Published: 19 November 2025

BUILDSYS '25: 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation
November 19 - 21, 2025
CO, Golden, USA

Conference Sponsors:
SIGENERGY

SHORT-PAPER

I Can't Read All That! Improving the Usability of Semantic Models Using Concise, Ontology-Agnostic, Building-Specific Schemas

LAZLO PAUL, Lawrence Berkeley National Laboratory, Berkeley, CA, United States

OZAN BARIS MULAYIM, Carnegie Mellon University, Pittsburgh, PA, United States

UMUT METE SAKA, Colorado School of Mines, Golden, CO, United States

ANAND KRISHNAN PRAKASH, Lawrence Berkeley National Laboratory, Berkeley, CA, United States

GABRIEL TOMAS FIERRO, Colorado School of Mines, Golden, CO, United States

MARCO PRITONI, Lawrence Berkeley National Laboratory, Berkeley, CA, United States

Open Access Support provided by:

Lawrence Berkeley National Laboratory

Colorado School of Mines

Carnegie Mellon University

BUILDSYS '25: Proceedings of the 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (November 2025)
https://doi.org/10.1145/3736425.3770107
ISBN: 9798400719455

I Can't Read All That! Improving the Usability of Semantic Models Using Concise, Ontology-Agnostic, Building-Specific Schemas

Lazlo Paul*
LPaul@lbl.gov
Lawrence Berkeley National Laboratory
Berkeley, CA, United States

Ozan Baris Mulayim
omulayim@andrew.cmu.edu
Carnegie Mellon University / Lawrence Berkeley National Laboratory
Pittsburgh, PA, United States

Marco Pritoni
mpritoni@lbl.gov
Lawrence Berkeley National Laboratory
Berkeley, CA, United States

Anand Krishnan Prakash
akprakash@lbl.gov
Lawrence Berkeley National Laboratory / Carnegie Mellon University
Berkeley, CA, United States

Mete Saka
saka@mines.edu
Colorado School of Mines
Golden, CO, United States

Gabe Fierro
gtfierro@mines.edu
Colorado School of Mines
Golden, CO, United States

ABSTRACT

Semantic ontologies have enabled the creation of formalized, machine-readable descriptions of heterogenous building systems by providing dictionaries of well defined concepts that can be applied to model them. Within a semantic model of a particular building, a subset of an ontology's concepts may be applied in different ways to represent a particular perspective of the building's systems. How the concepts were applied can only be understood by examining the large amount of instance data within a semantic model, which leads to usability challenges.

We propose a concise, ontology-agnostic method for defining building-specific schema (b-schema) graphs that summarize the structure and content of a semantic model. This approach provides a queryable and concise representation of the model's contents, separate from the instance data within a model, that can mitigate the challenges posed by the size and complexity of semantic models in processes such as visualization, querying, validation, and the use of large language models (LLMs).

We validate our approach on semantic models based on the Brick and ASHRAE S223 ontologies. Results demonstrate that b-schemas significantly reduce the complexity of visual interpretation, accelerate SPARQL queries and SHACL validation, and improve LLM-based knowledge graph question answering.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Information systems → Ontologies**.

KEYWORDS

Semantic Ontology, Large Language Models, Knowledge Graphs

BUILDSYS '25, November 19–21, 2025, Golden, CO, USA

ACM ISBN 979-8-4007-1945-5/25/11.
https://doi.org/10.1145/3736425.3770107

ACM Reference Format:
Lazlo Paul, Ozan Baris Mulayim, Marco Pritoni, Anand Krishnan Prakash, Mete Saka, and Gabe Fierro. 2025. I Can't Read All That! Improving the Usability of Semantic Models Using Concise, Ontology-Agnostic, Building-Specific Schemas. In *The 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BUILDSYS '25), November 19–21, 2025, Golden, CO, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3736425.3770107

1 INTRODUCTION

Resource Description Framework (RDF) ontologies such as Brick [2] and ASHRAE 223 [1] provide a standardized vocabulary of ontology concepts. These ontologies are designed to accurately represent building systems, and thus they must represent the heterogeneity found across different buildings, including variations in sensor placement, system configurations, and equipment layouts in semantic graph models [8]. These models represent the individual systems, equipment, and data points in a building, and can thus be very large.

