Proceedings of the Design Society, Volume 5: ICED25
https://doi.org/10.1017/pds.2025.10283

Redefining interfaces: a generic interface architecture for complex system integration

Daniel Aron and Yoram Reich

Tel Aviv University, Israel

yoramr@tauex.tau.ac.il

ABSTRACT: This paper proposes a redefinition of interfaces as dynamic, adaptive systems crucial for managing the increasing complexity of modern systems. Drawing on diverse domains, the paper identifies key interface properties such as adaptability, cost-efficiency, and error response. The paper introduces a novel Generic Interface (GI) architecture, utilizing a model-based systems engineering approach. The GI architecture features modular components, designed to handle integration, data management, and error resolution. A case study of smart grids demonstrates the effectiveness of the GI architecture in addressing challenges like integrating diverse energy sources, ensuring grid reliability, and enabling demand response. The proposed GI architecture provides a robust framework for integrating complex systems, emphasizing adaptability, cost optimization, and error response.

KEYWORDS: systems engineering (SE), design for interfaces, product architecture

1. Introduction

As modern systems evolve, their increasing complexity and integration requirements necessitate a significant focus on the interfaces between subsystems to support diverse interaction patterns. Integrating components from various sources in system development often presents challenges like data incompatibility, communication protocol mismatches, and difficulties in synchronization, dependency management, security, error handling, and debugging. Integrating diverse components requiring real-time decision-making, adaptability, and scalability, further complicates the challenges. This situation is further exacerbated in future scenarios where diverse ad-hoc socio-technical systems will operate (Reich et al., 2023; Tozic & Reich, 2023).

Traditional interfaces, often seen as static connectors of importance secondary to the system's components, may struggle to handle these complexities effectively; they are inadequate for meeting the dynamic requirements of systems like Smart Grids, where scalability, adaptability, and real-time interaction are essential. Using Smart Grids as a case study, this work illustrates how defining the concept of an interface can tackle future system engineering challenges and support the evolving needs of complex systems.

Rather than focusing on modular adaptable or self-organizing system architectures, this research focuses on the interfaces between the system's components; it redefines interfaces as dynamic, adaptive systems that effectively manage diverse interactions, error responses, and system scalability; such interface is termed 'Generic Interface (GI).' The paper presents GI design guidelines and architecture as new concepts in system engineering. By combining model-based systems engineering (MBSE) to develop the GI architecture and demonstrating it with real-world applications, this study lays the groundwork for a new generation of system interfaces that prioritize flexibility, efficiency, and adaptability.

The remainder of this paper is organized as follows: Section 2 reviews the literature on traditional interfaces and relevant domains. Section 3 redefines the concept of an interface and discusses the key properties of interfaces in complex systems. Section 4 outlines the methodology for developing the Generic Interface (GI) through a Smart Grids case study. Section 5 presents the GI architecture through its application in Smart Grid systems, demonstrating scalability and adaptability. Section 6 discusses some issues and future work while Section 7 summarizes key contributions.

2. Literature review

This section examines the traditional definition of an interface and, subsequently, existing research on interfaces across various disciplines leading to a comprehensive understanding of their current limitations and the need for a new Generic Interface definition.

2.1. Traditional interfaces

An interface can be defined as “(1) the place at which independent and often unrelated systems meet and act on or communicate with each other [and] (2) a surface forming a common boundary of two bodies, spaces, or phases” (Merriam-Webster, 2024). The words *inter* and *face* combined mean the common boundary of two components. A definition that diverges slightly from those considers an interface as an element that unrelated components use to interact. This position allows a more elastic view of interfaces that depends on the granularity of the perspective in which we view the interface (Engel & Reich, 2015). For example, considering Figure 1(a), the interface could be defined as the bolts and the gaskets between the flanges. Another definition may consider the interface as the six flanges and a third definition would consider the Tee Pipe with its three flanges as the

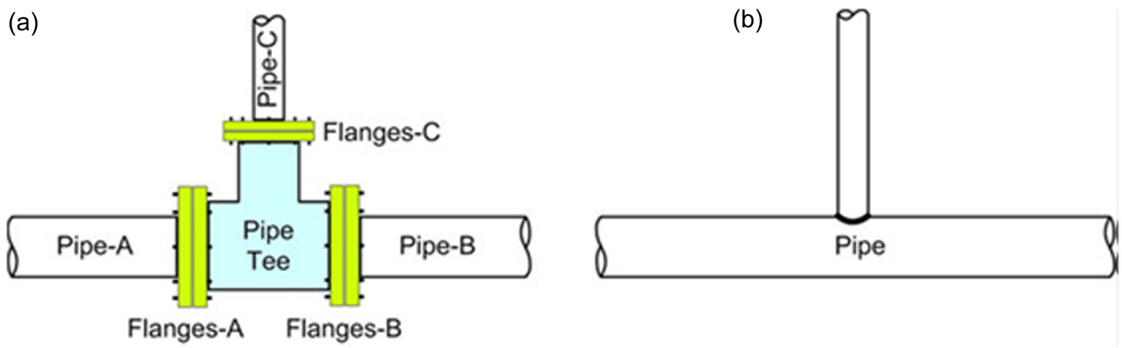

Figure 1. Two types of interfaces (Engel & Reich, 2015)

interface. Figure 1(b) defines the interface as the welding between the two pipes.

