Enabling Self-Adaptation of Renewable Energy Communities with a Capability-Based Digital Twin

Pio Antonio Perugini[1], Carmine Colarusso[1], Assunta De Caro[1], and Eugenio Zimeo[1, 3]

[1]*Dept. of Engineering*, Università degli Studi del Sannio, Benevento, Italy
[3]*CINI Smart Cities and Communities Lab*

***Abstract*—This work introduces both a capability-based digital twin model for Renewable Energy Communities (RECs) based on the Asset Administration Shell (AAS) standard and an architecture to realize it. RECs empower citizens to actively engage in the energy transition through a decentralized approach to energy production and consumption, yet they face challenges such as fragmented digital infrastructures and lack of standardization across appliances and energy systems. To overcome these hurdles, we model REC entities as self-* digital twin described by high-level, vendor-neutral capabilities. The proposed architecture enables real-time energy management and adaptability, establishing a foundation for future self-* functionalities within the REC digital twin ecosystem.**

***Index Terms*—AAS, Digital Twin, Autonomic Computing, Renewable Energy Community, Self-Adaptive**

I. Introduction

Digital twins are emerging as the technological backbone for understanding, engineering, and managing cyber-physical systems. They are being researched, developed, and deployed in a wide range of domains, including smart cities, manufacturing, and more [1]. As digital twins become more heterogeneous, enabling them to communicate and collaborate has become a major challenge. Moreover, they are often tightly coupled with their physical counterparts, which hinders their re-usability and limits interoperability. These limitations are particularly problematic in scenarios where digital twins from different vendors must work together. To address these challenges, the concept of autonomic digital twins can be explored. Autonomic computing, introduced by IBM in 2001, proposes that complex systems should manage themselves with minimal human intervention [2], [3]. A related concept has been introduced in the manufacturing domain under the term Plug & Produce: i.e., components can be added, removed, or adjusted with minimal human intervention, enabling systems to adapt dynamically to new tasks and workflows [4]. This paradigm is equally relevant in digital twin ecosystems, where the ability to dynamically integrate new digital twins would allow systems to adjust to changing requirements. This concept closely aligns with the self-configuration property of autonomic systems and forms the foundation for enabling further self-* capabilities such as self-optimization and self-healing. A high-level abstract description of the functional capabilities of a digital twin, decoupled from the specifics of its physical asset, is fundamental to enable Plug & Produce scenarios. These *capabilities* serve as semantic contracts that define what a twin can offer, independently of how the functionality is implemented. This mechanism allows deployed modules to autonomously advertise their offered capabilities and interoperate with other twins to create or adapt systems on demand [5]. Renewable Energy Communities (RECs) enable citizens to participate directly in the energy transition, offering a decentralized approach to energy production and consumption. They provide significant benefits, including increased energy efficiency, cost savings, reduced energy poverty, and the creation of local green jobs. Despite their potential, the effective management and optimization of RECs are hindered by the lack of a unified digital infrastructure for real-time monitoring and the absence of standardized models for appliances and energy systems. To overcome these challenges, REC components can be represented as digital twins using the Asset Administration Shell (AAS), a vendor-neutral standard, that supports semantic interoperability. This approach allows for the definition of high-level, ontology-based concepts specific to the REC domain, enabling self-adaptive systems where members and appliances can be dynamically integrated without manual reconfiguration. This paper proposes two key contributions. First, it explores and demonstrates the applicability of the AAS standard beyond manufacturing, showing that the same principles that enable production line integration and management can be effectively extended to broader digital ecosystems [1]. Second, the paper proposes and validates a modular, domain-agnostic architecture for autonomic digital twins based on AAS. It features a portable *Autonomic Manager* supporting self-* functionalities and interoperability across heterogeneous assets. The approach is validated via a Renewable Energy Community (REC) case study, demonstrating effective management of distributed resources.