Correctness of these models to a given set of constraints can be validated using SHACL [6]. Both the Brick and ASHRAE 223P ontologies use SHACL to find basic errors in semantic models according to validation rules within the ontologies to ensure that these diverse models can be used reliably in data-driven building applications. These applications retrieve specific data from the semantic model by issuing queries that rely on knowledge of the model's 'schema'—that is, the types of entities it contains and the relationships between them. Often, models based on the same ontology can have significant differences due to heterogeneity of buildings and subjective modeling choices made by the authors [3]. As a result, queries for semantic models are authored through manual exploration and iterative querying of the model. However, the large size, heterogeneity, and complexity of semantic models makes them difficult to interpret [13].

Essential operations like querying, validation, inference, and visualization should run at interactive speeds. These tasks can take minutes or hours on large graphs (hundreds of thousands of triples). It is typical for ontologies like ASHRAE S223 to produce large graphs even for small buildings. We note two important properties of these graphs: first, graphs only use a subset of an ontology's concepts; and second, graphs contain repetitive structures despite subjective modeling choices. These present an opportunity to summarize large models to reduce their size and complexity.

1.1 Background

Approaches for RDF graph summarization can reduce large and heterogeneous graphs into more interpretable forms [12]. Some approaches focus on identifying important graph patterns based on topological similarities, connectivity, or frequency [4, 10]. However, summarization approaches using graph structure are not effective on building semantic models, where essential information is represented in labels within the graph, such as class names. Other approaches leverage these labels, such as node-type aggregation methods [5], to provide useful coarse-grained overviews. However, these approaches hide essential structure, like the different ways a given ontology class is used within the model. For example, such methods would show that AHUs can connect to VAVs, and VAVs can connect to heating coils— yet would fail to differentiate two types of VAV: those that connect to coils and those that do not. The flaws of these methods become more critical when applied with ontologies that may use a class in multiple distinct ways. Representation of such distinctions increases the summary size and is therefore often neglected in the literature[13], yet it is essential for most applications of semantic models in buildings. Generic existing methods for graph summarization are thus insufficient, and new summarization approaches are needed.

Other recent approaches to avoid direct interaction with instance data in semantic models leverage Large Language Models (LLMs). Zero-and few-shot Knowledge Graph Question Answering (KGQA) tools provide a potential path to enable query of building models without requiring in-depth understanding of the semantic models or a wealth of building and ontology specific training data. Semantic parsing-based methods for KGQA show particular promise for retrieving complex data from semantic models by generating SPARQL queries from natural language [7]. Such methods have different strategies for managing the large size of semantic models and for providing relevant context to LLMs. The key challenge for all of these approaches is the identification of information that is relevant and information that is superfluous for query generation, as superfluous information can significantly degrade LLM performance [9]. While promising, these methods struggle significantly with translating natural language requests into semantic formalisms due to the vagueness of natural language [7]. For example, a user may request data about the cooling only VAVs present in their semantic model. However, which VAVs are "cooling only" can only be learned and queried for with the context of how reheat VAVs have been modeled, as containing a heating coil. The context required from a semantic model for an LLM to discover this may be difficult to determine or cover the whole contents of a model.

1.2 Our Approach

This paper proposes the concept of a b-schema, which leverages repeating structures within building semantic models to create a concise, ontology-agnostic representations of their contents. The b-schema maintains the graph structure of the original semantic model as well as links between it and the semantic model, so that it can be used to improve human interpretation of particular semantic models and the performance of processes such as querying and validation by reducing the amount of instance data that must be interpreted. Because b-schemas maintain the original RDF structure of the semantic model, while eliminating repeated instance information, we propose that the b-schema is also a superior representation for interpretation of semantic models by LLMs, which have rapidly become an important part of modern software tools.