Interfaces may serve several purposes. They provide physical connection and support and may transfer energy, information, and material (Eppinger & Browning, 2012; Engel & Reich, 2015). These interfaces are often defined by fixed protocols, rigid specifications, and pre-determined roles within the system architecture. Their design tends to focus on ensuring compatibility at the time of system development, often at the expense of long-term scalability and adaptability. Traditional interfaces are typically conceptualized as static, often standardized, connectors between system components. While effective in structured, predictable environments, traditional interfaces are inflexible to adapt to dynamic conditions, evolving requirements, or unexpected interactions.

The traditional way to deal with integrating diverse components and adapting systems to new requirements is by using standardized interfaces. They allow replacing one module with another having similar or improved functionality without changing the interface (Martin & Ishii, 2002). However, systems including standard interfaces have cost associated that need to be considered (Engel & Reich, 2015), hence providing limited support over future needs.

As systems become increasingly complex and interconnected, the limitations of traditional interfaces—such as difficulty in integrating new components or managing real-time interactions—highlight the need for more advanced, adaptive interface models capable of meeting the demands of modern engineering systems.

2.2. Innovative approaches to interface design

TRIZ, the Theory of Inventive Problem Solving (Petrov, 2019), provides insights into innovative problem-solving through principles derived from patent analysis. TRIZ has been previously applied to interface design, for example, TRIZ principles were used to resolve conflicts in interfaces between components (Wits & Vaneker, 2011) or to improve different properties of user interfaces (UI) (Mishra, 2009). More importantly, applying TRIZ principles to interface design offers a fresh insight into interfaces as a system. TRIZ principles such as "Intermediary" and "Another Dimension" suggest viewing interfaces not only as points of connection but as systems in themselves. This conceptual shift enables the application of TRIZ principles to enhance interface functionality and adaptability. TRIZ principles like "Dynamics," "Universality," "Beforehand Cushioning," and "Self-service" highlight the need for interfaces to be adaptable, multi-functional, capable of error handling, and self-sustaining. These properties form the foundation for a new interface definition capable of meeting the demands of complex systems. By applying TRIZ principles, interfaces are seen as dynamic systems that actively contribute to the overall performance and adaptability of their parent, larger system. By treating interfaces as systems governed by TRIZ rules, interfaces are not just connectors; they are versatile, adaptable systems that enhance the overall system's resilience, scalability, and ability to evolve in response to new challenges.

2.3. Interfaces in SoS modeling and Multi-Agent Systems

Research on Multi-Agent Systems (MAS) emphasizes the importance of obligation relations between agents (Gutierrez-Garcia et al., 2010). This concept can be applied to interface design, where interfaces act as facilitators of these obligations by enabling communication and interaction between system components, ensuring each component fulfills its responsibilities. By integrating the concept of obligation relations into the design of interfaces, we can create systems that are not only adaptable and scalable but also capable of maintaining a high level of coordination and efficiency. Interfaces, in this sense, become the backbone of the system's communication and interaction framework, ensuring that all agents can fulfill their obligations effectively, thus contributing to the system's overall stability and performance. Similarly, the SPEEDS project (Engel et al., 2008) introduces the concept of contracts composed of assumptions and promises to manage interactions between system components. This approach can be used to define clear expectations for interface behavior, enhancing reliability and predictability in system interactions.

While interfaces in complex systems have been studied from various perspectives, with emphasis on structured frameworks, modularity, and real-time data exchange, these approaches provide valuable insights into managing interactions across system components (Fosse & Delp, 2013; Shen & Su, 2012). They often treat interfaces as fixed or narrowly defined connectors, limiting their adaptability in dynamic environments. Current research has yet to fully address interfaces as adaptive, context-sensitive components that can evolve alongside system demands, responding both to immediate operational needs and future scalability. This work aims to bridge this gap by redefining interfaces to enhance both real-time flexibility and long-term adaptability.

