Towards Achieving Adaptive Behaviour of Agents Through Physics-Infused Descriptions of Cyber-physical Devices

Ganesh Ramanathan[1,*,†], Simon Mayer[2,†]

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

Abstract

Cyber-physical devices (CPDs) are now widely employed in environments like smart buildings and factories. Many of these devices are involved in managing physical processes, such as the transformation of substances and energy. When it comes to achieving adaptive behavior through autonomous artificial agents that use such CPDs as artifacts in the pursuit of their goals, understanding (by the agent) both the technical interface of a CPD and its underlying physical processes becomes important. The W3C Web of Things Thing Description (TD) standard aims to address this challenge; however, hitherto, only the possibility of achieving technical interoperability has been explored, while the inclusion of the semantics of underlying physical processes remains to be explored. In this article, we describe our work in progress, which seeks to demonstrate how existing models that enable the description of physical processes can be used to embed such knowledge directly in the TDs. We further explore how this may enable artificial agents in a hypermedia environment to discover and utilize distributed CPDs without relying on a centralized, authoritative source of knowledge. We present intermediate results from our evaluation of an intelligent room automation system that demonstrates adaptive behavior through runtime discovery of CPD affordances.

Keywords

autonomous systems, web of things, hypermedia multi-agent systems, knowledge graphs

1. Why Machine-understandable Knowledge of Physics is Essential

Electro-mechanical devices often interact with a physical environment where they transform streams of substances, effort, and energy (the collective term being *physical stuff* [1]). Such a device may be a heater in a room that transfers thermal energy to the air to increase its temperature, or a light bulb that changes the illuminance of a surface through the emission of photons. We humans learn about the role and functioning of such devices through everyday experience, which is, for some, complemented with expert in-depth knowledge. We often use this information to adapt when the availability of devices that can act on our physical environment changes, or to select the optimal means when multiple options are available. For example, in a room with light bulbs and windows equipped with sun blinds, we can select which to use to illuminate the room because we are aware of the physical effects of the two devices. However, when it comes to artificial agents, research has been primarily focused on agents learning the physical effects (of operating a device) through sub-symbolic reasoning over observations of actions and their consequences [2]. Such approaches not only require large training datasets, but also often need manual engineering to ensure that the models do not inadvertently capture physically unrelated causes and effects. In addition, a model trained based on observations of a particular environment might become tightly coupled to this environment and may not easily adapt to unforeseen changes.

The motivation behind achieving autonomy through Knowledge Graphs (KG), on the other hand, is driven by the possibility of using formal reasoning on expert knowledge captured through ontologies [3]. To support this approach, W3C's Semantic Web standards, such as RDF, OWL, and SPARQL, allow authoring of ontologies and KGs and making them accessible to software agents. Consequently, over

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); simon.mayer@unisg.ch (S. Mayer)

the past decade, several ontologies in engineering based on OWL/RDFS have been published, including those for requirements engineering, system design, control programs, and physical processes [4]. In our previous work [5, 6], we demonstrated how these ontologies can be integrated to represent the holistic knowledge of the system, enabling artificial agents in a multi-agent system to automate the functioning of industrial electro-mechanical systems. In particular, we have shown how artificial agents can use both qualitative and quantitative ontology-based models of physical processes to reason about control and coordination strategies [7, 5]. For example, using such models, an agent responsible for controlling sun blinds would know that under some circumstances, the goal of illuminating the room can be achieved in a more energy-efficient manner by raising the blinds instead of switching on the lights.

However, the classical notion of KG-assisted autonomy in industrial systems considers a central source of system knowledge [4]. In large, heterogeneous, and dynamic deployments of CPDs, authoring and maintaining such centralized KGs is challenging because it demands manual engineering and maintenance as compared to distributed KGs that are provided by individual sub-systems and vendors. In [5], we demonstrated that augmenting TDs with knowledge about the physics underlying physical devices facilitates the automated engineering of control systems. By using TDs as a vehicle for conveying knowledge about the underlying physical processes in the CPDs, we achieved a way of making knowledge available to the agents in a distributed manner by embedding the KGs directly in the CPDs. In this article, we demonstrate how this approach can be extended to support agents in hypermedia-based multi-agent systems that need to achieve goals assigned to them at runtime. We demonstrate that agents can discover the CPDs, their local roles in the physical process, and the necessary global coordination.

