

Templates Constrain, Agents Interpret: Early Evidence for Validated Brick Knowledge-Graph Construction from Industrial Building Data

Knut Nordanger^{1,*}, Luis Caetano¹, John Clauß¹, Reidar Kind¹ and Thomas Elvrum Lassen¹

¹SINTEF Community (SINTEF AS), Trondheim, Norway

Abstract

Knowledge graphs for the built environment fail less for want of awareness than for want of construction pipelines that are at once structurally correct and able to scale. We argue for a division of labour: language-model agents read source records and propose classifications, ontology checks decide whether a proposal is admissible, and templates are the only mechanism that writes triples, validated before anything is committed. We report early evidence from one room-automation system in one building: 785 substantive records across 32 rooms. Records are classified from their own row evidence, and the classifier can propose only admissible Brick point classes by construction: a guarantee of admissibility, not of correctness. The templates expand a graph of 4,998 triples that conforms to the project's shape library, and controlled faults, which also guided that library, show that structural conformance does not establish semantic truth. The prototype is incomplete: proposed classes do not yet parameterise the templates, and equipment identity still reads the standardised component identifier. The two reference artefacts, the classification table and the room fixture, are same-team work, not independent ground truth, and other buildings and input formats are unevaluated.

Keywords

knowledge graph construction, neuro-symbolic pipelines, ontology templates, SHACL, Brick, building automation

1. Introduction

The obstacle to semantic models of the built environment is no longer awareness. Building owners know that portable analytics, fault detection and control all depend on a machine-readable description of what a building contains, and ontologies such as Brick [1] and RealEstateCore [2] exist to provide it. The obstacle is *construction*: turning the raw, cryptic export of a building-management system into a correct knowledge graph, reliably and at scale, is still largely unsolved.

Three approaches are on offer, and each fails in a characteristic way. A pure language-model approach generates the RDF and checks it only afterwards, so structure is produced by the model rather than guaranteed [3]. A rule-based approach maps source names deterministically but breaks down where its rules do not fit [4]. Manual modelling uses expert judgement but does not scale: a single building runs many systems, and a public estate runs hundreds of buildings. Retrieval-based classifiers have lately made point-level typing much stronger [5], but a class attached to a point is not a graph: whatever holds the points together is still produced by one of the three.

The scaling question is not hypothetical. In Norway, the public owner Statsbygg mandates an equipment-tagging standard, the Tverrfaglig Merkesystem (TFM) [6], in everything it builds, so much structure is present in the data. The open question is what an automated construction pipeline must look like to exploit it safely: fast where labels are clean, careful where they are not, and never letting a model's mistake become a malformed graph.

SKGi 2026: Scaling Knowledge Graphs for Industry Workshop, co-located with SEMANTiCS'26: 22nd International Conference on Semantic Systems, September 15–17, 2026, Ghent, Belgium

*Corresponding author.

✉ knut.nordanger@sintef.no (K. Nordanger); luis.caetano@sintef.no (L. Caetano); john.clauss@sintef.no (J. Clauß); reidar.kind@sintef.no (R. Kind); thomas.lassen@sintef.no (T. E. Lassen)

🆔 0000-0003-3425-0115 (K. Nordanger); 0009-0009-0572-7501 (L. Caetano); 0000-0003-4593-6745 (J. Clauß); 0009-0008-1948-7767 (R. Kind); 0009-0001-5758-7936 (T. E. Lassen)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Our position is that the two requirements, flexibility and correctness, should be met by different mechanisms. Templates should fix the structure of the graph as an invariant, and language-model agents should do only the row-level interpretation that genuine ambiguity demands. Neural components propose; symbolic constraints decide. Standardised labels, where present, are evidence that interpretation may draw on, not a foundation it rests upon. We instantiate this in a pipeline that builds Brick and RealEstateCore graphs from TFM-coded BACnet exports.

The paper makes three contributions, of three different kinds. The first is a *position*: where the boundary between neural and symbolic work should fall in knowledge-graph construction for the built environment. The second is a *prototype* that instantiates it and is incomplete: classification from row evidence, ontology admissibility checking and template-only emission are implemented and exercised, while proposed classes do not yet parameterise the templates and equipment identity still reads the standardised component identifier. The third is *early evidence* from one room-automation system of one building. As a short position paper, its aim is to argue the boundary and show that it is buildable, not to report a finished system.

