On the Engineering of IoT-Intensive Digital Twin Software Systems

Luis F. Rivera
rivera@uvic.ca
University of Victoria
Victoria, British Columbia, Canada

Hausi A. Müller
hausi@uvic.ca
University of Victoria
Victoria, British Columbia, Canada

Norha M. Villegas
nvillega@icesi.edu.co
Universidad Icesi
Cali, Valle del Cauca, Colombia

Gabriel Tamura
gtamura@icesi.edu.co
Universidad Icesi
Cali, Valle del Cauca, Colombia

Miguel Jiménez
miguel@uvic.ca
University of Victoria
Victoria, British Columbia, Canada

ABSTRACT

Digital Twins (DT) are software systems representing different aspects of a physical or conceptual counterpart—the real twin, which is instrumented with several sensors or computing devices that generate, consume and transfer data to its DT with different purposes. In other words, DT systems are, to a large extent, IoT-intensive systems. Indeed, by exploiting and managing IoT data, artificial intelligence, and big data and simulation capabilities, DTs have emerged as a promising approach to manage the virtual manifestation of real-world entities throughout their entire lifecycle. Their proliferation will contribute to realizing the long-craved convergence of virtual and physical spaces to augment things and human capabilities. In this context, despite the proposal of noteworthy contributions, we argue that DTs have not been sufficiently investigated from a software engineering perspective. To address this, in this paper we propose GEMINIS, an architectural reference model that adopts self-adaptation, control, and model-driven engineering techniques to specify the structural and behavioural aspects of DTs and enable the evolution of their internal models. Moreover, we introduce an approach for engineering IoT-intensive Digital Twin Software Systems (DTSS) using GEMINIS' capabilities to deal with uncertain conditions that are inherent to the nature of mirrored physical environments and that might compromise the fidelity of a DT. With GEMINIS and the proposed approach, we aim to advance the engineering of DTSS as well as IoT and cyber-physical systems by providing practitioners with guidelines to model and specify inherent structural and behavioural characteristics of DTs, addressing common design concerns.

CCS CONCEPTS

• Software and its engineering → Software design engineering; **• Computer systems organization → Real-time system architecture**; **Self-organizing autonomic computing**.

ICSEW'20, May 23–29, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7963-2/20/05…$15.00
https://doi.org/10.1145/3387940.3392195

KEYWORDS

Digital twin, IoT, reference model, self-adaptive, models at runtime, megamodel, adaptive control, MRAC, MIAC, dynamic context management, GEMINIS, DTSS

ACM Reference Format:
Luis F. Rivera, Hausi A. Müller, Norha M. Villegas, Gabriel Tamura, and Miguel Jiménez. 2020. On the Engineering of IoT-Intensive Digital Twin Software Systems. In *IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW'20), May 23–29, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3387940.3392195

1 INTRODUCTION

Creating complete or partial digital representations of physical and conceptual entities is driving the understanding and management of things, processes, systems and humans [14], by augmenting them in multiple ways with technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), Autonomous Computing, and Big Data Analytics. This digitalization of information and its organization through meaningful structures are transforming both industries and societies [6]. On the one hand, it is allowing industries to advocate for flexible and primarily autonomous products, services, and manufacturing processes. In this context, as defined in the Industry 4.0 initiative [13], the exchange of longitudinal data across systems and organizations facilitates improved efficiency, productivity, and adaptability. On the other hand, as proposed in the concept of Society 5.0 [16], the confluence of digital technologies will extend human capabilities and provide invaluable assistance in almost every aspect of our lives including healthcare, transportation, and economy.

Undoubtedly, the convergence of virtual and physical spaces is crucial to promising technological advances such as Industry 4.0 and Society 5.0, whose realization relies upon the development and exploitation of robust IoT systems [13, 16]. Conceived in the context of product lifecycle management (PLM) [7], the concept of Digital Twin (DT) represents a step forward in the evolution of this kind of systems. A DT comprises a set of virtual representations describing the fundamental structural and behavioural characteristics of a physical entity, that is, its real twin, throughout its entire lifecycle. Each of the virtual representations comprising a DT can address a different aspect of the entity. This opens up possibilities for both real-time analysis in different dimensions and behavioural predictions that support data-driven Continuous Engineering (CEng) processes. As depicted in Figure 1 (A), DT systems are, to a large extent, IoT-intensive systems that include control mechanisms to accurately mirror physical or conceptual entities (*i.e.*, real twins). Moreover, DTs extend IoT capabilities and enrich Cyber-Physical Systems (CPS) [10] by enhancing the management and exploitation of data coming from the physical space to assist control actions and decision making in the virtual space (*cf.* Figure 1 (B)).

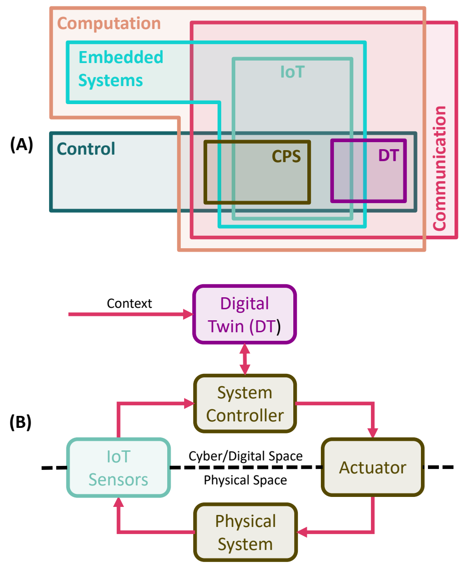

Figure 1: *(A)* Euler diagram depicting the overlapping definitions between modern systems foundations (*i.e.*, computation, communication, and control) and recent technologies (*i.e.*, Internet of Things (IoT) [26], Cyber-Physical Systems (CPS) [17, 25] and Digital Twin (DT) [7]). *(B)* Envisioned closed control loop of IoT, DT, and CPS. Adapted from [27].

