Towards Web-based Autonomous Industrial Systems

Ganesh Ramanathan[1,*,†], Maria Husmann[1,†] and Simon Mayer[2,†]

[1]Siemens AG, Switzerland
[2]University of St. Gallen, Switzerland

Abstract

Industrial infrastructures employ control systems to manage their complex electro-mechanical installations in a safe and energy-efficient way. Semantic Web technologies have been utilized to create machine-understandable descriptions of requirements, technical design, and fundamental domain knowledge, yielding promising results in enabling the automated engineering of controls and, more broadly, in the development of autonomous systems. Concurrently, industrial automation systems are increasingly adopting the Web architecture. In this article, we first report on our experience gathered over the past decade at Siemens AG in bringing the fragmented Semantic Web-based ontologies in engineering towards creating *holistic system knowledge* of smart infrastructures. Then, based on our practical experience, we share that the role of Semantic Web technologies goes beyond enabling automated engineering and highlight its potential to create *knowledge-infused web-based autonomous systems*. However, we note that the potential is currently under-exploited in the context of evolution in industrial automation systems, where classical closed architectures are giving way to large-scale web-based deployments. We then describe our ongoing research to address this dissonance by adopting hypermedia multi-agent systems as an architectural paradigm in industrial automation.

Keywords

industrial automation, web-based autonomous systems, semantic web,

1. Machine-understandable Knowledge of Industrial Systems

Knowledge required for the engineering of industrial automation systems can be broadly categorized as the descriptions of (1) domain knowledge of control engineering, (2) system design, and (3) requirements (cf. [1]). In the following Sections 1.1-1.3, we summarize the state of the art in achieving machine-understandable representations of each of these aspects. We highlight that the respective ontological models were fragmented, thus hindering a holistic representation of the system knowledge. Simultaneously, we provide a brief overview of our past work, which addressed the missing integration of engineering knowledge. Then, in Section 2, we demonstrate how the achieved capability of a holistic system description is now leading us towards attaining autonomous behavior in evolving web-based systems in industrial automation.

1.1. Control Engineering

The foundational knowledge required to automate a system can be broadly categorized as (1) knowledge about the *physical processes* like energy conversion and heat exchange, and (2) knowledge about *control and coordination* strategies of such processes. Computational modeling of physical processes, a topic widely researched for many decades, is grounded in mathematical equations. However, a way to enable software agents to reason about and gain a *high-level* understanding of how physical processes work is not in widespread use. In [2], we proposed ontological concepts (in OWL) to describe the underlying physics qualitatively. With that approach, it is, for instance, possible to automatically infer that increasing the hot water flow input to a heat exchanger will raise the air temperature. Where

The Second International Workshop on Hypermedia Multi-Agent Systems (HyperAgents 2025), in conjunction with the 28th European Conference on Artificial Intelligence (ECAI 2025); October 26, 2025, Bologna, Italy

*Corresponding author.
†These authors contributed equally.
ganesh.ramanathan@siemens.com (G. Ramanathan); maria.husmann@siemens.com (M. Husmann); simon.mayer@unisg.ch (S. Mayer)

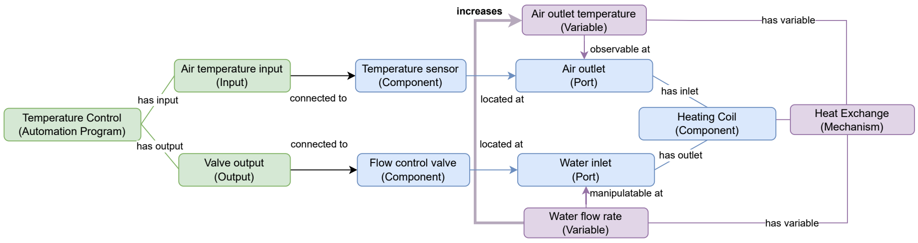

Figure 1: An example description of a control program that is also linked to the description of the system and the physical process it has been designed for. Using such a description, an agent can infer the suitability of the program to achieve its goal(s)

a more elaborate explanation of the process *dynamics* is required (e.g., to know how long it would take to raise the temperature), we further proposed a way to link the description of simulation tools (that embed a suitable mathematical model) so that the software agents may use it to predict more detailed process response (cf. [2]). Semantic Web-based qualitative modeling of physics also served as the foundation to describe how the control and coordination programs work in conjunction with the physical process. Hitherto, the programs were described solely based on their inputs, outputs, and state machines representing their *internal* functioning. By infusing physics knowledge into the description of the programs, we can now both describe the goals of the program in terms of what it will accomplish in the physical environment and identify which variables in the addressed physical mechanism are associated with the program's inputs, outputs, and parameters. We created an openly shared ontology called Elementary [1] [2], which provides high-level concepts to integrate the currently fragmented ontologies in engineering.