2. Background and Related Work

We target two ontologies. Brick [1] models the equipment, points and relationships inside a building; RealEstateCore (REC) [2] supplies the spatial model into which that equipment is placed. Together they give a portable description of a building’s automation system.

The symbolic mechanism is OTTR, the Reasonable Ontology Templates framework [7]: named, parameterised macros whose expansion produces ontology instances, so a modelling decision is written once and reused wherever it applies. We expand them with the maplib engine [8]; raw template and mapping languages being demanding for practitioners, higher-level interfaces over them have recently been proposed [9]. The pattern has industrial precedent: Lupp et al. [10] use layered OTTR template libraries to keep asset-maintenance ontologies scalable by separating input structure from target semantics, the separation we rely on. That such modularity is central to LLM-era ontology engineering is Shimizu and Hitzler’s [11] argument; binding a model to template parameters, so that it fills well-defined slots rather than emitting free-form output, is our instance of it.

Two recent systems bracket ours. BrickLLM [3] generates Brick RDF directly with an LLM and enforces compliance by validating against Brick’s SHACL shapes and regenerating the whole graph on failure, a design whose semantic correctness the authors report declining as graphs grow more complex. The Semantic-IoT framework of Du et al. [4] generates Brick graphs deterministically with RML mappings, matching terminology by Levenshtein similarity; it is reliable where its rules fit, but its matching accuracy stays under 60% and offers no interpretive fallback for the rest (though it does test reuse across synthetic hotels of up to 1,000 rooms, a broader scaling test than ours). We sit between them: a language model supplies interpretation, templates fix structure by construction and emit all RDF, and validation withdraws implicated bindings or escalates rather than regenerating the graph.

The row-level classification we delegate to the classifier is itself an active problem: Brick-DICL [5] maps building-management points to Brick classes with retrieval-augmented in-context learning, and related work applies retrieval to the same step [12]. Point2Brick [13] pushes that line furthest, reaching 78.4% exact match with a retrieval-grounded agent pipeline, and its baselines are the external case for our admissibility constraint: free-form prompting proposed non-existent Brick classes for 38.9% of points and 12.6% even with the full class list, falling to zero only when the answer was restricted to retrieved candidates. That validation checks whether each point’s class exists, not whether the assembled graph holds together. Closer to this workshop, Zhang et al. [14] aim to cut manual KG-construction effort with a rule-plus-language-model pipeline, and Mehmandarov and Waaler [15] formalise semantic identifier resolution to ground LLM workflows. Both share our aim of trustworthy construction at scale, but neither makes graph structure an invariant.

Our contribution is placement, not a new algorithm: the neural-proposer / symbolic-constructor composition instantiates a catalogued neuro-symbolic design pattern [16], and what differs is where

the boundary between them is drawn. Making OTTR templates the structural invariant, so the model binds parameters and never emits RDF, yields structural validity by construction rather than by post-hoc check, and leaves standardised labels as evidence rather than foundation. That is the intent; the class-binding seam is not yet wired, and §3 says where the boundary falls today. To our knowledge the arrangement has not been reported for Brick/REC building-automation data, though absence from a search is not proof of novelty.

3. Our Position

The separation of concerns is the whole design: templates fix the *structure* of the graph in advance, and only the *interpretation* of each source row is left to a language model. The interpretation layer, which absorbs the variety of the input, writes nothing directly: the design binds its proposals as parameters to OTTR templates [7].