We will use a running example of a room heating system that needs to work in coordination with a central boiler to illustrate the practical application of our approach. The deployment scenario we address for industrial systems involves individual agents being designated to one or more *functional roles* required in the physical system (e.g., a heating controller in a room). The agents need to discover and use the CPDs as artifacts in the (physical) environment. We evaluated our approach in a real-life setting of a smart room application. In this scenario, the presence of CPDs, their technical interfaces, and the description of the physical processes they are involved in are not known a priori to agents responsible for automating heating, ventilation, and lighting functions in the room. We found that in all cases, the agents adapted successfully to changes in requirements, regulations, and the availability of CPDs.

2. Existing Work on Modeling Physical Processes

Physical processes are composed of interlinked (physical) *mechanisms*. A mechanism—for example, *heat exchange*—is traditionally represented using mathematical models of its static and dynamic behavior. The resulting behavior of the process itself can be understood by iteratively solving the linked mechanisms. Although ontological models (like Bond graphs) can be used to represent the mathematical models [1], the solvers required—especially for dynamic systems—are non-trivial [8]. Therefore, in practice, mathematical models are encapsulated within a simulation program, which is then executed by the agents that need to understand how the process behaves. Such a simulation program can be packaged as a portable and interoperable *functional mockup unit* (FMU), the interfaces of which can be described using an ontology [9]. Often, a qualitative model of physics is sufficient to describe the working of a CPD. For example, such a model can be used to state that increasing the energy input to the heater would increase air temperature. In [6], we proposed an ontology that allows designers of CPDs to create *stereotypical* models of a CPD's physical behavior.

Recent efforts in the semantic modeling of physical processes within the Web of Things context have shown promising potential. Researchers have proposed extensions to TD models that incorporate physical semantics through specialized vocabularies (e.g., [10]). These vocabularies enable the description of physical properties, units, and relationships between physical quantities, making TDs

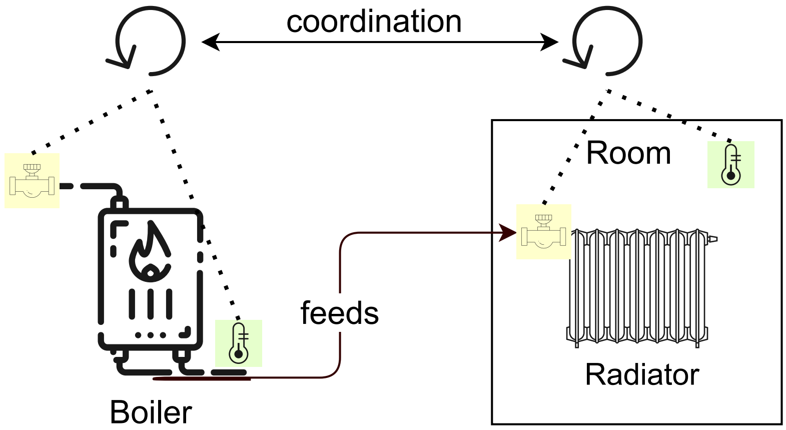

Figure 1: In the example scenario, the agents needs to coordinate to establish thermal energy flow

more expressive for representing physical processes. The SOSA/SSN ontology [1] has been adapted to describe sensors and their physical measurement contexts, while the QUDT ontology provides standardized representations for physical quantities and units [2]. These approaches aim to establish a shared vocabulary for well-known actions and perceptions in real-life physical systems; however, they require *syntactical* interpretation (of class names) by the agents' part. For example, the definition of the TD action affordance `iotschema:IncreaseTemperature` proposed in IoT Schema is devoid of a description of the underlying physics (i.e., the quantity and its trajectory) and the causation (e.g., the precondition towards exploiting the affordance is that there should be a supply of input energy). Therefore, compared to the state of the art, we argue that autonomous agents, especially in a hypermedia setting, would benefit from knowledge grounded in physics; this would avoid the need for tight coupling to specific vocabularies when reasoning about affordances provided by CPDs.