Despite the increasing acceptance and proliferation of DTs [4, 5, 12, 21], further research on how to model and engineer DT systems is still required [1, 4, 14]. Particularly, we submit that although DTs' potential primarily stems from their software-based capabilities, their software engineering is still in its infancy. In most cases, DTs have only been studied from a perspective that does not elaborate on their internals. Until now, most researchers have addressed the design of DTs from a very abstract standpoint that does not provide enough details about DT's fundamental internal structures and their associated responsibilities, independently from the application domain. Usually, DTs are conceived as black boxes that provide services to larger systems or misinterpreted as merely AI or simulation models. Only a limited number of approaches consider the definition of detailed building blocks for Digital Twin Software Systems (DTSS) [10]. Nonetheless, it has not been analyzed whether DTs should inherently have particular sub-structures or characteristic behaviours that enable them to continuously mirror their real-twin counterparts precisely.

In this paper, we argue that DTs should invariably, and independently of their application domain, manage uncertainty arising from the inherent dynamism faced by real twins and the variable nature of their surrounding environments. Consequently, DTs should be instrumented with control and adaptation mechanisms that enable DT models to continuously adjust according to changes in their real counterparts, including their environment (context), therefore favouring the fidelity of DT representations.

Our contributions include an approach to engineer IoT-intensive DTSS using GEMINIS,[1] our novel reference model based on research advances in self-adaptive systems, adaptive control, models [2] and megamodels [24] at run-time (MARTs an MeMARTs, respectively), and model-driven engineering (MDE). GEMINIS addresses relevant concerns of DTSS that are crucial to the achievement of promises and expectations of the DT concept and its impact on application domains such as Industry 4.0. and Society 5.0. In particular, we identify the following concerns as key to the concept of DT to succeed and get materialized through the engineering of DTSS:

i Coping with the dynamic and uncertain characteristics of the physical world (i.e., faced by the real-twin).
ii Facilitating real-time analysis and dynamic management of observable characteristics from the entities mirrored by the DT.
iii Guaranteeing consistency between DT models and the real state and behaviour of the mirrored entity toward the achievement of high fidelity representations supporting evolution cycles such as CEng processes.
iv Preserving the satisfaction of high-level DT goals.

The remainder of this paper is structured as follows. Section 2 describes our proposed GEMINIS reference model for DTSS. Section 3 introduces our GEMINIS-based approach to engineer DTs by outlining its application on a relevant and real case study of Smart Urban Transit Systems. Section 4 discusses related and preliminary work, and challenges regarding the development of IoT-intensive DTSS. Finally, Section 5 concludes the paper and proposes future work.

2 THE GEMINIS REFERENCE MODEL

This section introduces GEMINIS, the main component of our proposal for engineering DTSS. GEMINIS is a reference model for architecting DTSS that enforces functional decomposition, separation of concerns (*i.e.*, management of DT context sources, DT model management and evolution, and preservation of DT high-level objectives) and the establishment and maintenance of association relations between twins. To describe GEMINIS, we first discuss DT characteristics by considering modelling aspects and application dimensions that influence the development of DTs that had a significant impact on the design of our proposed reference model. Then, we detail GEMINIS by specifying its internal structures and their core responsibilities.

2.1 Characteristics of Digital Twins

DTs are complex entities whose design, development and maintenance demand the integration of diverse disciplines to ensure the construction of precise digital representations of conceptual or physical entities and their associated phenomena. Initially, although modelling DTs might resemble creating object-oriented models, DTs present distinctive characteristics that require extending or combining existing modelling approaches. Both DT researchers and

[1] GEMINIS ['he-mi-nis]: **G**eneral Referenc**E** **M**odel for Dig**I**tal Twi**N**-Dr**I**ven **S**ystems

industry practitioners acknowledge that three major factors impact DT design and development (and consequently the associated models) [4]: *real-time data aggregation*, in which DT models serve as a consolidated data repository that enables uncovering hidden knowledge by exploiting operational and continuously produced data by the real twin (usually through IoT sensors); *integration and interoperability*, referring to DT models' ability to interact, integrate, and provide services to other (possibly larger) systems; and *fidelity*, which involve providing DTs with mechanisms and appropriate models to describe and correspond to a real twin precisely. Evaluating the fidelity of a DT requires understanding the level of detail or abstraction (*i.e.*, *Resolution*), coverage (*i.e.*, *Completeness*), and update regularity (*i.e.*, *Frequency*) of its constituent models [22].

Figure 2 depicts our conception of the models that comprise the definition of a high-fidelity DT. Certainly, the development stage and the type of a real twin will determine the appropriate set of models for the composition of its digital counterpart. For example, in the early stages of manufacturable real twins, DTs might be comprised of 3D CAD and structural models. Moreover, the selection of these models can also be affected by the different stages of development of a DT, including [1]: *offline monitoring and reactive control*, where the automation of gathering and understanding of DT data might be minimum or absent; *online control*, in which automatic DT-driven control actions are based on forthright data retrieval and processing; *predictive control*, which comprises smart adaptations for DT fine-tuning; and *proactive control*, involving self-optimization and adaptions for the achievement of goals.