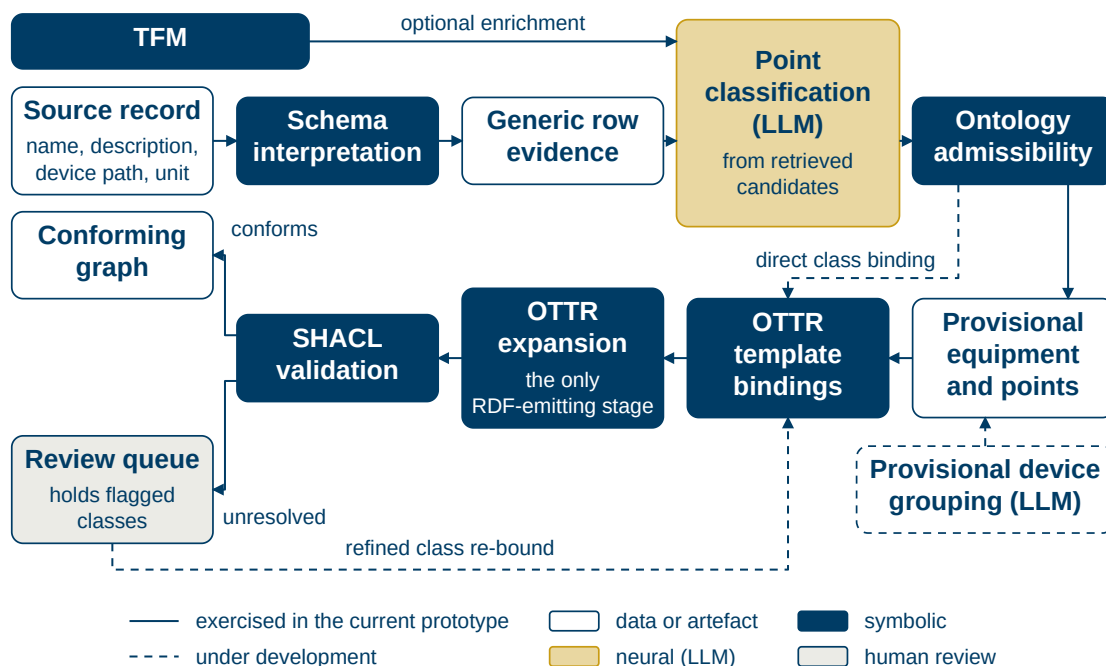


Figure 1: Intended division of labour. Agents interpret source evidence and structured metadata may enrich it; ontology checks and OTTR templates constrain what enters RDF. Solid stages are exercised in the current prototype; provisional device grouping, direct class binding and the review loop remain incomplete.

The intended pipeline is a single flow (Figure 1). Schema interpretation profiles the source’s columns, and from that profile each record’s own evidence, its name, description, device path and optional unit, is assembled generically to support both a proposed point classification and a provisional grouping of records into devices. A labelling standard, where the source carries one, enriches that evidence with standardised device semantics: an input to interpretation rather than a precondition for it. Every proposed class is checked for ontology admissibility, meaning that it exists in the target vocabulary and sits at the grain the slot expects. Admissible interpretations bind generic OTTR templates, whose expansion alone emits RDF, and SHACL validation then decides what is committed and what is escalated.

Row-level interpretation, meaning point classification and provisional device grouping, is the only neural work; schema inference, admissibility checking, template selection, expansion and validation are procedural. The pipeline is agentic in a narrow sense: components cooperating inside a symbolic frame, not an agent with free rein. The discipline, not the autonomy, is the contribution.

Validation and review close the flow, and neither of them is automated correction. When a shape fails, implicated bindings are withdrawn or the case is escalated, never the graph edited triple by triple,

and what does not resolve goes to a review queue. “Nothing yet, pending review” remains a decision the symbolic core is allowed to make.

Three claims follow, and they are of three different kinds. The first is a position for which §4 offers early evidence: *templates constrain the agent’s output, not its capability*. The classifier may reason about an ambiguous row however it likes, but it emits parameters to a template and never RDF, so structural validity is established once, in the templates, rather than reproduced by the agent on every row. The qualification is part of the claim rather than a concession to it: templates guarantee structural *validity*, not semantic *correctness*. A wrong class bound into a well-formed slot conforms by construction; the admissibility check tests vocabulary and grain, not whether the reading of the row was right.

The second is an architectural requirement, realised in part: *structured metadata is optional evidence, not a foundation*. Records are classified from their own row evidence, and TFM supplies national device semantics as enrichment where the source carries it. Standards change how good the evidence is, not the shape of the result: the pipeline should degrade as that structure thins, not stop.

The third is design intent, and nothing here measures it: *generalisability is a continuum*. The more structure a source carries, the less interpretation the agent must supply; the less it carries, the more it must. A TFM-coded BACnet export, a KNX project and a bare point list are the same problem at different points on that continuum, but we have exercised only the first.