2.4. Financial and contractual perspectives in interface engineering

The AMISA project (Engel & Reich, 2015) introduces the concept of treating interfaces as "real options" in financial terms. This approach allows system designers to evaluate the cost and value implications of interface and module choices, enabling strategic decision-making regarding upgrades and modifications while maintaining overall system efficiency. The AMISA project, through its Architecture Options framework, expands the definition of interfaces by integrating financial evaluation and strategic planning into its design. Interfaces are highlighted as vital components contributing to the system's long-term adaptability, scalability, and cost-efficiency. This approach underscores the role of interfaces as dynamic and strategic elements within complex system architectures.

2.5. User Interfaces (UI) and system interface definition

User Interfaces (UI) provide insights into how interfaces can be designed to facilitate intuitive and efficient user interactions (Beaudouin-Lafon, 2004). Modern UIs are characterized by adaptability, personalization, and context-awareness, highlighting the need for interfaces to be responsive to user needs and system changes (Mishra, 2009). The example of a scrollbar on a web page, which acts as a user interface element between the user and the content displayed serves as an interactive tool that enhances user experience by translating actions like scrolling into system responses. This example reinforces the argument that a UI is more than a passive conduit for user commands; it actively shapes the interaction by offering intuitive controls and real-time feedback.

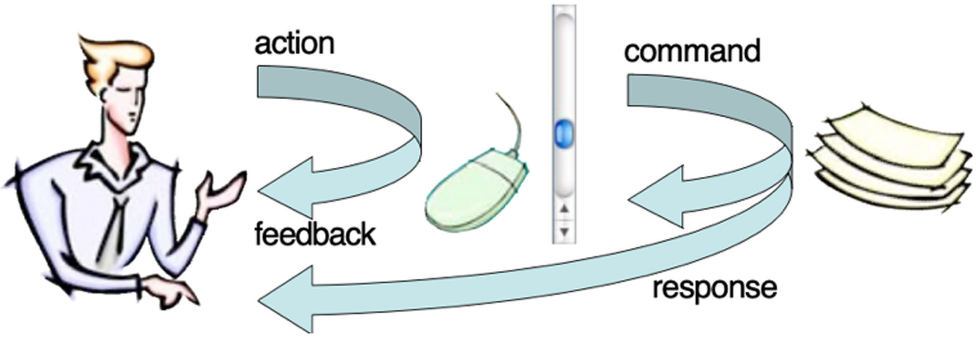

Figure 2. Scrollbar as an example of an instrumental interaction (Beaudouin-Lafon, M., 2004)

The UI's design reflects modularity, allowing for the dynamic connection and disconnection of components, which enables the integration of new modules or isolation of faulty ones without disrupting the entire system. Ultimately, the UI transforms the traditional concept of an interface from a basic connector to a sophisticated component essential for system operation. It manages and optimizes connections, ensuring adaptability, scalability, and security over time, marking a significant evolution in the definition of interfaces needed for modern, complex systems.

2.6. Summary of literature review

In reviewing the various perspectives from the literature, we identified three key themes essential for effective interface design: adaptability, cost, and error response. Table 1 summarizes these themes, with each property defined based on insights from TRIZ, SoS, MAS, and other relevant studies. This structured approach guarantees that the assumptions supporting our new interface definition are both comprehensive and rooted in existing research.
Despite the contributions related to the properties summarized in Table 1, there remains a void for a design methodology of adaptive and evolving interfaces for supporting the evolution of complex systems. We further note that while we focus in this paper on three properties, introducing additional properties could be handled as well with our approach.

Table 1. Interface properties that emerge from the literature

#	Property	Description	Sources
1	Adaptability	An adaptive interface can connect with new modules using different communication protocols, requiring hardware and software integration for interpretation and adjustment.	TRIZ – Dynamics, Gutlérrez-Garcia (2010), Engel et al. (2008), Nel et al. (2009), Fosse and Delp (2013), Shen and Su (2012), Zhang, Xue, and Gu (2015)
2	Cost	The interface's adaptability may differ in price tag due to the need for suitable hardware and software integration.	Engel and Reich (2015), Zhang, Xue, and Gu (2015), NASA (2016)
3	Error response	Compliance must be maintained in the face of various potential issues. These issues may include module identifier changes, naming collisions, telemetry loss, or unresponsive modules.	TRIZ – Self-Service and Skipping model, Nel et al. (2009)