Understanding the development stages of real and (intended) digital twins and defining structural and behavioural models accordingly are crucial to the success of IoT-intensive DTSS. On the one hand, structural models describe the static structure of an entity, that is, the parts or elements that compose it or abstract relevant properties, at different levels of abstraction, and whose specification does not involve behaviour considerations. For advanced IoT-intensive DTSS, this kind of DT models must inherently consider (i) the relationship with the real twin, mediated by the sensors and IoT devices connected to it; (ii) the type and frequency of data being gathered (iii) data storage repositories supporting continuous and diverse flows of data; (iv) the DT's models interrelations; and (v) the structural components for supporting minimum self-management behaviour. On the other hand, behavioural models characterize the dynamic aspects of a real twin. This refers to changes in the mirrored twin over time. We advocate to address three aspects when considering runtime behavioural characteristics of a highly-developed IoT-intensive DTSS: (i) the computational state of DT models must accurately reflect, at every moment, the current or predicted state of the real twin; (ii) the interactions between the real twin, data repositories, behaviour models, and DT-related systems and operators must be clearly defined; and (iii) the results of data exploitation (*e.g.*, via machine learning, simulation, experimentation, or other forms of data exploitation) should augment the DT self-management capabilities. In addition to the structural and behavioural perspectives of a DT, we follow the clear distinction between DT prototypes (DTP) and DT instances (DTI), as proposed by Grieves[2] & Vickers [8]. Under this differentiation, knowledge acquired trough each DTI can contribute to the continuous evolution of models originally conceived in a DTP.

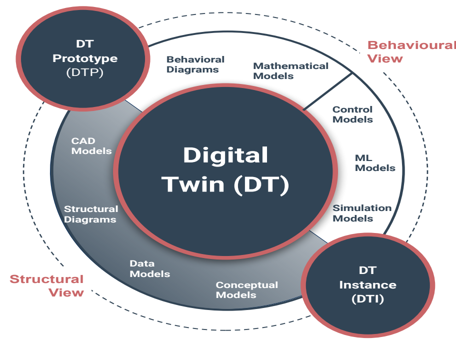

Figure 2: Structural and behavioural views in DT modelling.

The application dimension (*i.e.*, use case) of a DT will also help to delineate the type of models involved in its definition and the relationships among them. For instance, describing and predicting the behaviour of one desired entity through a DT might require the use of Machine Learning (ML) or Genetic Fuzzy Trees (GFT) models, simulating events associated with the real twin could be realized using discrete-event simulation (DES) models, and conducting experimentation through DTs might demand a combination of both ML and simulation capabilities. Assuredly, developing high-resolution and complete DTs, capable of accurately describing all the possible extents of real twins (*i.e.*, tridimensional aspects, mechanics, among others) will be the most demanding challenge since it can involve a combination of a wide variety of knowledge areas. In our research, we aim to advance the modelling of DTs from a data-driven, control, and MDE-based perspective, in which ML-based descriptive and predictive behaviour, and simulation and experimentation services can be supported, thus providing a platform for the development and extension of IoT-intensive DTSS.

2.2 The Reference Model

Figure 3 depicts the GEMINIS reference model, which addresses the modelling aspects discussed in Section 2.1 by combining principles from self-adaptive software systems, autonomic computing, adaptive control, and MDE. Moreover, it incorporates a novel application of the concepts of Model Reference Adaptive Control (MRAC) and Model Identification Adaptive Control (MIAC) for providing DTs with adaptable behaviour to deal with inevitable changes in the behavioural characteristics of their counterparts, the real twins. GEMINIS is conceptually influenced by the ACRA reference architecture [9] and the DYNAMICO [23] reference model. We adopt the hierarchical orchestration of MAPE-K [11] feedback loops contemplated in the former, and the separation of concerns and the three-layer structure of feedback loops defined in the latter. In our case, each feedback loop addresses: (i) DT control objectives; (ii) DT behavioural adaptation (with MRAC/MIAC); and (iii) DT context monitoring. In Figure 3, the MAPE-K components are represented

[2]Michael Grieves gave early definitions for DT in 2003 in the context of Product Lifecycle Management [7].