1.2. System Design

A technical system can be described in terms of its components, their taxonomical classification, and the topological inter-relationships. Several domain-specific ontologies in engineering, such as Brick, IFC, SAREF, etc., are available today for this purpose. For example, using the Brick ontology, it is possible to state that a central hot water boiler supplies a heating radiator in a room. However, a way to describe the *physical behavior of a component* was missing; for example, a way to infer that by opening the valve of a radiator, the temperature of the surrounding air would increase. Similarly, a way to express the physical dependencies between the components was missing; for example, to state that the boiler supplies the thermal energy required by the radiator. For this purpose, we introduced concepts in Elementary that enable designers to describe *stereotypical* components that include the qualitative model of their underlying physical mechanisms [2]. In this way, given an instance of a component (like a radiator), a software agent can infer the actions it may invoke on the component and also understand their consequent physical effects before deciding whether to invoke these actions through the component's API. Such an agent now also has access to knowledge that can be used to infer dependency on other components (e.g., an agent tasked with managing the radiator knows that it needs to coordinate with the agent managing the boiler). When it comes to semantically describing the API (to act on a component and perceive its current state), we found that the approach of Web of Things *Thing Description* (TD) [2], an RDF document that describes the interactions offered by technical components, is uniquely suited because they can be directly infused with knowledge about the physical processes [3].

[1] http://w3id.org/elementary
[2] https://www.w3.org/TR/wot-thing-description11/

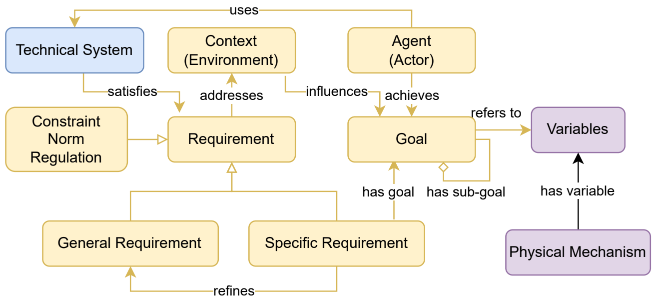

Figure 2: The description of requirements following the GORE model can be linked to both the system design and physical process descriptions. In this way, an agent becomes aware of the system components that were deployed to fulfill a requirement.

1.3. System Requirements

Several ontologies exist (see [1]) that facilitate the description of requirements and the decomposition of its goals. However, to the best of our knowledge, it is only the approach of goal-oriented requirements engineering (GORE) [4] that provides a way to *formally describe the goal* in terms of the physical variables. However, a reusable Semantic Web-based ontology of GORE was missing. Furthermore, concepts in GORE were not connectable to the system design (e.g., stating that the temperature of a room is maintained using radiators). We modeled GORE using OWL and provided relationships to link the requirements to system components and goal variables to the physical process. As a result, for example, it can be used to automatically infer which system components are involved in fulfilling a requirement and if, indeed, it is physically feasible to achieve the goal.

Furthermore, based on the integrated knowledge of controls and system design described in the previous subsections, we are now able to match control and coordination programs that are compatible with a given system installation and its requirements. Hence, the otherwise required manual engineering is avoided.

Our vision is, however, to progress beyond automated engineering towards attaining *autonomous* automation. We argue that the kind of autonomy observed in traditional automation systems is limited to a priori designed strategies that are valid for runtime conditions that conform to design-time assumptions. In the next section, we will describe how web-based automation systems can benefit from an integrated and holistic system description, enabling not only the selection of suitable control and coordination strategies at runtime, but also the dynamic adaptation to changes.