3. Interfaces: redefinition and characteristics

This section proposes a new broad definition of an interface, considering the insights from the literature review. *An interface is a dynamic boundary that enables interaction and communication between different systems, modules, or components.* This definition encompasses both simple connectors, like how LEGO units connect, and complex mechanisms, such as subway control systems, which manage intricate interactions. Interfaces can be classified as simple or complex. A complex interface not only mediates but also provides error handling, supports connections across various mediums, and engages in systematic decision-making. It represents an architecture of connections and decisions across different fields, allowing for a novel thinking framework. This view aligns with Systems of Systems (SoS) characteristics noted in the literature (Zhou et al., 2023; Fosse and Delp, 2013), where an interface acts as a dynamic agent facilitating communication and adaptability among components. Key characteristics of an interface include adaptability, error management, scalability, and the capacity to evolve with future changes. The key characteristics—adaptability, cost, and error response—are characteristics essential for managing interactions within complex systems. Adaptability ensures that interfaces can dynamically adjust to changing operational conditions, integrating new components or subsystems seamlessly. It is important to differentiate between adaptability and resilience. Adaptability refers to the interface's ability to incorporate new components and adjust to changing conditions. In contrast, resilience encompasses this adaptability along with the capacity to maintain operational stability and recover from errors or external shocks. A resilient interface not only adapts but also withstands adverse conditions, ensuring continuous performance in the face of disruptions. Cost efficiency highlights the importance of designing interfaces that minimize resource usage, both during integration and throughout the system lifecycle, ensuring scalability without excessive expenditures. Error response reflects the ability of interfaces to detect, isolate, and correct faults in real-time, maintaining system reliability and reducing downtime. A distinctive feature of these properties is their extensibility, enabling the integration of additional characteristics as the system needs evolve. For example, applying security as a future interface property would enhance the system's ability to safeguard data through mechanisms like encryption and access control. This capacity to extend interface properties demonstrates their flexibility and relevance in addressing the challenges of emerging technologies, ensuring that the interface remains a robust and scalable solution for dynamic and evolving systems.

4. Methodology - Smart Grid systems as a case study

A case study methodology is employed to explore the implementation and impact of interfaces within complex systems. The Smart Grid is chosen as the primary case study due to its highly integrated and distributed nature presenting a valuable context for analyzing the effectiveness of interfaces in managing diverse communication protocols, real-time data exchange, and system scalability. The study aims to identify key characteristics of interfaces that contribute to adaptability, error management, and cost-efficiency within the Smart Grid. Through developing interfaces for this case study we design, evolve, and validate the architecture of a Generic Interface (GI).

Smart Grids are advanced electrical power grids that leverage modern technologies to enhance the efficiency and reliability of electricity distribution. They integrate diverse energy sources, including renewable energy, and utilize advanced communication systems, automation, and data analytics to create a more sustainable and efficient energy ecosystem. Some of the challenges of Smart Grid implementation include

1. Energy Sources Integration: Seamless integration of various energy sources, including solar, gas, and wind power.
2. Grid Reliability: Enhanced reliability through real-time monitoring and automated fault detection.
3. Demand Response and Load Management: Empowerment of consumers to manage energy consumption based on pricing signals or demand response programs.
4. Grid Optimization and Efficiency: Optimization of electricity distribution using data analytics and predictive modeling.
5. Electric Vehicle Integration: Support for electric vehicle charging infrastructure and intelligent charging management.
6. Energy Management and Billing: Detailed energy consumption monitoring for consumers and accurate billing based on actual consumption.

If we consider the aforementioned properties of interfaces concerning Smart Grid we obtain:

7. Adaptability: Smart Grids require adaptable interfaces to integrate new technologies, manage fluctuating energy demands, and adapt to regulatory changes.

8. Cost: The cost of implementing and maintaining Smart Grid systems, including infrastructure upgrades, technology integration, and system maintenance, is a significant consideration.
9. Error Response: Smart grids rely on robust error response mechanisms to maintain grid reliability and minimize the impact of faults or disruptions.

Our goal now is to design an interface for Smart Grid that addresses the challenges and provides the required functionality of Smart Grid.

5. Generic Interface architecture

Integrating various systems into a cohesive whole is a key challenge in system engineering. The architecture presented in this chapter through Object Process Methodology (OPM) diagrams (Dori, 2016), a model-based systems engineering (MBSE) approach, provides a solid framework for Generic Interfaces. OPM provides modularity in developing the GI architecture through studying the Smart Grid case study. OPM also provides a simulating tool that ensures the model's correctness and evolution for new scenarios. The architecture presented next was developed by iterative refinement with use cases and simulations available in OPM. This architecture enhances adaptability, optimizes costs, and ensures effective error response by using a Generic Interface that connects components like sensors, adapters, and action modules, enabling seamless communication and data flow.

5.1. Guidelines for designing a Generic Interface (GI)

The guidelines for designing a Generic Interface (GI) focus on several issues to ensure seamless integration and long-term evolution of the system. *Modularity* allows components to be added, replaced, or upgraded without disrupting system operations, while *scalability* ensures that the interface can accommodate future growth and technological advancements. *Standardization* is essential for ensuring compatibility between system components and simplifying integration processes through shared communication protocols and data formats. Additionally, *flexibility* enables interfaces to dynamically adjust to changing operational requirements, addressing current and emerging system demands. Effective *error management* is necessary, with interfaces capable of autonomously detecting anomalies and implementing corrective actions. Finally, *cost efficiency* is a foundational principle that minimizes operational and maintenance expenses throughout the system's lifecycle. Together, these guidelines support the development of robust and flexible interfaces for complex systems.