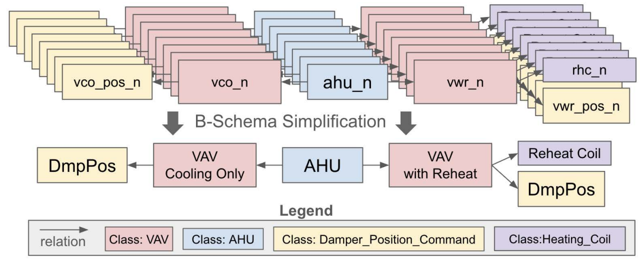

Figure 1: Conceptual diagram of b-schema

We demonstrate the b-schema through a case study involving two example models based on <Building Name/Location>(removed for anonymity). The example building is modeled using two ontologies, Brick and ASHRAE S223, as a 4-floor office building, utilizing one Air Handling Unit (AHU) per floor. Each AHU feeds 15 cooling only VAV terminal units for core zones and 15 VAV terminal units with reheat coils for perimeter zones. Figure 1 provides a conceptual representation of the Brick semantic model and b-schema representations of this building.

2 THE B-SCHEMA

The b-schema is a summarized RDF graph that maintains the structure (i.e. the class identities and relation topologies) of nodes within a semantic model, while eliminating repititious instance information. For semantic models using the b-schema, there are three separate graphs. The first graph is the semantic model (G), which contains the instance data, the second graph is the b-schema (H), which summarizes the contents in the semantic model, and the third is the member graph (M) which links nodes in the b-schema to the nodes in the semantic model they are isomorphic to.

Let G and H be valid RDF graphs. There exists a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ of subgraphs of G such that each S_i is isomorphic to H, given the following restrictions: (1) edges must have the same label, (2) nodes are isomorphic only if they are not classes, and they have the same `rdf:type`. The union of the subgraphs in $\mathcal{S}$ equals G; that is, $G = \bigcup_{i=1}^{n} S_i$. Overlap is allowed, i.e., S_i and S_j may share triples for $i \neq j$. For any valid SPARQL query Q that does not filter on node identity or cardinality, if Q returns results on G it will also on H. The graph H is *minimal* with respect to these conditions; that is, there does not exist a subgraph $H' \subset H$ such that H' also satisfies all the conditions above.

For each isomorphism between H and a subgraph S_i of G, and for each node v in H, a new edge is established between v and its corresponding node in S_i in G using the `rdfs:member` relation.

These are added to a third graph, M, and provide an interpretable link between H and G.

An illustrative semantic model and b-schema are shown in figure 1. For example, all nodes in the semantic model representing position commands for cooling only VAVs are represented by a node in the b-schema, specifically, the `DmpPos` node at the left-most side of the figure. All position command nodes of VAVs with reheat would be represented by a different node in the b-schema, specifically, the `DmpPos` node at the right-most side of the figure. In the model, cooling only VAVs and VAVs with reheat are distinct because VAVs with reheat have an additional relation to a node with the class `brick:Heating_Coil`.

Nodes in the b-schema reference nodes in the semantic models so that a user may leverage it to improve the performance of querying workloads on the semantic model and so that a user may know what in their graph is represented by the given b-schema, as graphs can be stored together arbitrarily.

The b-schema for a graph can be produced automatically by creating a SPARQL query based on the graph and transforming the results [1]. However, such a query is inefficient. The schema may be produced more easily when the model is created, and it could be provided to users with the model. Future work may investigate more efficient methods of creating this representation.

3 CASE STUDY

The b-schema is evaluated using two semantic models defined using Brick and ASHRAE S223. These ontologies were chosen due to their relevance to the building domain and because of their different methodologies for representing building information, demonstrating generalizability of the b-schema approach.