2. Web-based Systems in Industrial Automation

We are witnessing a clear shift in technologies and architectural patterns in industrial automation. Vendor-specific and proprietary implementations of automation controllers and peripherals are giving way to more Web-based systems [5, 6, 7]. A key driver for this change is the rapid increase in the capacity, bandwidth, and reliability offered by local and wide area networks in industrial installations [8, 7]. Concurrently, the capability of embedded devices in terms of computing power and network connectivity has improved significantly, driven by the cost-effective and capable system-on-chip (SoC) microcontrollers. Consequently, the reasons to use legacy communication protocols, such as Modbus (see [9] for a comprehensive overview), which were designed to cater to constrained networks, are no longer compelling. For example, HTTP-based communication with automation devices, which was once considered impractical, is now easily implementable and satisfies the performance needs of commonly known use cases [10]. In general, the boundaries established by the classical three-layer architecture consisting of field devices, automation controllers, and supervisory systems that is well-known in traditional automation systems (and standardized in IEC 61499 and IEC 61850) is blurring away while heading towards a more flatly distributed and federated architecture where even peripheral devices like sensors and actuators, which were once hardwired to and in the space of the automation controllers through analog signaling, are now accessible directly over the industrial communications networks [11, 12]. For example, using networking technologies like OpenThread [3], even low-power devices can be directly accessed using CoAP over an IPV6 network without going through manually engineered gateways. The timely effort of W3C's Web of Things group towards standardizing the description of interactions offered by such devices has proven to be of significant value in achieving technical interoperability [6]. But we argue that though Web-based hypermedia *interactions* that are being adopted in automation systems are indeed advancing technical interoperability, the reliance on off-band exchange of knowledge regarding requirements, system design, physical processes, and control strategies is hindering the progress towards attaining truly autonomous automation. This dissonance is especially noteworthy because the standards behind the Semantic Web are designed to enable *semantic interoperability* in the Web context. Therefore, to achieve autonomously adaptive systems, we require an architectural paradigm that enables the embedding, loosely coupled access, and use of semantic descriptions in large industrial automation systems.

The part of the automation system that can most directly benefit from machine-understandable system knowledge is the control programs and system-level applications, such as automated fault detection and diagnostics [13]. Such software programs, instead of running on predefined hosts, are now being flexibly deployed on infrastructure, such as available edge devices and runtime hosts in the cloud. For example, a cloud-based application such as IFTTT [4] can be easily used to set up control of a Web-connected sun blinds in a room based on data obtained from a Web-based weather service. However, when it comes to *engineering* the controls of such systems, the advances in technology developments leading towards such web-based deployments are not complemented by an approach that allows exploitation of the available machine-understandable descriptions of the working of a system. For example, in order to formulate the reactive control logic for controlling the lighting and blinds, one has to (manually) understand the physical interplay of outdoor and indoor lighting and the functioning of the sun blinds. Furthermore, if the basis on which the control logic was designed changes, for example, through the addition of lamps in the room or by changing the type of sun blinds, the program must be manually redesigned. This is despite the possibility we have today (see Section 1.2) to describe the physical functioning of the components and the control programs in a library.

As detailed in Section 1, we have achieved the possibility to create machine-understandable knowledge of the system holistically. For large systems, such knowledge is often complex and distributed, and Semantic Web technologies provide the ideal support for human and software agents to author, access, and reason about the knowledge in a hypermedia environment. However, what is missing is an *architectural paradigm* that brings together Web-based industrial systems and Semantic Web-based Knowledge Graphs (KG) to attain autonomous and adaptive automation of electro-mechanical systems.

In the next section, we will describe our ongoing work, which is yielding promising results in achieving autonomous and adaptive automation through hypermedia-based multi-agent systems. These systems also align well with developments in industrial automation and efforts towards creating Semantic Web-based integrated descriptions of the system.

3. Hypermedia Multi-agent Systems for Industrial Automation

We now outline our approach in modeling and constructing industrial automation systems as hypermedia multi-agent systems (HMAS). We first describe how the integrated systems knowledge introduced in Section 1 can be used to automatically synthesize the input knowledge needed to support the three modeling dimensions in multi-agent systems (MAS), namely the organization, the deployment of the agents, and the description of the environment (see Figure 3). Following this, we discuss our approach to making the deployment of MAS more distributed, in the sense that organizational management and the runtime representation of the environment appear as services and resources on the Web. Finally,

[3]https://openthread.io/
[4]https://ifttt.com/

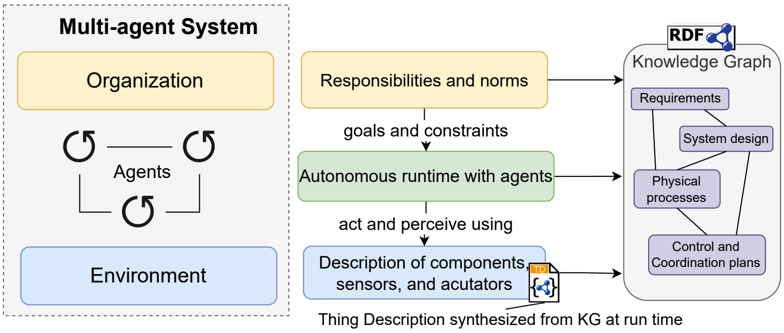

Figure 3: Knowledge Graphs are ideal for synthesizing and supporting multi-agent systems – a combination that enables the creation of autonomous industrial systems