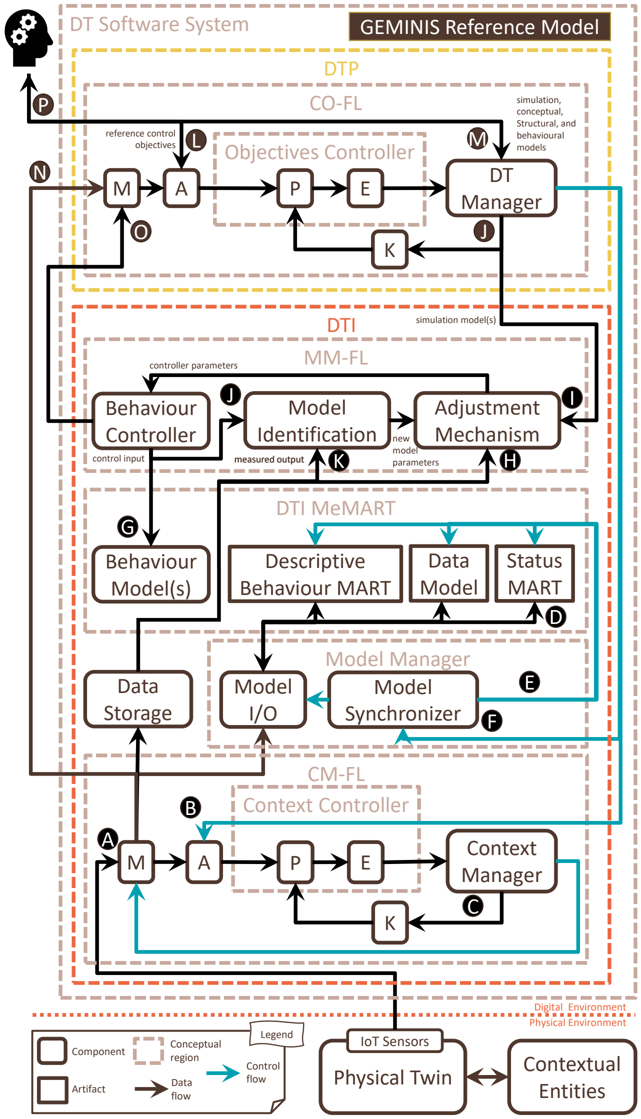

Figure 3: A view of our GEMINIS reference model that illustrates the proposed functional decomposition together with data and control flows. The acronyms used in this view stand for DTP: Digital Twin Prototype, DTI: Digital Twin Instance, MART: Model at Runtime, MeMART: Megamodel at Runtime, CO-FL: Control Objectives Feedback Loop, MM-FL: MRAC/MIAC Feedback Loop, and CM-FL: Context Monitoring Feedback Loop.

with their initials (*i.e.*, M: Monitor, A: Analyzer, P: Planner, E: Executor, and K: Knowledge). The following sections describe the different roles and responsibilities of the elements that comprise GEMINIS.

2.2.1 Context Monitoring Feedback Loop (CM-FL). The CM-FL is responsible for the acquisition of context data (batch and real-time), using sensors attached to the real twin of a particular DTI (*cf.* interaction Ⓐ). The monitoring process (M) of this CM-FL is continuously analyzed to verify its compliance with dynamic high-level control objectives, and adaptions might be triggered in case an adjustment in the monitoring scheme is needed (*cf.* interaction Ⓑ). Hence, the continuous and diverse flows of data originated in the physical world are aggregated by an adaptive monitoring mechanism that provides flexibility with respect to sampling and data fusion processes. Changes in DTSS goals might require adjustments in the sampling rates, data fusion mechanisms, and sensing sources. Furthermore, context symptoms, performed adaptations, and faulty or detected malicious sensors may form a relevant knowledge base (*cf.* interaction Ⓒ) to improve decision making in the planning of future adaptations to the monitoring scheme.

2.2.2 Model Manager & DTI MeMART. The *Model Manager* provides an interface to operate and synchronize the structural and behavioural models associated with each DTI. On the one hand, the *Status MART* describes a real-time view of a physical or conceptual entity that might be used by operators or other systems to seamlessly comprehend the latest known status of the entity (*cf.* interaction Ⓓ). This structural model establishes a causal connection between the defined set of relevant measurable characteristics of the DTI and the phenomena being monitored in the physical world. It also provides the structure for the *Data Model* that describes the schema for data storage. Thus, the *Status MART* delineates the observable properties and relationships of the real twin that are relevant for an operator or a larger system that requires services from a DTSS. This model might define additional structural characteristics by specifying the mappings between the properties and relationships being observed and the context sources (*e.g.*, IoT sensors) that provide the data for further analysis. Together with structural definition models (*e.g.*, SysML Block Definition and Internal Block diagrams), the *Status MART* and the *Data Model* comprise the elementary static view of a DT in GEMINIS. On the other hand, our reference model addresses behavioural descriptions of DTs through the definition of *Descriptive Behaviour MARTs* (*e.g.*, SysML Activity, State, and Parametric diagrams), which outline the dynamics of a physical twin for their analysis; and *Behaviour Models* (*e.g.*, ML and GFT models), expressing behaviour through executable models that can be used for prediction and forecasting. Since both structural and behavioural models might contain elements with corresponding semantics, a *Model Synchronizer* guarantees model coherence (*cf.* interaction Ⓔ) and propagates changes when adaptation is required (*cf.* interaction Ⓕ).

In our approach, DTI models are grouped into a MeMART (*i.e.*, runtime megamodel) that establishes relations (*e.g.*, dependencies and transformations) between them and improves their management by providing a centralized and explicit model registry for further real-time operations on models (*e.g.*, analysis, navigation, and change replication) [24]. This macro-model provides a structured view of a DTI for a wide variety of stakeholders (*e.g.*, clients, operators, other related systems). Moreover, it enables comprehending the impact of model changes and defines the route to propagate them.

2.2.3 MRAC/MIAC Feedback Loop (MM-FL). The MM-FL addresses the achievement of high fidelity representations of behavioural aspects in the DT concept. For this, our reference model incorporates a novel mechanism, based on the MRAC and MIAC models from adaptive control theory, to deal with the potential lack of real-time data at the early stages of CEng processes and the continuous adjustment (evolution) of behavioural models to assess and improve—if required—the way they mirror phenomena associated with the operation of physical or conceptual twins.

Newly created DTIs will include one or more conceived behavioural models intended to describe and forecast the dynamics of the associated physical or conceptual entity. To ensure the calibration of this kind of models, GEMINIS defines a *Behaviour Controller* that adjusts model parameters appropriately (*cf.* interaction 🅗). In our reference model, at the early stages of operation, an MRAC-based procedure is used to define the way behavioural models are calibrated by the corresponding controller. To achieve this, an *Adjustment Mechanism* produces new control parameters by comparing real behaviour (*cf.* interaction 🅗) with expected behaviour coming from regular flows of simulation reference models (*cf.* interaction 🅘).