The prototype realises this architecture in part, and we state its current extent plainly. Generic-first classification and ontology admissibility are implemented and exercised, and template expansion is the only path into RDF. The proposed class does not yet parameterise the templates (the current family templates fix their class), and equipment identity, grouping and dispatch still use the TFM component identifier: withholding it yields no RDF. Neither is a property of the architecture: the point template already accepts a class parameter and the classifier already receives the full device path, so evidence-based grouping and class-to-template binding are the designed next steps. Review, likewise, is a queue rather than a loop today: records are held, and the path back into binding is not yet wired.

4. Early Evidence from the ZEB Laboratory

We evaluate the pipeline on system 569 of the ZEB Laboratory [17], a zero-emission research building operated by NTNU and SINTEF. System 569 is the building’s room-automation system, exported as a TFM-coded CSV. The export carries 819 rows, of which 34 are delimiter-only artefacts of the export format and 785 carry a substantive record; those 785 records, spread across 32 rooms, are the corpus throughout this section. It is one system among several (heating and ventilation are separate, and out of scope here), but it is real, standardised, operational data of the kind a public estate produces, not a curated benchmark. Table 1 collects the measured quantities.

For each of the 785 substantive records the classifier builds an evidence view from the row’s own name, description, full device path and optional unit, and proposes a Brick point class wherever the row’s evidence supports one. Proposals are admissible by construction: candidates are drawn from the ontology’s Point subtree and the answer must be one of them, and across the 785 records the run produced no inadmissible proposal. What this guarantees is admissibility, not correctness: a class can exist, sit at the right grain, and still be the wrong reading of the row.

An earlier design put the standard’s lookup first and resolved 601 of the 785 records (76.6%) without a model call; the current point-classification path inverts that: the lookup result is evidence, not the class-selection route. That earlier design also passed the standard’s code to the classifier as a hint, and in that configuration it agreed with a corrected hand-authored reference table on 13 of 18 distinct classifications. The table is same-team work, so it measures consistency with our own modelling, never accuracy.

The 785 records account for themselves end to end: 570 reach the committed graph with a timeseries identifier and 20 without one; 4 are aggregate window readings the pipeline excludes; and 191 are held for modelling review, not forced into an ill-fitting class. That queue is the design working, not failing: a family that Brick 1.4 does not type faithfully belongs there, not in the graph under an approximate class.

Table 1

Measured counts for system 569: the census of 785 substantive records and the resulting graph. The 34 delimiter-only export rows are counted separately.

Quantity	Value
Substantive records; delimiter-only rows	785; 34
Rooms	32
Inadmissible proposals (constrained by construction)	0, across 785 records
Disposition of the records	570 graph with timeseries identifier; 20 without; 4 excluded; 191 to modelling review
Template expansion	4,998 triples, 14 template types
Room 3.12 vs. same-team fixture	isomorphic, 307 triples each
Project shape library	25 SHACL files; 0 violations, 4 warnings
Fault detection (unique injected sites)	3 of 3 deleted relationships; 49 of 53 cross-room; 46 of 49 identifier collisions; none of 36 wrong-but-valid
Identifier ablation	no component identifier, no RDF

A controlled ablation locates the dependency this disposition still carries. Withholding the component identifier yields no RDF at all, because identity is minted before classification runs. Evidence-based device grouping is the designed replacement.

The template selector expands all 32 rooms into a single graph of 4,998 triples, invoking 14 distinct template types. For Room 3.12 we hand-authored a Brick and RealEstateCore reference model, and given the CSV and the project’s own templates and mappings the slice produced for that room is isomorphic to it, 307 triples on each side, up to blank-node renaming. The reference model, templates and mappings are same-team work, so the check establishes reproduction of our own modelling rather than correctness of it.

Validation runs over the expanded graph. A project library of 25 SHACL shape files (structural checks, not full Brick or RealEstateCore semantic validation) reports no violations and four advisory warnings, all from a single shape noting four sensor-only rooms with no room controller. This establishes conformance to the project’s shapes, not Brick semantic correctness.

Controlled fault injection tests what that validation can see. It detects all three deleted required relationships, 49 of 53 cross-room misassignments and 46 of 49 identifier collisions, counted as unique injected sites, and none of 36 substitutions of a wrong-but-valid class. The last is the first of two residuals we treat as principled rather than unfinished: a class that exists and fits its slot conforms by construction to checks that read only structure; domain constraints may catch particular cases, but no growth of a structure-only library removes the residual. The same fault suite also guided the shape library’s development, so detection is reported on fault classes its authors had seen. On a violation the implementation withdraws the implicated bindings or escalates the case, within a three-iteration cap; an automated corrector is not implemented, so we report detection and escalation only.