The Brick ontology represents equipment and their properties (i.e. "points") primarily using a taxonomy of classes. On the other hand, the ASHRAE S223 ontology is designed to represent equipment and their properties using a composition of ontology concepts in the graph. It includes a few basic classes for equipment, their properties, and concepts such as connection points to model connectivity. It allows users to annotate properties with additional details, such as their quantity kind and unit of measure. For this reason, ASHRAE 223P models are larger, as they represent more information in the graph structure, while Brick represents more information in class names.

We propose that a b-schema can be used to address challenges caused by the large size of semantic models. Particularly impacted processes include human interpretation of semantic models, query and validation execution times, and interpretation of semantic models by LLMs.

4 RESULTS

In this section, we evaluate the b-schema in four common tasks – model visualization, validation, query, and LLM-driven KGQA. We find that the b-schema can be used to improve performance in each of these tasks, improving interpretability, execution speed of queries and validation, and the accuracy of KGQA while reducing the amount of input tokens.

[1]linked github repo removed for anonymity

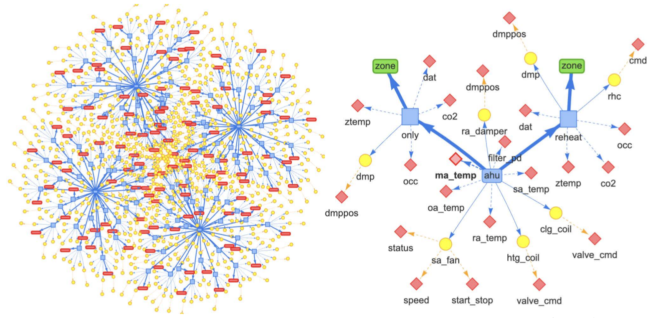

Figure 2: Node-edge visualization of brick model (left) and its b-schema (right).

```
1 SELECT ?dat ?VAV ?ahu WHERE {
2     ?dat a brick:Discharge_Air_Temperature_Sensor .
3     ?VAV a brick:VAV .
4     ?VAV brick:hasPoint ?dat .
5     ?ahu brick:feeds ?VAV
6     FILTER NOT EXISTS {
7        ?VAV brick:hasPart ?rc .
8        ?rc a brick:Heating_Coil . } }
```

```
1 SELECT ?dat ?VAV ?ahu WHERE {
2     ?VAV a brick:VAV .
3     ?VAV_B rdfs:member ?VAV .
4     ?ahu brick:feeds ?VAV .
5     ?VAV brick:hasPoint ?dat .
6     ?dat a brick:Discharge_Air_Temperature_Sensor .
7     FILTER NOT EXISTS {
8         ?VAV_B brick:hasPart ?rc .
9         ?rc a brick:Heating_Coil . } }
```

Figure 3: SPARQL queries for Brick model using only the semantic model (top) and leveraging the b-schema with the semantic model (bottom).[2]

Improvement to Visual Interpretability: For a user to understand the contents of their model, they only have to use the b-schema (H). Our example brick model of a medium building was 3500 triples long and our ASHRAE S223 model was 6900. The b-schema representing the content of these semantic models reduced their size by approximately 94%. The interpretability improvement applies even when the semantic models are viewed as a text file, but is most easily shown by visualizing the models, as in figure 2. While the labels in the figure may not be easy to understand without familiarity with the naming convention, the structure and content of the model is readily apparent.

Reduced SHACL and SPARQL Query Execution Time: The b-schema can be used in concert with the original semantic model to improve the performance of querying and validation workflows. Figure 3 (top) shows a typical query a user may run against a Brick model. It retrieves instance information for AHUs, the cooling only VAVs fed by the AHUs, and the discharge air temperatures of the VAVs. The query shown in figure 3 (bottom) retrieves the relevant topological information from the b-schema and the instance information from the semantic model, using the rdfs:member relation shown on line 3 to link between the two. Querying topological information from the b-schema rather than the individual instances significantly reduces the amount of nodes that have to be evaluated, and allows queries to be executed more quickly. Additionally, because the b-schema maintains the original structure of the semantic model, the queries are nearly identical.