Once DTIs are fully operational and a representative amount of data has been collected and processed, the *Adjustment Mechanism* might opt for adopting an MIAC-based approach for model tuning. Under this paradigm, control inputs (*cf.* interaction 🅘) and extended observed behaviour (*cf.* interaction 🅚) serve as the basis for *Model Identification*. Thus, simulation data is replaced by data coming from the constant operation of the real twin. This approach is particularly suitable for evolution scenarios with high uncertainty, where anticipation (*e.g.*, through simulation) becomes impractical [15].

The MM-FL plays a vital role in DT-driven CEng processes, such as those proposed in Industry 4.0 and Society 5.0. On the one hand, required recurrent adjustments to the models might expose the deviation of a DTI from its DTP due to unexpected degradation. Thus, predictive or corrective maintenance actions should take place. On the other hand, the data-based identification of behavioural models might reveal improvement opportunities for the products, services or processes being mirrored by DTs. Applying this to the context of Industry 4.0, companies might trigger a new evolution cycle for developing products featuring the enhancements unveiled by DTs. In this case, a new DTP is created and corresponding DTIs generated from it are instrumented with data and models coming from the previous cycle (*cf.* interaction 🅚), thus closing the DT-driven evolution loop.

2.2.4 Control Objectives Feedback Loop (CO-FL). The CO-FL supervises the satisfaction of high-level goals defined for DTSS by operators or other systems (*cf.* interaction 🅛). These goals might include the satisfaction of functional and non-functional requirements such as required fidelity, bandwidth usage, among others. The CO-FL analyzes critical contextual conditions (*cf.* interaction 🅝) and significant deviations of behavioural models (*cf.* interaction 🅞) and reports them to relevant external entities (*cf.* interaction 🅟). The *DT Manager* defined in this feedback loop is in charge of governing the resources associated with a DTP (*e.g.*, structural definition models and simulation models, *cf.* interaction 🅜), enabling the creation of new DTIs from it, and triggering the synchronization of models when required.

3 ENGINEERING DIGITAL TWIN SOFTWARE SYSTEMS (DTSS)

This section presents the second component of our proposal, a preliminary version of a process for engineering DTSS using GEMINIS. To ease the comprehension of both GEMINIS and our proposed approach, we illustrate their application on a relevant scenario related to smart urban transit systems. In this scenario, we address SITM-MIO, the mass transportation system of Cali, the third-largest city of Colombia, with a population of nearly 2.2 million. At the end of 2019, the SITM-MIO system had 98 lines (routes) and 455,000 passengers on a business day, using a fleet of 828 vehicles.

The SITM-MIO system can be considered as an IoT-intensive CPS in development, as wide as the city of Cali. SITM-MIO buses are equipped with multiple embedded systems such as a small central processing unit, a GPRS-based communications device, and more than 30 IoT sensors reporting geolocation coordinates, velocity, acceleration, among others. These buses generate around 2.5 million data records per day. Currently, we are investigating the application of IoT-intensive DTs on the SITM-MIO scenario to improve decision making and ensure the system's autonomy.

Figure 4 depicts an overview of our proposed approach and possible models involved when designing DTSS using GEMINIS. The following sections outline the main phases of the approach, adopting the SITM-MIO system as the application scenario and elaborating on the definition of DT structure and behaviour through GEMINIS. The order of the phases being described might vary according to specific conditions of different scenarios.

3.1 Selecting a Real Twin

In a CPS as complex as the SITM-MIO, there exist many opportunities to implement DTs that augment the capabilities of its constituent components to make it more resilient, efficient, autonomous, and self-managed. Moreover, after several evolutions, multiple DTSS might form an intelligent system of systems capable of driving or autonomously effectuating control or optimization actions over the entire SITM-MIO system by leveraging the integration and collaboration between diverse DTs. Achieving this, however, is a formidable challenge, for which we need to start by understanding first how to model individual DTs that add value to the current operation of the system.

At this stage of our research, and considering the main service provided by the SITM-MIO system, we decided that the concept of a bus route is the ideal candidate for a real twin that will be manifested as a DT in the virtual realm for augmented management and control. Each route in the system is an entity that conceptually has both structure and behaviour. Routes' performance affects not only the citizens who use it but also the mobility of the whole city from a global perspective.

3.2 Determining the DTSS Application Dimension & Control Objectives

As described in Section 2.1, determining the DT application dimension is key to delineating the set of models that is suitable for describing the structural and behavioural characteristics of a desired real twin under particular conditions. In the bus route modelling scenario, we aim to leverage DT predictive capabilities (*e.g.*,

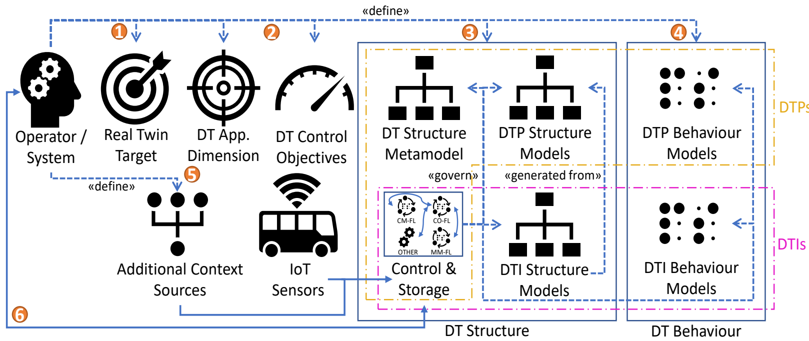

Figure 4: General view of our proposed approach for engineering DTSS using GEMINIS.

through ML models) to allow the SITM-MIO system to become more resilient in face of unexpected events during operation (*e.g.*, unusual traffic jams, blocking road accidents, emergencies, unexpected flow of passengers). In this context, the main objective of the DTSS in charge of representing the bus route is to detect or anticipate events that might have a significant or disastrous impact in the SITM-MIO operation, alert human controllers, and suggest plausible actions to counteract their consequences. Moreover, the DTSS can be instrumented with control objectives that will drive the management of associated DT models and data acquisition for computing key performance indicators (KPIs).