We are deliberate about what is measured and what is not. The reference table and the Room 3.12 fixture are same-team artefacts and are not independent ground truth, so no figure here is an accuracy. Settling that would need independent annotation by two experts on a held-out set, which remains future work, as does a same-input baseline against a language-model generator. The ablation measures the identity dependency, not the standard’s semantic contribution: with the identifier withheld there is no graph to compare against. One building, one of its several automation systems and one export format are the whole of the evaluated scope, and the 785 records are a census rather than a sample, so we report counts and not intervals.

5. Discussion

Templates limit structural variation; they do not establish semantic truth. An agent that binds template parameters works inside a structure decided once, before any row was read, whereas an agent that emits free RDF can emit an invalid graph that must be checked after generation. What that buys is a smaller structural error surface, and nothing beyond it. A classification that is admissible and still wrong

enters the graph looking exactly like a correct one, which is what the faults show. Four cross-room misassignments also escaped, marking a second residual of the same kind: a check that reads only the graph cannot see a fault that leaves the graph internally consistent, and telling that case apart needs the source, since validating a mapping with its own output would be circular.

A standard is not one thing. The Tverrfaglig Merkesystem is a national vocabulary published by a public owner; the alignment from it to Brick is ours; the conventions a site follows in applying it are neither; and a vendor's device-path tail is neither again. Four authorities, four rates of change, four ways to be wrong. Our own catalogue keys them together, which is why moving to a second building is not one task but four. Separating them is the architectural work the measurements point at.

The two open seams are the next piece of that architecture. Binding a proposed class into a template would turn classification from a graph-inert proposal into something that changes what is emitted (today a wrong class and a right one expand identically), and grouping records into devices from their own evidence would remove the identifier dependency the ablation measured. Both are designed and neither is wired; automated correction sits behind both and remains future work.

Generalisation is still design intent: each of the building's other systems would need its own component families and templates rather than a repeat of this run, and other input formats are argued for in §3 and untested here. One defect belongs here rather than in the evidence. All 785 records are accounted for on this input, but the room parser's site-specific rules could silently omit substantive records on another: a small fault with a large moral, since the input-facing edge of the pipeline is where genericity is won or lost.

6. Conclusion and Next Steps

This paper argues a boundary: templates and ontology checks decide what enters the graph, agents interpret rows, and standardised metadata contributes evidence where it exists. The prototype exercises the construction side of it in parts (classification from row evidence, admissibility checking, template-only emission, structural validation) and measures what remains.

Two remainders are principled: conformance to a structure-only shape library, however far it grows, does not establish semantic truth, and a check that reads only the graph is blind to a fault that leaves the graph internally consistent. Both are why the architecture keeps admissibility checks and human review rather than resting on validation alone. Everything else is a development state: proposed classes are not yet bound into templates, equipment identity still reads the component identifier, and a standard's four authorities are still keyed together. Other systems, buildings and formats remain unevaluated.

Acknowledgements

This publication is based upon work from COST Action CA24121 – Knowledge Graphs in the Era of Large Language Models (KGELL), supported by COST (European Cooperation in Science and Technology, <https://www.cost.eu>).