3. How we Integrated Physics into Thing Descriptions

Our previous work [6, 5, 7] enables designers of CPDs to describe the underlying physical processes. The openly-available *Elementary* ontology[3] integrates this past research.

We will use a running example of a building's heating system to demonstrate how concepts presented in *Elementary* can be applied to augment the interaction affordances in TDs with knowledge about the underlying physical processes. The scenario illustrated in Figure 1 shows two agents, one responsible for maintaining the thermal comfort in the room and the other responsible for managing the hot water supply. However, these agents are not a priori aware of the artifacts available in the environment; in our scenario, these are the room heat radiator, the central boiler, and the temperature sensors. We emphasize the absence of upfront engineering because it serves to highlight later that the agents can adapt when features or the availability of CPDs change at runtime (e.g., a new means of heating the room becomes unavailable, or a device is upgraded).

In this section, we first show how the TDs of the CPDs can be augmented with information about physical processes. In Section 4 we describe semantic rules that agents pursuing goals in hypermedia multi-agent systems can use on top of the *physics-infused TDs* to determine (1) the relevant interaction affordances of a TD to use and (2) the coordination required with agents managing dependent CPDs.

Listings 1 and 2 show the TDs of the two central artifacts involved in the scenario; namely, the room radiator and the boiler. The action affordance `fuel-valve-actuation` of the boiler has been

[1] https://www.w3.org/TR/vocab-ssn/
[2] https://www.qudt.org/
[3] http://w3id.org/elementary

augmented with two key facts: (1) the physical quantity that the action directly manipulates in the mechanism, i.e., the fuel flow rate, and (2) the expected consequence of the manipulation, i.e., the temperature of water supplied by the boiler. The TD of the radiator in the room similarly expresses the semantics of the heat exchange process, i.e., opening the valve will cause the air temperature to increase. In addition, to indicate a physical dependency, the TD of the radiator, through the property `fedBy`, points to the TD of the boiler.

Listing 1: Fragment of the TD of a boiler augmented with knowledge of physical process

```
{
"@type": "brick:Boiler", "hasStereotype": "hvac:FuelFiredBoiler",
"@id": "http://building-a/boiler-1",
"simulationModel":"http://server/boiler.fmu",

"actions": {
  "fuel-valve-actuation": {
    "actuator": {"@type": "brick:Valve"}, "relatedTo": {"@type":"hvac:Combustion"},
    "manipulates": {
       "atComponent": {"@type": "hvac:Burner"},
       "stuff":"brick:Fuel", "position": "inlet",
       "quantityKind":"qudt:VolumeFlowRate"
    },
    "affects": {
       "trajectory": "directly-porportional",
       "atComponent": {"@type":"hvac:BoilerTube"},
       "stuff":"brick:Water", "quantityKind": "qudt:Temperature", "position": "outlet",
       "observedBy": "http://building-a/boiler-1/outlet-temperature-sensor"
    },
    "forms": [{ "href": "/actions/fuel-valve-position" }]
}
```

Listing 2: Fragment of the TD of a radiator in the room

```
{
"@type": "brick:Radiator", "hasStereotype": "hvac:HeatExchanger",
"@id": "http://building-a/room-1/radiator-1",
"simulationModel":"http://server/radiator.fmu",
"brick:fedBy": "http://building-a/boiler-1",

"actions": {
  "increase-water-flow": {
    "actuator": {"@type": "brick:Valve"}, "relatedTo": {"@type":"hvac:HeatTransfer"},
    "manipulates": {
       "atComponent": {"@type": "hvac:Radiator"},
       "stuff":"brick:Water", "position": "elem:inlet",
       "quantityKind":"qudt:VolumeFlowRate"
    },
    "affects": {
       "trajectory": "directly-porportional",
       "atComponent": {"@type":"hvac:Room"},
       "stuff":"brick:Air", "quantityKind": "qudt:Temperature", "position": "elem:contained",
       "observedBy": "http://building-a/room-1/air-temperature-sensor",
       "precondition": {
          "stuff": "brick:Water", "position": "elem:inlet",
```

```
        "quantityKind":"qudt:Temperature", "minimumValue": 80}
    },
    "forms": [{ "href": "/actions/water-valve-position" }]
}
```

We highlight that where quantitative evaluation of physical processes (often for dynamic systems) is required, the agent has access to the simulation FMU by following the link pointed to by the `simulationModel` relationship. With the two example TDs as background, we now explain how agents can discover and use the CPDs to achieve the goals that are assigned to them.