3.3 Modelling DTSS Structure

Defining the structure of a DTSS will require not only the use of models describing the structural characteristics of a real twin, but also the establishment of control and storage structures for management and evolution of models together with a specification of their organization and allocation to computing resources.

First, domain experts or SITM-MIO operators can use DT-ready metamodels to define structural graph-based DTP models describing the inherent and measurable properties of a bus route (*e.g.*, bus stops, planned arrival and departure times for buses, operating fleet, buses location, among others) and mappings between IoT sensors and identified properties. As proposed in our previous work [19], MDE transformations can be used to derive and update (if required) DTI structural models (*i.e.*, route MARTs and their associated data models) from the DTP definitions. These models will reflect the real-time state of bus routes in the SITM-MIO system and can be grouped into route MeMARTs that specify model interrelationships.

Second, as described in Section 2.1, besides including the models comprising the intrinsic structural characteristics of a DT, the structure of a DTSS must also consider structures to support model manipulation and self-management capabilities for improved DT control. For this, designers of the SITM-MIO management system can take advantage of the functional decomposition provided in GEMINIS and instrument the system with the feedback loops defined in the reference model to establish the way route DTI models are: (i) fed with batch or real-time data coming from route buses and stations (CM-FL), (ii) updated to continuously mirror changing operation conditions such as an unexpected flow of passengers (MM-FL), and (iii) governed by system KPIs such as the fulfillment of planned route schedules defined in DTPs (CO-FL).

Third, the structure of a DTSS should also consider the application of deployment patterns or architecture styles that specify ideal solutions for organizing DTSS components and their distribution over available computation resources. For instance, designers of the SITM-MIO scenario will have to consider optimal ways to organize and deploy DT models and related control and storage components for 96 routes, while striving to achieve minimum operational cost without comprising system performance.

3.4 Modelling DTSS Behavior

Following GEMINIS, in a similar way as in the previous section, behavioural aspects of a DTSS should include intrinsic behavioural characteristics of the mirrored real twin, described or predicted trough defined models, and adaptive behaviour exposed by control and storage structures (*i.e.*, feedback loops) for model management and evolution. Moreover, as depicted in Figure 4, we advocate for a differentiation between DTP and DTI behaviour models. We illustrate the relevance of these considerations by describing a particular scenario in the SITM-MIO system as follows.

In order to reduce passenger agglomeration in bus stations and optimize the performance of critical routes in the system, a SITM-MIO operator might exploit predictive DT behavioural models to estimate the ideal scheduling of buses (*i.e.*, the interval between each bus departure from the route origin) during particular (possibly critical) operation hours. The accuracy of these models, part of route DTIs, is inevitably susceptible to uncertain factors such as weather conditions, thus adaptions are required to enable DTIs to mirror their real twins appropriately. Initially, model adaptations can be based on performed historical control actions on the way the ideal scheduling of buses is estimated, using previous DTP models as a reference. However, as the city dynamics and boundaries evolve, reference route DTPs might become obsolete and new models and control actions should be identified. Hence the relevance of GEMINIS and the adaptability provided by the orchestration of its proposed feedback loops to preserve the fidelity of DTSS.

3.5 Defining Additional Context Sources

DT modelling should consider external elements interacting with and affecting their twin counterparts. In the SITM-MIO scenario, bus route DTs can be linked to additional context sources such as real-time traffic data services and weather forecasting to improve the level of correspondence (*i.e.*, fidelity) between DT models and their associated entities. However, we do not consider any additional context sources at this stage of our research.

3.6 Providing Interfaces to Associated Systems

When designing DTSS, engineers must consider defining interfaces supporting the continuous interaction between DTSS and associated client systems (*e.g.*, CPS) that demand real-time operation insights or prediction and forecasting capabilities offered by DT models. These interaction interfaces can augment cyber controllers' (*i.e.*, DTSS clients) capabilities for properly managing associated physical entities (*cf.* Figure 1 (B)). In the context of the SITM-MIO system, exposing interaction interfaces that provide graph-based analytics based on bus routes data accumulated through DT models can improve decision making and enable the main system controller to automatically or semi-automatically re-define a specific segment of a scheduled route and notify the adjustments to the system users. This, in turn, might cause the system to change the scheduling of buses autonomously and therefore adjust the original plan defined for a particular period.

4 RELATED WORK & PRELIMINARY WORK

This section presents an overview of previous and ongoing relevant software engineering research regarding DTs, outlines our preliminary work, and introduces challenges regarding the further development of IoT-intensive DTSS.

4.1 Related Work

In industry, IT companies have proposed high-level conceptualizations, designated as architectures, that represent organizational schemes for the implementation of DTs as key drivers of business operations. Microsoft [20] highlighted the significance of DT for digital transformation processes and provided an abstract cloud-based reference architecture that exposes the way the company's products might support the construction and operation of DTs. In a similar fashion, VANTIQ [3] describes a general architecture that features a set of components intended to address important considerations for DTs including data acquisition and processing, modelling and simulation, and corrective actions based on situational awareness and real-time analysis of physical twin states. Both architectures are presented as commercial solutions where DT elements are directly associated with vendor-specific platforms.