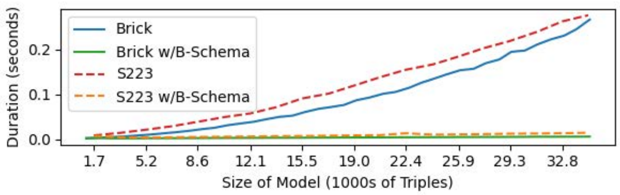

Figure 4: Query times as model size increases.

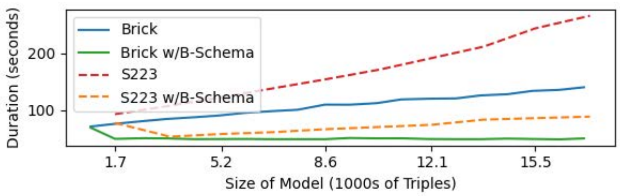

Figure 5: SHACL Validation times as model size increases.

Queries were tested against building models using Oxigraph [3]. For these queries, the semantic models, b-schemas, and member graphs (G, H, and M) are parsed together, then queries are run on the composition of these graphs. Results are shown in Figure 4. The smallest models included only one instance of AHU and VAV, and queries thus take approximately the same time with and without the b-schema. As the size of the model grows, including multiple AHUs and VAVs as our example building does, query time increases if the b-schema is not used, as more of the graph must be traversed to identify the queried for equipment. For this simple example, using a state of the art query processor and a powerful computer, results are returned quickly for all model sizes. However, building applications commonly use multiple queries, more complex queries, and may be run on resource constrained devices, leading to performance challenges.

Validation workloads can also benefit from the use of the b-schema to reduce the amount of nodes for evaluation. Validation was run using the topbraid SHACL validator via BuildingMOTIF [4]. Validation was stopped when it exceeded 5 minutes, which occurred for ASHRAE S223 models exceeding 16,000 triples. Figure 5 shows improved validation runtimes for the b-schema (H) as opposed to the semantic models (G).

Improved LLM Question Answering Performance: Table 1 compares use of the b-schema (H) for KGQA with two existing methods DA-KGQA (DA) [7] and ReAct (RA) [11]. DA attempts to extract a relevant subgraph from an ontology and building semantic model to provide as context for query generation. RA is a prominent agentic approach that enables iterative self-correction for LLMs. Our implementation of RA uses a two-agent system including query writing and critiquing agents. In each iteration, the writer generates a SPARQL query and executes it to retrieve information from the graph, and the critiquer observes the result and provides feedback or a decision *final*. This step of query generation and execution is repeated for three iterations or if the critiquer returns *final*. Two variations of this method are used, RA100, which provides the first 100 triples of the graph as context to begin query writing, and RA5000, which provides the first 5000 triples. The table shows if a correct query was generated for the questions and the amount of input tokens that were used. DA was run only for the Brick model, as that is the ontology it was designed for, while the other methods were run for both S223 and Brick.[5]

While the RA and DA methods attempt to extract relevant subgraphs from the semantic models to use as context, the b-schema improves LLM KGQA performance by providing a more complete summary of the semantic model, while limiting superfluous information for query writing. Particularly, the b-schema allows an LLM to better address vague questions, or questions that don't directly reflect the content of the semantic model, by providing information that other methods wouldn't identify as relevant.

For question 1, all baseline methods were successful on the Brick model because this question mirrors the triple structure and labels present in the brick model (e.g. Discharge_Air_Temperature_Sensor, VAV, AHU). This language does not mirror the labels present in the S223 Model, potentially leading to query failure. For question 2, RA5000 successfully writes a query for S223, because the language in the question does mirror what is present in the model, and the initial graph context is large enough to retrieve the relevant information.