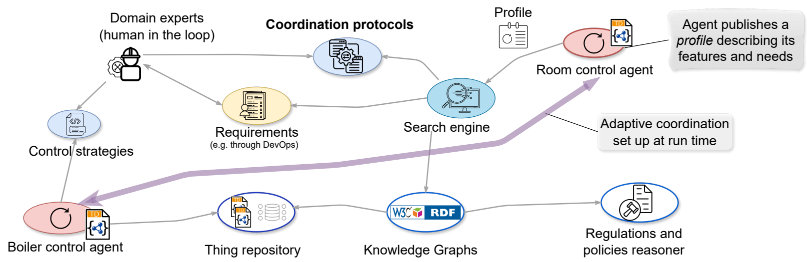

Figure 4: In a hypermedia-based automation system, agents can establish coordination at run time by publishing a profile which is then either evaluated and responded to or forwarded by other services and agents. For example, in this way the room control agent in the top right corner, when it determines that it has to coordinate with a central heating system, not only discovers its collaborating peer (the boiler control agent at bottom left), but also finds an appropriate coordination protocol (from a library) to use

we outline some of the key challenges in achieving a practical deployment of HMAS-based automation in an industrial setting.

Synthesizing Organization Specification In [14], we demonstrated how integrated knowledge about the system can be leveraged to synthesize the specification of the MAS organization automatically. The integration of requirements and system design descriptions played a key role in achieving automated inference about groups and roles within the organization. However, the approach required a centralized service for managing the organization, which incurs long communication paths in large networks. This is especially detrimental to energy-constrained devices like battery-powered wireless sensors and actuators. Therefore, in ongoing work, we are examining mechanisms to decentralize the organization's management infrastructure by partitioning the specification and hosting it on edge devices that are closest to the system context relevant to the specification fragment.

Supporting Agent Runtime In [3] we showed that the commonly known control strategies can be semantically described in a manner that their use for fulfilling a given combination of requirements and system can be automatically inferred. Consequently, when an agent is required to achieve a goal (formally described using the GORE ontology), it can select a suitable control strategy from a library of options. Because the agent also knows about the physical dependency between the system's components, it becomes aware of the potential coordination that may be needed. In [15] we showed how agents can publish their abilities and goals (including the need for coordination) in the form of agent profiles [16]. We are currently exploring ways in which such a profile can be disseminated in the network so that coordination can be established in a loosely coupled manner at run time (Figure 4) illustrates our approach through an example).

Hypermedia Environment as a Reflection of the Physical Environment In industrial systems, we deal with physical environments where the automation agents [5] are situated. The electro-mechanical components (including sensors and actuators) are artifacts that the automation agents need to use in order to fulfill the roles they have elected or been assigned to. In our current implementation, the artifacts register their Thing Descriptions [6](TD) with a centralized registry service (labeled as Thing repository in Figure 4). In a deployment where artifacts are directly accessible over the network, a central repository may seem unnecessary; however, such a repository helps reduce the time required to search and discover artifacts. Additionally, for energy-constrained devices, hosting TDs externally helps conserve energy. However, we aim to distribute such a TD repository throughout the network, contextualizing it to the electro-mechanical subsystems (not only to reduce network latencies but also to reduce the size of the in-memory triple store, thereby increasing query performance). We believe this approach also resonates well with the idea of agents using the hypermedia environment to discover peer agents and artifacts [17].

Challenges During our research, we have continuously validated our approach against real-life electro-mechanical systems and their simulations. One of our key findings was that the engineering of HMAS-based automation requires a significant shift in the workflow: hitherto, automation engineers had manually interpreted the system specification (from human-understandable sources only) and programmed the reactive procedures. In this process, incomplete facts were implicitly handled during the programming of the controls. However, our approach requires that the engineers shift their focus to ensure the correctness of the sources used for automatically synthesizing the KGs. Similarly, domain experts need to ensure that reusable program artifacts, such as standard control and coordination programs, are described in a manner that allows for the automated inference of compatibility with a system installation. However, currently, the tooling support for knowledge engineering (by those who are not Semantic Web practitioners) is sparse, and we need to address this. Nevertheless, in our interactions with automation engineers and domain experts, we received feedback that they see the value in this approach because the downstream workflows (e.g., selecting suitable control programs or adapting to changes) are not only automated, but the deployed system is also robust.