In academia, recent research on the engineering of DTSS has been conducted. Redelinghuys *et al.* [18] proposed an abstract six-layer DT architecture in the context of Cyber-Physical Production Systems. Their approach, however, specifically targets manufacturing cells and does not consider adaptability factors of DTs regarding their fidelity. Bauer *et al.* [1] explored the way the adoption of DTs permeates the design and development of software systems expected to exploit their capabilities. Although they do not elaborate on the functional decomposition of individual DTs and the explicit interrelationships among the elements that compose them, they suggest a set of action points to motivate further research on the architectural design of systems interacting or orchestrating DTSS. Their action points are considered within the list of concerns we described in Sections 1 and 2.1. To the best of our knowledge, the approach proposed by Josifovska *et al.* [10] is the only study, from a software engineering perspective, that has directly addressed the definition of building blocks of DTSS and their interrelationships. They introduced a reference framework that divides DTs in terms of a *Physical Entity Platform* that describes entities residing in the physical spaces; a *Virtual Entity Platform* that comprises the virtual models that mirror physical entities; a *Data Management Platform* that orchestrates data-related operations; and a *Service Platform* that provides optimization capabilities. Even though this work represents a relevant step forward toward the definition of internal structures for engineering DTSS, it does not provide software engineers with reference mechanisms to instrument DT models with adaptability and evolution capabilities to improve the way they mirror physical entities continuously—and support CEng processes—, to accommodate uncertainty. Moreover, the impact of model evolution and its mitigation are not addressed in their proposal. Therefore, we submit that the current approaches are complementary to GEMINIS with our reference model being to the best of our knowledge, the only one that explicitly provides a standard decomposition of DTSS that defines the parts, responsibilities and interactions that are required to approach the concerns that are key in the realization of DT promises by addressing of DTSS dynamics and their continuous evolution.

4.2 Preliminary Work

Previously, we presented an initial exploration of the DT concept and its application in personalized healthcare [19]. We described DT's main requirements and leveraged autonomic computing research to engineer adaptive DTs in healthcare. More specifically, we devised a reference model for engineering DTs for humans, featuring two feedback loops, namely: context monitoring and care management. The former is equivalent to the one defined in GEMINIS and the latter controls the satisfaction and optimization of high-level goals for a particular medical treatment. We aimed to augment traditional healthcare methods with mechanisms for monitoring and treatment personalization for diabetes patients.

One of our next steps is to validate GEMINIS using the concrete application scenario in Section 3. We have already developed three projects that aim at realizing the context management and data storage elements from GEMINIS. These projects focus on: (i) collection, pre-processing and storage of data sensed from the operation of buses in the system; (ii) the prediction of bus arrival times to a set of relevant stops, using collected data and ML techniques; and (iii) event identification, management and reporting regarding the buses' operation.

4.3 Challenges Ahead

By designing and applying GEMINIS to the personalized healthcare and smart urban transit scenarios, we have identified research and implementation challenges for the realization of IoT-intensive DTSS. Thus, the initial evaluation of our approach presented in this paper faces the following tasks and challenges:

- Define a DT metamodel with sufficient expressiveness to properly describe the structural characteristics (*i.e.*, observable properties) of a real twin and their relationships with context sources.

- Identify and undertake convenient actions regarding the possible drift of DT models (*i.e.*, inaccuracy of originally expected behaviour).
- Determine when a DT is sufficiently mature to consider switching the control of behavioural models from an MRAC-based approach, where simulations are used as reference inputs to adapt the behaviour of DT models, to an MIAC-based approach that identifies control parameters using data and knowledge from the operating real twin.
- Propagate required changes in definitions of structural models and address their impact on related behavioural models.
- Enable the composition of DTs into larger DT structures and guarantee the interoperability of DT models.

5 CONCLUSIONS

This paper presented an approach for engineering DTSS and the GEMINIS reference model which provides explicit functional decomposition together with data and control flows to guide the design, development and operation of robust IoT-intensive DTs that accurately mirror physical or conceptual entities. GEMINIS builds upon research contributions on self-adaptive software systems (*i.e.*, feedback loops, MARTs and MEMARTs) and adaptive control mechanisms (*i.e.*, MRAC and MIAC) for enabling DTs to (i) cope with the dynamic and uncertain characteristics of the physical world; (ii) facilitate real-time analysis and dynamic management of observable characteristics from entities mirrored by the DT; (iii) guarantee the consistency between DT behavioural models and real behaviour of the mirrored entity toward the achievement of high fidelity representations supporting evolution cycles such as CENG processes; and (iv) preserve the satisfaction of high-level DT goals. GEMINIS constitutes an advancement toward the engineering of adaptive DTSS that can cope with uncertain conditions associated with a dynamic and continuously evolving physical world. It benefits not only software engineers by serving as a guide for DTSS modeling and construction, but also software engineering researchers by paving the road for further discussions about the maturity of software engineering for DTSS and its impact on cyber-physical and IoT systems. In particular, we highlight interesting research opportunities related to the implementation of the MM-FL defined in our proposed reference model.

We are currently working on the validation of GEMINIS to consolidate it as a suitable reference model that facilitates the construction of DTSS in different application domains. Our future work will focus on addressing the challenges described in Section 4.3 in smart urban transit systems and personalized healthcare scenarios.