For question 3, cooling only VAVs are indirectly defined in the semantic models as VAVs that *do not* contain heating coils. The baseline methods likely failed to answer this question because they are unable to determine the relevant subgraphs showing how the cooling only VAVs were modeled in comparison to the VAVs with reheat. The b-schema method for S223 also failed. VAVs are defined indirectly in S223, potentially contributing to greater lexical misalignment of what the query asks for and what the model represents. Creation of a correct query would require mapping the term VAV to the S223 class `SingleDuctTerminal` and identifying that a VAV with reheat as opposed to cooling only is represented by the terminal connecting to a node with the class `Coil`, which is then connected to the label `Role-Heating`.

Question 4 is the most vague, and depends on domain knowledge of the specific HVAC system. This question represents a cold call, which is one of the most common issues a facility manager may use a semantic model to respond to. The HVAC system represented in these semantic models is a single-duct multi-zone VAV system. Domain experts know that the most likely cause of cold calls for such systems are improper zone air balances, caused by faults of the AHU supply fan and VAV Dampers, or faults with the reheat coils. A correct KGQA query should return these components. This question provides no information about the systems present in the semantic model. While LLMs have the capability to identify the equipment most likely causing the cold call, this depends on complete information of the building system in question. Only the b-schema methods are successful, because they give the most clear and concise picture of the building.

We also see that the b-schema methods use significantly less input tokens than the baseline methods. Not only do the baseline methods select context with potentially repetitive information, but they load the semantic models with the ontologies to provide definition of the concepts in the models. The b-schema methods, by comparison, use the b-schema as the only input, and do not include

[2]Prefix declarations excluded to reduce page length. Standard prefixes are used, viewable at (linked github repo removed for anonymity)
[3]https://github.com/oxigraph/oxigraph
[4]https://github.com/NREL/BuildingMOTIF

[5]all methods used OpenAI GPT 4.1.

Table 1: Comparison of LLM-Driven Methods for HVAC Questions (Success and Token Count)

Question	RA100-b	RA5000-b	DA-b	RA100-s	RA5000-s	BS-b	BS-s
1. What are the discharge air temperature sensors for all VAVs and their associated AHUs?	✓ (3889)	✓(119122)	✓(8825)	×(11415)	×(322690)	✓(1608)	✓(1601)
2. What are all the components of each air handling unit?	✓ (8757)	✓(178369)	×(7396)	×(12823)	✓(107891)	✓(1599)	✓(2313)
3. What are all the instances of valve commands for cooling only VAVs?	× (7149)	×(177881)	×(9388)	×(10853)	×(322115)	✓(1604)	×(2318)
4. I'm too cold, what equipment is most likely at fault	× (8921)	×(178917)	×(6027)	×(12830)	×(325241)	✓(1607)	✓(2321)

the ontology. The b-schema methods thus rely on LLMs understanding elements of the ontology without being provided their definitions, so the ontology must use well-named concepts that can be mapped to the input questions. Ideally the LLM would also be trained on the ontology, and the semantic model would be validated against the ontology, to ensure it is well formatted and applies the concepts correctly.

5 CONCLUSION AND FUTURE RESEARCH

This paper proposes an ontology-agnostic abstraction of semantic models, the b-schema, that can improve their user-friendliness in the building domain. By abstracting repeating patterns, a b-schema can improve visual interpretability, accelerate querying and validation, and enhance the performance of LLMs in tasks such as KGQA.

Future research should focus on developing more efficient and scalable methods for generating b-schemas, potentially by integrating b-schema creation into the model authoring process or by developing more efficient algorithms for their automatic construction from existing semantic models. Semantic models of real buildings can include data for multiple domains (e.g. mechanical, electrical) that is interconnected while having different patterns of repetition. Further work is needed to refine strategies for partitioning of linked data graphs that cross multiple domains of interest into focused semantic models, allowing users to create the most effective b-schemas for their specific needs. Semantic models can also have detailed information distinguishing near duplicate structures, that may be erroneous or relevant for a few use cases but not others. Methods to allow users to easily filter unneccesary information from their models can reduce inconsistencies, supporting summarization and use with LLMs via b-schema. Such approaches to normalize semantic models will support the robustness of the b-schema approach in real buildings.