II. Digital Twin Modeling

The AAS, developed under the leadership of the International Digital Twin Association (IDTA), provides a standardized digital representation of any asset (a domain valuable physical or digital element). It serves as the single source of truth for asset-related data and is structured hierarchically through modular units known as submodels, each capturing a specific functional or descriptive aspect of the asset [1]. The AAS integrates diverse asset information and acts as a standardized communication interface to support interoperability. To ensure semantic consistency and reusability, submodels are based on standardized templates provided by the IDTA. These templates define structured parameters with unique identifiers and semantic descriptions, often aligned with well-established vocabularies such as ECLASS and the IEC Common Data Dictionary. In this paper, we adopt the AAS standard to define the digital twin of a REC based on a behavior-enabled IoT (BeT) use case [7]. Within the BeT framework, a REC is represented as a system composed of multiple Energy Storage Systems and a variety of members. Members are classified as *Producers*, who own energy generation systems, *Consumers*, who own only energy-consuming devices, and *Prosumers*, who possess both generation and consumption capabilities. Each appliance is treated as a smart device, capable of transmitting its consumption data and receiving control signals. Appliances are classified based on their flexibility and controllability into *shiftable* and *non-shiftable* loads. Non-shiftable appliances include both interruptible and non-interruptible fixed loads. Shiftable appliances are further distinguished into *non-interruptible* devices, which must complete their operation once started, and *interruptible* devices, which can be temporarily paused. For interruptible devices, a minimum uptime constraint specifies the number of time slots they must remain active once switched on. In this context, the REC is conceived as a digital twin of digital twins: each building is modeled as a digital twin that encapsulates the behavior and characteristics of its associated appliances. This multilevel modeling enables accurate computation of both energy consumption and production across the community. The digital twin of each building aggregates data from connected appliances to derive building-level insights. For each device, information such as timestamped switch events and periodically sampled energy consumption is continuously collected. The availability of structured, real-time data enables REC Twin to offer services such as energy flow optimization. To this end, the REC Twin includes a scheduler that computes production and consumption plans based on device flexibility and aggregated data. When direct control is possible (e.g., shifting or pausing operation), the plan is enforced automatically (Self-Optimization). Otherwise, the REC Twin switches to advisory mode, issuing personalized recommendations to promote efficient and sustainable behaviors, ultimately lowering energy costs. Given the dynamic nature of the REC ecosystem, the REC Twin must support automatic discovery and integration of new devices. This self-configuration capability, also known as *plug and produce* in industrial contexts, is achieved by modeling systems through high-level abstractions of their *required* and *provided capabilities*.

III. AAS-based Autonomic Digital Twin Architecture

While the AAS provides a standardized and extensible structure to describe both static and dynamic aspects of a physical asset, it alone does not enable the bidirectional communication and closed-loop control that define true digital twins. To realize autonomic behavior, additional software components are required. This architecture introduces an external, portable *Autonomic Manager* that interfaces with any AAS-compliant digital twin and implements the MAPE-K control loop. Using AAS submodels that describe capabilities, skills, interfaces, technical attributes, and runtime data, the Autonomic Manager enables runtime adaptation, self-management, and orchestration across heterogeneous assets. The process starts with a high-level goal defined by required *capabilities*. The Autonomic Manager discovers available capabilities across twins, selects the optimal one based on criteria, identifies the corresponding *skill* (the capability's concrete implementation), and retrieves invocation details via the asset's interface description. The skill is then automatically invoked, enabling end-to-end autonomic orchestration. To support this, a set of AAS submodels provides the necessary semantic and operational information. A *Capability Description* submodel, inspired by preliminary specifications, abstracts and semantically describes the twin functionalities. The link between capabilities and their concrete implementations (skills) is established through the *Control Component* submodel. The *Asset Interface Description* submodel specifies the twin's communication interfaces. Using an Interface Description Decoder, the Autonomic Manager retrieves endpoint and invocation semantics for autonomous execution. The *Asset Interface Mapping Configuration* submodel maps abstract interface definitions to concrete asset data fields, enabling dynamic and automatic data feeding for real-time adaptation. For dynamic composition, the *Bills of Material* submodel models a digital twin as a composition of twins. Finally, the *Technical Data* submodel captures essential physical and logical asset characteristics critical for capability selection during orchestration.

IV. Conclusion

This work presents a domain-agnostic modular architecture for AAS-based autonomic digital twins, featuring a portable *Autonomic Manager* that uses MAPE-K to enable dynamic orchestration and self-* capabilities across heterogeneous assets.

References

[1] J. Zhang, C. Ellwein, M. Heithoff, J. Michael, and A. Wortmann, "Digital twin and the asset administration shell," *Software and Systems Modeling*, Jan. 2025.

[2] P. Horn, "Autonomic computing: IBM's perspective on the state of information technology," IBM, 2001. [Online]. Available: http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf

[3] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing—Degrees, models, and applications," *ACM Computing Surveys*, vol. 40, no. 3, Art. no. 7, Aug. 2008.

[4] X. Ye, J. Jiang, C. Lee, N. Kim, M. Yu, and S. H. Hong, "Toward the plug-and-produce capability for Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 14, no. 4, pp. 72–81, Dec. 2020.

[5] A. Köcher, C. Hildebrandt, L. M. V. da Silva, and A. Fay, "A formal capability and skill model for use in Plug and Produce scenarios," in *Proc. IEEE 25th Int. Conf. Emerging Technol. Factory Autom. (ETFA)*, Vienna, Austria, Sep. 2020.

[6] Industrial Digital Twin Association (IDTA), *Specification of the Asset Administration Shell Part 1: Metamodel*, Version 3.0, Frankfurt am Main, Germany, Mar. 2023. [Online]. Available: https://www.industrialdigitaltwin.org/

[7] A. De Caro, E. Zimeo "Behavior enabled IoT: A software architecture for self adapting a renewable energy community," *Internet of things*, May 2024.