ACKNOWLEDGMENTS

This work was funded in part by the National Sciences and Engineering Research Council (NSERC) of Canada, The NSERC CREATE Program in Dependable Internet of Things Applications (DITA), IBM Canada Ltd., University of Victoria and Universidad Icesi (Colombia). The authors would like to thank IBM's Centre for Advanced Studies, Canada for supporting this project.

REFERENCES

[1] Thomas Bauer, Pablo Oliveira Antonino, and Thomas Kuhn. 2019. Towards Architecting Digital Twin-pervaded Systems. In *7th International Workshop on Software Engineering for Systems-of-Systems and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (SESoS-WDES '19)*. IEEE Press, 66–69.

[2] Gordon Blair, Nelly Bencomo, and Robert B. France. 2009. Models@run.time. *Computer* 42, 10 (2009), 22–27.

[3] Paul Butterworth. 2017. *White Paper: VANTIQ Digital Twin Architecture.* Technical Report. VANTIQ.

[4] Luiz Fernando C. S. Durão, Sebastian Haag, Reiner Anderl, Klaus Schützer, and Eduardo Zancul. 2018. Digital Twin Requirements in the Context of Industry 4.0. In *Product Lifecycle Management to Support Industry 4.0*, Paolo Chiabert, Abdelaziz Bouras, Frédéric Noël, and José Ríos (Eds.). Springer, 204–214.

[5] Martin Robert Enders and Nadja Hoßbach. 2019. Dimensions of Digital Twin Applications - A Literature Review. In *25th Annual Americas Conference on Information Systems (AMCIS) (AMCIS '19)*. Association for Information Systems, In press.

[6] Mayumi Fukuyama. 2018. Society 5.0: Aiming for a new human-centered society. *Japan Spotlight* 1 (2018), 47–50.

[7] Michael Grieves. 2015. *Digital Twin: Manufacturing Excellence through Virtual Factory Replication.* Technical Report.

[8] Michael Grieves and John Vickers. 2017. *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems.* Springer, 85–113.

[9] IBM Corporation. 2005. *An Architectural Blueprint for Autonomic Computing.* Technical Report. IBM Corporation.

[10] Klementina Josifovska, Enes Yigitbas, and Gregor Engels. 2019. Reference Framework for Digital Twins within Cyber-Physical Systems. In *2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*. 25–31.

[11] Jeffrey O. Kephart and David M. Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (Jan 2003), 41–50.

[12] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. 2018. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* 51, 11 (2018), 1016 – 1022. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[13] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. 2014. Industry 4.0. *Business & Information Systems Engineering* 6, 4 (01 Aug 2014), 239–242.

[14] Somayeh Malakuti and Sten Grüner. 2018. Architectural Aspects of Digital Twins in IIoT Systems. In *12th European Conference on Software Architecture: Companion Proceedings (ECSA '18)*. ACM, Article 12, 2 pages.

[15] Hausi Müller and Norha Villegas. 2014. *Runtime Evolution of Highly Dynamic Software.* Springer, 229–264.

[16] Cabinet Office. 2016. *The 5th Science and Technology Basic Plan.* Technical Report. Microsoft.

[17] Ragunathan Rajkumar, Insup Lee, Lui Sha, and John Stankovic. 2010. Cyber-physical systems: The next computing revolution. In *Design Automation Conference.* 731–736.

[18] Anro Redelinghuys, Anton Basson, and Karel Kruger. 2019. A Six-Layer Digital Twin Architecture for a Manufacturing Cell. In *Service Orientation in Holonic and Multi-Agent Manufacturing*, Theodor Borangiu, Damien Trentesaux, André Thomas, and Sergio Cavalieri (Eds.). Springer, 412–423.

[19] Luis F. Rivera, Miguel Jiménez, Prashanti Angara, Norha M. Villegas, Gabriel Tamura, and Hausi A. Müller. 2019. Towards Continuous Monitoring in Personalized Healthcare through Digital Twins. In *29th Annual International Conference on Computer Science and Software Engineering (CASCON '19)*. IBM Corp., 329–335.

[20] Microsoft Services. 2017. *White Paper: The promise of a digital twin strategy - Best practices for designers and manufacturers of products and industrial equipment.* Technical Report. Microsoft.

[21] Fei Tao, He Zhang, Ang Liu, and A. Y. C. Nee. 2018. Digital Twin in Industry: State-of-the-Art. *IEEE Transactions on Industrial Informatics* (2018).

[22] The OpenSpace Team. 2019. *Frequency: The New Dimension for Digital Twins.* Technical Report. Open Space Labs, Inc.

[23] Norha M. Villegas, Gabriel Tamura, Hausi A. Müller, Laurence Duchien, and Rubby Casallas. 2013. *DYNAMICO: A Reference Model for Governing Control Objectives and Context Relevance in Self-Adaptive Software Systems.* Springer, 265–293.

[24] Thomas Vogel, Andreas Seibel, and Holger Giese. 2011. The Role of Models and Megamodels at Runtime. In *Models in Software Engineering*, Juergen Dingel and Arnor Solberg (Eds.). Springer, 224–238.

[25] Fei-Yue Wang. 2010. The Emergence of Intelligent Enterprises: From CPS to CPSS. *IEEE Intelligent Systems* 25, 4 (July 2010), 85–88.

[26] Mark Weiser. 1993. Ubiquitous Computing. *Computer* 26, 10 (Oct. 1993), 71–72.

[27] Feng Xia, Xiangjie Kong, and Zhenzhen Xu. 2010. Cyber-Physical Control over Wireless Sensor and Actuator Networks with Packet Loss. *Computing Research Repository* abs/1011.3115 (2010).