNATIONAL STANDARD ISO / IEC / IEEE Systems and software engineering — agile environment," ISO/IEC/IEEE 26515
11. First Ed. 2011-12-01; Corrected version 2012-03-15, vol. 2012, 2011.
12. R. D. H. T. M. S. DAVID D. WALDEN, GARRY J. ROEDLER, KEVIN J. FORSBERG, For INCOSE member, Corporate Advisory Board, and Academic Council use only. Do not distribute. 2015.
13. P. H. Schmitt, "First-Order Logic," in *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, pp. 23–47.
14. Alexander, I. F. (2011). A Taxonomy of Stakeholders. *International Journal of Technology and Human Interaction*. <https://doi.org/10.4018/jthi.2005010102>
15. Vargo, S. L., & Lusch, R. F. (2016). Institutions and axioms: an extension and update of service-dominant logic. *Journal of the Academy of Marketing Science*. <https://doi.org/10.1007/s11747-015-0456-3>