Another practical challenge is that real-life systems are multi-vendor installations, where an HMAS-based automation will require the design, commissioning, and maintenance of an almost parallel infrastructure. Therefore, until all major automation vendors are technologically aligned towards supporting hypermedia-based systems (e.g., delivering KGs of their systems and physical processes and providing semantic descriptions of the interaction interfaces), costly manual integration of legacy systems would be required. In other words, the *software and network engineering* of HMAS-based systems is an important issue to be addressed. However, as we witness a growing interest in building more open, flexible, and robust systems in industrial automation, we are hopeful that this challenge will be addressed in the industry.

4. Conclusion

In this article, we provide a brief overview of the state of the art in industrial automation, highlighting the potential benefits and applications of adopting HMAS-based automation systems. A key finding in our work so far has been that the progress in achieving machine-understandable representation of domain knowledge and the development of hardware and network technologies in industrial automation have not yet come together to benefit from the advantages of HMAS-based systems. Through active research and demonstrations of its results in the industry, we are optimistic that we can motivate researchers and practitioners to address the gaps.

[5]These are the autonomous artificial agents that are deployed to manage electro-mechanical systems, components, and functions

[6]https://www.w3.org/TR/wot-thing-description11/

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] G. Ramanathan, M. Husmann, Semantic description of equipment and its controls in building automation systems, in: European Semantic Web Conference, Springer, 2022.

[2] G. Ramanathan, S. Mayer, Reasoning about physical processes in buildings through component stereotypes, Data-centric Engineering (2024).

[3] G. Ramanathan, S. Mayer, A match made in semantics: Physics-infused digital twins for smart building automation, in: International Conference on Automation Science and Engineering, IEEE, 2024.

[4] A. Van Lamsweerde, Goal-oriented requirements engineering: A guided tour, in: Proceedings fifth ieee international symposium on requirements engineering, IEEE, 2001.

[5] A. K. Tyagi, T. F. Fernandez, S. Mishra, S. Kumari, Intelligent automation systems at the core of industry 4.0, in: International conference on intelligent systems design and applications, Springer, 2020, pp. 1–18.

[6] G. Ramanathan, M. Husmann, S. Mayer, Interoperability vs. tradition: benefits and challenges of web of things in building automation, in: Proceedings of the 11th International Conference on the Internet of Things, 2021, pp. 57–63.

[7] C. Dey, S. K. Sen, Industrial automation technologies, CRC Press, 2020.

[8] M. Wollschlaeger, T. Sauter, J. Jasperneite, The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0, IEEE industrial electronics magazine 11 (2017) 17–27.

[9] C. E. Pereira, C. Diedrich, P. Neumann, Communication protocols for automation, in: Springer Handbook of Automation, Springer, 2023, pp. 535–560.

[10] A. Jestratjew, A. Kwiecien, Performance of http protocol in networked control systems, IEEE Transactions on Industrial Informatics 9 (2012) 271–276.

[11] M. Attaran, The impact of 5g on the evolution of intelligent automation and industry digitization, Journal of ambient intelligence and humanized computing 14 (2023) 5977–5993.

[12] T. Samad, P. McLaughlin, J. Lu, System architecture for process automation: Review and trends, Journal of Process Control 17 (2007) 191–201.

[13] G. Ramanathan, M. Husmann, C. Niedermeier, N. Vicari, K. Garcia, S. Mayer, Assisting automated fault detection and diagnostics in building automation through semantic description of functions and process data, in: Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, 2021, pp. 228–229.

[14] G. Ramanathan, Synthesizing multi-agent system organization from engineering descriptions, in: International Workshop on Engineering Multi-Agent Systems, Springer, 2023, pp. 167–177.

[15] G. Ramanathan, S. Mayer, A. Ciortea, Towards hypermedia environments for adaptive coordination in industrial automation, in: International Conference on Emerging Technologies and Factory Automation, IEEE, 2024.

[16] D. Vachtsevanou, B. de Lima, A. Ciortea, J. F. Hübner, S. Mayer, J. Lemée, Enabling bdi agents to reason on a dynamic action repertoire in hypermedia environments, in: Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, 2024, pp. 1856–1864.

[17] M. Castellucci[1], S. Burattini[1], A. Ciortea, J. Lemée, D. Vachtsevanou, A. Ricci[1], S. Mayer, Towards agents' embodiment in hypermedia multi-agent systems, in: Multi-Agent Systems: 21st European Conference, EUMAS 2024, Dublin, Ireland, August 26–28, 2024, Proceedings, volume 15685, Springer Nature, 2025, p. 361.