5.2. GI suggested architecture and framework

The proposed architecture for Generic Interfaces is illustrated through schematics, where each detailing the interaction and workflow among the various system components. The central figure, Figure 3, the Generic Interface, is the hub connecting multiple Sensor, Adapter and Action Modules. It consists of three main components responsible for three different objectives – the Integrating Unit (responsible for the integration and security checking of edge devices), the Memory Management Unit (responsible for data storage and management), and the Analysis Unit (responsible for resolving and analyzing stored data). Each unit is designed to perform specific functions, contributing to a cohesive system that maintains high reliability and efficiency in complex environments. In this distributed framework, each module operates independently yet cohesively. This decentralized data collection is foundational to the architecture's flexibility.

The Integrating Unit is the main point of contact for all incoming data, performing critical integration and security checks to maintain data integrity and authenticity of existing and newly integrated devices. It oversees validation and ensures a seamless flow of data from various modules. As shown in Figure 4, data from sensors first enters the Integrating Unit, undergoing a two-step process: Integration and Security Checking, followed by Component ID Validation. The first step verifies data authenticity using encryption and secure protocols to prevent tampering. The second step cross-references component IDs with a database to confirm legitimacy and compatibility. The Integrating

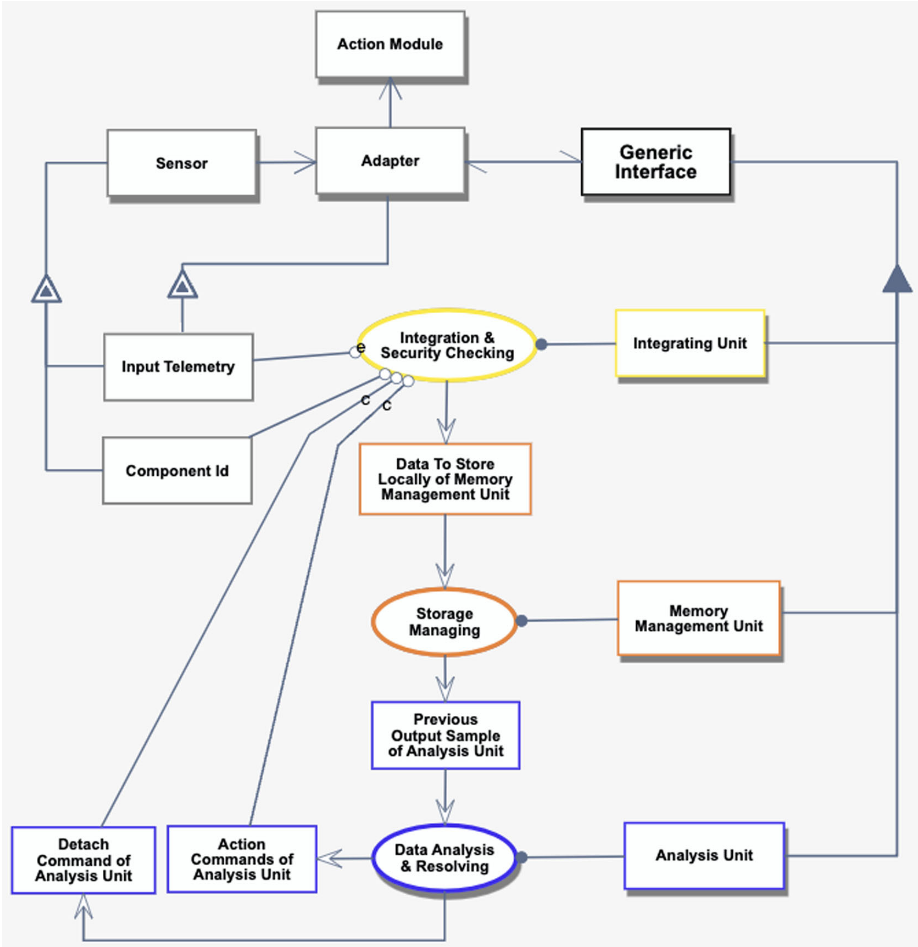

Figure 3. Main architecture flow for a Generic Interface

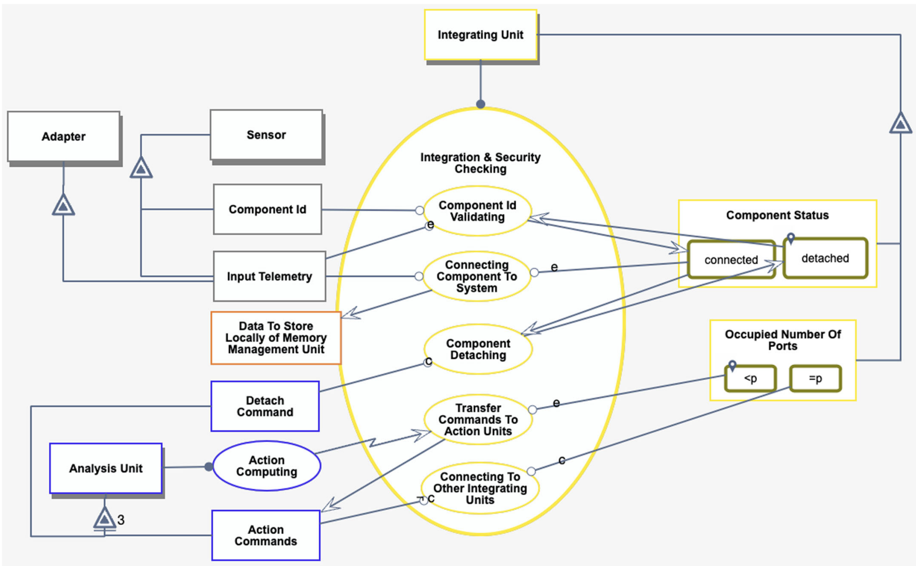

Figure 4. Integrating unit architecture

Unit can also connect or detach modules dynamically. If a component is faulty or unauthorized, it isolates it and alerts the Error Handling sub-unit. New components are integrated after passing the necessary checks.