4. How Agents can use the Physics-infused TDs

In [6] we showed that goals related to physical processes can be expressed in terms of achieving, maintaining, or avoiding a given value (or range of values) of a physical variable. For example, the goal to maintain the room's temperature can be expressed as:

```
:maintain-temperature a req:MaintenanceGoal;
 req:hasContext :room-1; req:maintain :air-temperature;

:air-temperature a phy:PhysicalVariable;
 phy:dealsWithStuff phy:air; phy:hasQuantityKind phy:temperature;
 elem:position elem:contained; phy:hasDesiredValue :setpoint;
 phy:allowedTolerance 5.
```

Given the above goal description, the agent queries a Thing Directory [4] (where CPDs register their TDs) to find TDs and their interaction affordances that may be useful in fulfilling the goal. The following query in SPARQL[5] shows how this can be matched (in a generic way):

```
SELECT ?comp ?rel ?dep WHERE{
    ?thing a td:ThingDescription.
    ?thing td:hasInteractionAffordance ?affordance.
    ?affordance elem:manipulates ?affectedVariable.
    ?affectedVariable elem:position ?position; elem:stuff ?stuff; elem:quantity ?q.
    :maintain-temperature req:addresses ?variable.
    ?variable elem:position ?position; elem:stuff ?stuff; elem:quantity ?q.
}
```

In the context of the example scenario, the above query returns the `increase-water-flow` action affordance of the radiator. The agent can then infer the protocol-binding-related information to invoke the action. In addition, from the semantic description, the agent also infers (through the `precondition`) that the manipulation is dependent on the temperature of the inflowing water; and from the dependency indicated by the `fedBy` relationship, the agent knows that it must coordinate with the boiler. In [7] we showed how agents can use hypermedia interactions to discover their collaborating peers and establish coordination. Therefore, if the physical-process-related precondition of a manipulation is not met, the agent infers the components on which the mechanism (i.e., the object of manipulation) is dependent, and attempts to establish collaboration with the agent that manages the corresponding component.

5. Evaluation

We evaluated our approach using a real-life setup of a small office room. The room was equipped with several devices that could fulfill a given function. Specifically:

1. Air heating through a hot-water radiator and an electric heater

[4] https://github.com/thingweb/thingweb-directory
[5] https://www.w3.org/TR/sparql11-query/

Change	Expected reaction	Fulfilled
Add energy-efficient lamp	Lighting control agent avoids use of ceiling lamps	Yes
Remove central heating	Heating control agent switches to using local electric heater	Yes
Add air quality constraint	Ventilation control agent is activated and uses the exhaust fan	Yes
Add air humidity constraint	Heating control agent has additional goal and uses the humidifier	Yes

Table 1
The results of our evaluation show that the automation agents could adapt to changes

2. Air humidification through a vapor humidifier
3. Lighting through ceiling and floor standing lamps
4. Ventilation through window-mounted exhaust fan

All devices were fitted with actuators and sensors that could be accessed over a wireless network. The TDs of the devices were stored in a Thing Repository service[6], which also provided an endpoint for executing SPARQL queries.

To test the adaptive behavior of our deployed system, we made the following synthetic changes during run time:

1. We removed access to the radiator and replaced it with the electric heater.
2. We added a floor-standing lamp in addition to the ceiling lamps; the standing lamp has a considerably lower power consumption.
3. We revised the requirements by adding and removing room humidity and air quality maintenance goals.
4. We simulated the failure of the central boiler.

We deployed three agents, responsible for heating (and humidification when required), lighting, and ventilation, respectively.

Table 1 shows the obtained results: our approach enabled the agents to react to changes and (re)discover CPDs that can be used to achieve their goals.