The effects of b-schema on tasks such as validation, query performance, and LLM-driven workflows should also be more systematically evaluated in real-world environments. The b-schema may provide a more effective abstraction of a semantic model for use with LLMs, and subsequent evaluation should go beyond just KGQA.

ACKNOWLEDGMENTS

This research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies of the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231, and by the California Energy Commission under EPC-19-301 and EPC-19-309. Additional support was provided by the National Alliance for Water Innovation (NAWI), funded by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE), Industry Technology Office, under DE-FOA-0001905. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

REFERENCES

[1] ASHRAE. 2025. ASHRAE Standard 223 User Documentation. https://docs.open223.info/intro.html. Accessed: 2025-2-5.

[2] Bharathan Balaji, Arka Bhattacharya, Gabriel Fierro, Jingkun Gao, Joshua Gluck, Dezhi Hong, Aslak Johansen, Jason Koh, Joern Ploennigs, Yuvraj Agarwal, Mario Berges, David Culler, Rajesh Gupta, Mikkel Baun Kjærgaard, Mani Srivastava, and Kamin Whitehouse. 2016. Brick: Towards a Unified Metadata Schema For Buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16)*. Association for Computing Machinery, New York, NY, USA, 41–50.

[3] Imane Lahmam Bennani, Anand Krishnan Prakash, Marina Zafiris, Lazlo Paul, Carlos Duarte Roa, Paul Raftery, Marco Pritoni, and Gabe Fierro. 2021. Query relaxation for portable brick-based applications. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. ACM, New York, NY, USA.

[4] François Goasdoué, Paweł Guzewicz, and Ioana Manolescu. 2020. RDF graph summarization for first-sight structure discovery. *VLDB J.* 29, 5 (Sept. 2020), 1191–1218.

[5] Jimao Guo and Yi Wang. 2022. RDF graph summarization based on node characteristic and centrality. *J. Web Eng.* 21, 7 (Dec. 2022), 2073–2093.

[6] Holger Knublauch and Dimitris Kontokostas. 2017. Shapes Constraint Language (SHACL). https://www.w3.org/TR/shacl/. Accessed: 2025-7-16.

[7] Ozan Baris Mulayim, Gabe Fierro, Mario Bergés, and Marco Pritoni. 2025. Towards zero-shot question answering in CPS-IoT: Large language models and knowledge graphs. In *Proceedings of the 2nd International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things*. ACM, New York, NY, USA, 7–12.

[8] Marco Pritoni, Drew Paine, Gabriel Fierro, Cory Mosiman, Michael Poplawski, Avijit Saha, Joel Bender, and Jessica Granderson. 2021. Metadata Schemas and Ontologies for Building Energy Applications: A Critical Review and Use Case Analysis. *Energies* 14, 7 (2021). https://doi.org/10.3390/en14072024

[9] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by Irrelevant Context. *arXiv [cs.CL]* (Jan. 2023).

[10] Georgia Eirini Trouli, Alexandros Pappas, Georgia Troullinou, Lefteris Koumakis, Nikos Papadakis, and Haridimos Kondylakis. 2022. SumMER: Structural summarization for RDF/S KGs. *Algorithms* 16, 1 (Dec. 2022), 18.

[11] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. http://arxiv.org/abs/2210.03629 arXiv:2210.03629 [cs].

[12] Mussab Zneika, Dan Vodislav, and Dimitris Kotzinos. 2019. Quality metrics for RDF graph summarization. *Semant. Web* 10, 3 (April 2019), 555–584.

[13] Šejla Čebirić, François Goasdoué, Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu, Georgia Troullinou, and Mussab Zneika. 2019. Summarizing semantic graphs: a survey. *The VLDB Journal* 28, 3 (June 2019), 295–327. https://doi.org/10.1007/s00778-018-0528-3