The Memory Management Unit (MMU), Figure 5, is responsible for efficiently managing and storing real-time and historical data. Its primary functions include validating incoming data and facilitating

comparisons between current system performance and previously stored telemetry data. This enables the system to detect patterns, trends, and anomalies by utilizing the Previous Telemetry Comparing process. The MMU also manages the storage capacity of both Local Memory and the Database, ensuring that data storage is organized and accessible. In cases where past data is needed for decision-making or analysis, the MMU retrieves data via the Previous Data Collecting function. Furthermore, the MMU connects to other memory units, ensuring scalability and broad system integration, particularly crucial in complex environments like Smart Grids. The MMU's role is integral to the system's ability to maintain a comprehensive data history, enabling real-time adjustments and informed decision-making across the architecture.

The Analysis Unit serves to monitor and optimize system performance by analyzing output data and executing corrective actions when needed. Its main purpose is to compare the Desired Output with the Previous Output Sample, identifying any System Output Errors that exceed or fall below a defined threshold. If errors are detected, the Error Handling component processes them, and the Action Computing module determines the appropriate Action Commands to restore system stability. In cases of critical failure, a Detach Command can be issued to isolate problematic components. Overall, the Analysis Unit ensures continuous performance monitoring, error detection, and real-time adjustments, safeguarding system efficiency and reliability as outlined previously.

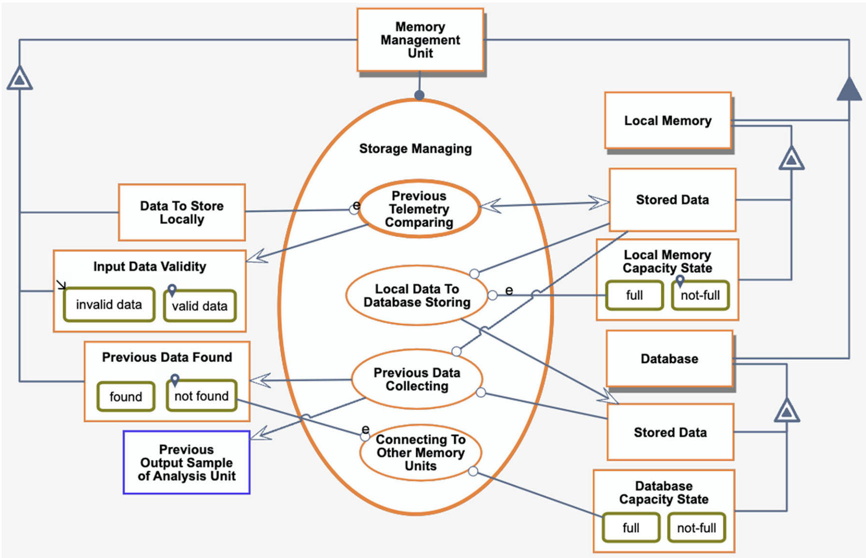

Figure 5. Memory management unit architecture

All the GI processes described above were simulated by the OPM facility to ensure their correctness. The modularity of the models and the simulation capability support continuous evolution and validation of the models for new scenarios and complex systems.