6. Discussion and Conclusions

We have highlighted that the already prevalent intuition about enabling autonomy through KGs can be further strongly argued for in industrial automation systems if we also explicitly include the knowledge about physical processes. Our work has shown that even a naive qualitative description of the underlying physics goes a long way towards achieving adaptive behavior. When only a high-level description of the physical process is available, it can be utilized to support data-oriented machine learning methods. This is particularly relevant for dynamic interlinked physical processes, where full-fledged qualitative or quantitative models are difficult to construct. The challenge posed by *linked dynamic systems* to purely data-oriented methods is illustrated by an extension that we formulated to the often demonstrated example of cart pole [11]: by linking two carts with a spring-damper connection, a Q-Learning-based implementation did not succeed to learn a policy even after more than 8000 episodes (for an isolated cart, on the other hand, the algorithm needs only about 400 episodes). However, by incorporating a basic and high-level physics model of the connecting spring, the two agents successfully balanced the poles after approximately 1200 episodes. Our insight is that when linked dynamic mechanisms are involved, the agents responsible for individual parts must coordinate. Knowledge of system construction and physical processes enables agents to recognize coordination needs. Therefore, quantitative and qualitative knowledge of physical processes will also benefit AI-assisted methods for automation.

We hope that our work shows the benefit of making low-level knowledge about physical processes directly accessible to agents in hypermedia multi-agent systems dealing with electromechanical installations, and this prompts further research collaboration between the engineering, Semantic Web, and multi-agent systems communities.

[6] https://github.com/thingweb/thingweb-directory

Declaration on Generative AI

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

Appendix

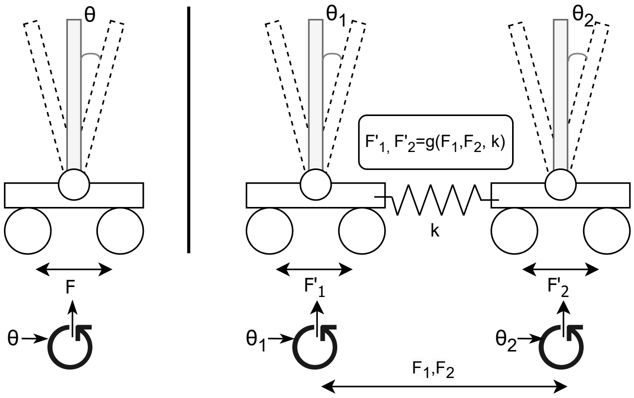

Figure 2: The cart pole problem is a classical example used for evaluating RL methods. But instead of a single cart (left), if two of them were to be coupled by a spring (right) then the individual agents learning to balance the poles would benefit from knowing the physical dependency - for example, they could exchange their individually planned action (F1 and F2) and recompute them considering the model of the spring

References

[1] P. Borst, J. Akkermans, A. Pos, J. Top, The physsys ontology for physical systems, in: Working Papers of the Ninth International Workshop on Qualitative Reasoning QR, volume 95, 1995, pp. 11–21.

[2] R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo, J. Barata, Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook, IEEE access 8 (2020) 220121–220139.

[3] J.-P. Calbimonte, et al., Autonomy in the age of knowledge graphs: vision and challenges, Transactions on Graph Data & Knowledge (2023).

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

[5] 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.

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

[7] 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.

[8] J. L. Top, H. Akkermans, Computational and physical causality., in: IJCAI, 1991, pp. 1171–1176.

[9] M. Mitterhofer, G. F. Schneider, S. Stratbücker, K. Sedlbauer, An fmi-enabled methodology for modular building performance simulation based on semantic web technologies, Building and Environment 125 (2017) 49–59.

[10] A. S. Thuluva, D. Anicic, S. Rudolph, Iot semantic interoperability with device description shapes,

in: The Semantic Web: ESWC 2018 Satellite Events: ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers 15, Springer, 2018, pp. 409–422.
[11] S. Nagendra, N. Podila, R. Ugarakhod, K. George, Comparison of reinforcement learning algorithms applied to the cart-pole problem, in: 2017 international conference on advances in computing, communications and informatics (ICACCI), IEEE, 2017, pp. 26–32.