6. Discussion

The GI architecture in Smart Grids effectively addresses adaptability, cost optimization, and error response challenges. Its modular design allows for easy integration of new GI functionalities and technologies for Smart Grid subsystems; it enables efficient energy management and proactive fault detection, enhancing system resilience, cost-effectiveness, and long-term adaptability. Integration of components like the Integrating Unit, Memory Management Unit, and Data Analysis Unit optimizes resource allocation and maintains system stability. An important factor is the approach to connecting the GI modules - Depending on the scenario, the interface can range from simple to complex. In simpler cases, modules can connect directly to the GI; in more complex situations, additional GI modules can be interconnected to the existing GI, adhering to the same guidelines. In that case, the GI architecture remains generic when addressing the GI architecture as it can recursively be integrated into the model itself. In complex scenarios, interconnected GIs can operate concurrently, forming a collaborative system with decentralized control. In contrast to common system design practice where the focus is on the system components and their integration into a whole while the interfaces merely connect the components, we offer a complementary approach where the focus is on intelligent generic interfaces that can connect diverse components.

The GI architecture can be implemented in Smart Grid systems by leveraging the GI three main components. This architecture employs an Integrating Unit for real-time validation and security checks, a Memory Management Unit for storing and comparing historical and current telemetry, and a Data Analysis & Resolving Unit to detect anomalies and trigger corrective actions. By doing so, it directly addresses Smart Grid challenges such as inconsistent data flows, delayed error detection, and scalability issues. The architecture's ability to dynamically connect or detach components without major reconfiguration enhances grid stability, optimizes resource allocation, and improves overall adaptability to fluctuating energy inputs and operational demands. Rather than serving as a power manager, the GI is designed to function as an intelligent mediator that orchestrates real-time data exchange, error detection, and communication between diverse subsystems. This shift in focus ensures that even within power-centric environments, such as Smart Grids, the interface's primary role is to facilitate robust data connectivity and adaptability, ultimately optimizing system performance without directly governing energy distribution.

Future research could extend the real-world testing of the GI architecture into the healthcare domain. For example, a pilot project could deploy the GI framework in a hospital setting to integrate data from medical devices, patient monitoring systems, and electronic health records. This would enable real-time error detection and adaptive decision-making while utilizing machine learning algorithms to analyze patterns and predict potential system faults or patient deterioration. By incorporating advanced data analytics and ML, the architecture can not only streamline data flows and enhance system resilience but also tailor interventions based on predictive insights, ultimately improving patient care and operational efficiency.

7. Conclusion

In this research, we developed and tested a new definition of system interfaces and presented the design of Generic Interface (GI) architecture, driven by a case study of an evolving and scalable system. The new GI architecture underscores the importance of rethinking interfaces as adaptable and strategic components capable of evolving with system demands. The architecture supports complex system integration characterized by its modularity and scalability, ensuring long-term system flexibility and efficiency. It supports the redefined concept of interfaces as dynamic systems that play a crucial role in managing interactions, data flow, and error handling. Key components like the Integrating Unit, MMUs, and Data Analysis & Resolving Unit contribute to the system's ability to adapt to evolving demands, optimize resource allocation, and maintain stability. The Generic Interface architecture, while primarily designed for complex systems such as smart grids, is fundamentally focused on the management of information flow rather than on physical interconnectivity. Its modular design allows it to be scaled down for simpler systems where only basic data management is required, demonstrating its versatility across dynamic and static environments. Additionally, the smart grid serves as a compelling case study due to its dynamic and distributed nature, but the proposed Generic Interface architecture is inherently applicable to a wide range of complex systems.

This research contributes to the field of system design by refining the Interface Definition and expanding the traditional concept of interfaces to encompass dynamic and adaptable systems. Additionally, proposing a GI Architecture offers a modular and scalable framework for integrating various components and enhancing system resilience. By demonstrating the effectiveness of the GI architecture in tackling real-world challenges in smart grid systems, its impact on current system engineering issues is validated.

References

Ahmed, S. D., Al-Ismail, F. S. M., Shafiullah, M., Al-Sulaiman, F. A., & El-Amin, I. M. (2020). Grid Integration Challenges of Wind Energy: A Review. *IEEE Access*, 8, 10857–10871.

Albadi, M. H., & El-Saadany, E. F. (2008). A summary of demand response in electricity markets. *Electric Power Systems Research*, 78(11), 1989–1996.

Amin, S. M., & Wollenberg, B. F. (2005). Toward a smart grid: Power delivery for the 21st century. *IEEE Power and Energy Magazine*, 3(5), 34–41.

Beaudouin-Lafon, M. (2004). Designing Interaction, not Interfaces. Retrieved from [https://www.lri.fr/~mbl/papers/AVI2004/paper.pdf].

Dori, D. (2016). Model-based systems engineering with OPM and SysML (Vol. 15). New York: Springer.
Engel, A., & Reich, Y. (2015). Advancing Architecture Options Theory: Six Industrial Case Studies. *Wiley Online Library*.
Engel, A., Winokur, M., Döhmen, G., & Enzmann, M. (2008). Assumptions/promises-shifting the paradigm in systems engineering. *In INCOSE International Symposium* (Vol. 18, No. 1, pp. 58–78).
EppingerSD, BrowningTR (2012) Design structure matrix methods and applications. *MIT Press*.
Fang, X., Misra, S., Xue, G., & Yang, D. (2012). Smart grid – The new and improved power grid: A survey. *IEEE Communications Surveys & Tutorials*, 14(4), 944–980.
Fosse, E., & Delp, C. (2013). Systems Engineering Interfaces: A Model-Based Approach. *IEEE Aerospace Conference Proceedings*, 1–11.
Gutierrez-Garcia, J. O., Zaragoza-Rios, J. A., Ramos-Corchado, F. F., Koning, J.-L., Ramos-Corchado, M. A., & Siller, M. (2010). Integration of agricultural information systems assisted by knowledge. *Intelligent Automation and Soft Computing*, 16(6), 913–922.
Gharavi, H., & Ghafurian, R. (2011). Smart grid: The electric energy system of the future. *Proceedings of the IEEE*, 99(6), 917–921.
Gungor, V. C., Sahin, D., Kocak, T., Ergüt, S., Buccella, C., Cecati, C., & Hancke, G.P. (2013). Smart grid technologies: Communication technologies and standards. *IEEE Transactions on Industrial Informatics*, 7(4), 529–539.
Ipakchi, A., & Albuyeh, F. (2009). Grid of the future. *IEEE Power and Energy Magazine*, 7(2), 52–62.
Madni, A. M., & Sievers, M. (2018). Systems integration: Key perspectives, experiences, and challenges. *Systems Engineering*, 21(3), 203–220. [https://onlinelibrary.wiley.com/doi/10.1002/sys.21431]
Maier, M. W., & Rechtin, E. (2000). The Art of Systems Architecting. *CRC Press*.
Martin, M. V., & Ishii, K. (2002). Design for variety: developing standardized and modularized product platform architectures. *Research in engineering design*, 13, 213–235.
Merriam-Webster.com Dictionary, Merriam-Webster, https://www.merriam-webster.com/dictionary/interface. Accessed 4 Dec. 2024.
Mishra, U. (2009). Improving Graphical User Interface using TRIZ. *Umakanta Mishra*.
NASA. (2016). NASA Systems Engineering Handbook (NASA SP-2016-6105 Rev2). *National Aeronautics and Space Administration*.
Nel, A., Xia, T., Mädler, L., & Li, N. (2009). Toxic potential of materials at the nanolevel. *Science*, 311(5761), 622–627.
Palensky, P., & Dietrich, D. (2011). Demand-side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transactions on Industrial Informatics*, 7(3), 381–388.
Petrov, V. (2019). TRIZ Theory and Applications. *Cambridge University Press*.
Reich, Y., Sitton, M., Engel, A., Orion, U., Danielli, A., Hauptman, A., Blekhman, A., & Shabi, J. (2023, March). Context-Dependent Research Agenda for Systems Engineering in 2050. *In Conference on Systems Engineering Research (pp. 655-661)*. Cham: Springer Nature Switzerland.
Shen, Q., & Su, H. (2012). Information system integration model of manufacturing enterprise based on object process methodology and its application. *Asia-Pacific Journal of Chemical Engineering*, 7(5), 651–659.
Tozik, S., & Reich, Y. (2023). A Framework for Analysis and Design of Dynamic Ad Hoc Socio-Technical Systems. *Proceedings of the Design Society*, 3, 151–160.
U.S. Department of Energy. (2022). Smart Grid System Report. Retrieved from [https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/SGSRMain_090310.pdf].
Wits, W. W., & Vaneker, T.H. (2011). TRIZ based interface conflict resolving strategies for modular product architectures. *Procedia Engineering*, 9, 30–39.
Zhang, J., Xue, D., & Gu, P. (2015). Adaptable design of open architecture products with robust performance. *Journal of Engineering Design*, 26(1-3), 1–23. [https://doi.org/10.1080/09544828.2015.1012055]
Zhou, Z., Li, X., Zhou, X., & Wang, X. (2023). Modeling System of Systems: A Generic Method Based on System Characteristics and Interfaces. *Journal of Manufacturing Systems*, 67, 345–356.