Declaration on Generative AI

During the preparation and revision of this work, the authors used Claude and Claude Code (Anthropic) in order to: Paraphrase and reword, Improve writing style, Formatting assistance and Peer review simulation, and to write the scripts recomputing its figures; and Codex (ChatGPT, OpenAI) in order to: Peer review simulation. The tools worked from claims, numbers and structure the authors fixed first. No generated image appears here: the figure is drawn by a committed script, and an early FigureLabs iteration (Generate images) was discarded. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Bergés, D. Culler, R. K. Gupta, M. B. Kjærgaard, M. Srivastava, K. Whitehouse, Brick: Metadata schema for portable smart building applications, *Applied Energy* 226 (2018) 1273–1292. doi:10.1016/j.apenergy.2018.02.091.
- [2] K. Hammar, E. O. Wallin, P. Karlberg, D. Hälleberg, The RealEstateCore Ontology, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), *The Semantic Web – ISWC 2019*, Springer International Publishing, Cham, 2019, pp. 130–145. doi:10.1007/978-3-030-30796-7_9.
- [3] M. Perini, D. Antonucci, R. Giudice, M. S. Piscitelli, A. Capozzoli, BrickLLM: A Python library for generating Brick-compliant RDF graphs using LLMs, *SoftwareX* 30 (2025) 102121. doi:10.1016/j.softx.2025.102121.
- [4] J. Du, Y. Tian, M. Berktold, R. Streblow, D. Müller, Semantic-IoT Framework: Knowledge Graph Generation from IoT Platforms, in: *CEUR Workshop Proc.*, volume 4064, CEUR-WS, 2025.
- [5] Y. Qian, S. Zhang, H. Song, N. Sokhandan, H. Marlowe, D. Socolinsky, Brick-DICL: Dynamic In-Context Learning for Automated Brick Schema Classification, 2026. doi:10.48550/arXiv.2606.17637, arXiv:2606.17637 [cs.AI].
- [6] Statsbygg, PA 0802 Tverrfaglig merkesystem (TFM), Prosjekteringsanvisning (project design guideline) PA 0802, rev. 3, Statsbygg, 2024. URL: <https://dok.statsbygg.no/wp-content/uploads/2024/02/PA-0802-Tverrfaglig-merkesystem-TFM-rev-3.pdf>, in Norwegian.
- [7] M. G. Skjæveland, L. H. Karlsen, The Reasonable Ontology Templates Framework, *Transactions on Graph Data and Knowledge* 2 (2024) 5:1–5:54. doi:10.4230/TGDK.2.2.5.
- [8] M. Bakken, maplib: Interactive, Literal RDF Model Mapping for Industry, *IEEE Access* 11 (2023) 39990–40005. doi:10.1109/ACCESS.2023.3269093.
- [9] E. J. Nystad, F. Martín-Recuerda, A Pythonic Functional Approach for Semantic Data Harmonisation in the ILIAD Project, 2026. doi:10.48550/arXiv.2604.13042, arXiv:2604.13042 [cs.DB].
- [10] D. P. Lupp, M. Hodkiewicz, M. G. Skjæveland, Template libraries for industrial asset maintenance: A methodology for scalable and maintainable ontologies, in: *CEUR Workshop Proc.*, volume 2757, CEUR-WS, 2020, pp. 49–64.
- [11] C. Shimizu, P. Hitzler, Accelerating knowledge graph and ontology engineering with large language models, *Journal of Web Semantics* 85 (2025) 100862. doi:10.1016/j.websem.2025.100862.
- [12] S. Gerin, L. Joblot, N. Makhoul, F. Merienne, Ontology-driven LLM framework for knowledge graph in smart buildings, in: *Proc. Int. Symp. Autom. Robot. Constr.*, International Association for Automation and Robotics in Construction (IAARC), 2025, pp. 1057–1064. doi:10.22260/ISARC2025/0137.
- [13] Z. Wang, J. Ma, J. Cai, M. Qu, Point2Brick: Automating building semantic model creation via multi-stage confidence-aware Large Language Model agents, *Energy and Buildings* 362 (2026) 117516. doi:10.1016/j.enbuild.2026.117516.
- [14] H. Zhang, W. J. Schmidt, X. Shen, Q. Cao, S. Monka, A. Paschke, Knowledge Graph Construction towards a Graph RAG-Enhanced Intelligent Maintenance Chatbot, in: *CEUR Workshop Proc.*, volume 4064, CEUR-WS, 2025.
- [15] R. Mehmandarov, A. Waaler, A Framework for Semantic Identifier Resolution and LLM Grounding in Engineering Systems, in: *CEUR Workshop Proc.*, volume 4064, CEUR-WS, 2025.
- [16] M. H. T. De Boer, Q. S. Smit, M. Van Bekkum, A. Meyer-Vitali, T. Schmid, Modular Design Patterns for Generative Neuro-Symbolic Systems, in: *CEUR Workshop Proc.*, volume 3749, CEUR-WS, 2024.
- [17] A. Nocente, B. Time, H. M. Mathisen, T. Kvande, A. Gustavsen, The ZEB Laboratory: The development of a research tool for future climate adapted zero emission buildings, in: *J. Phys. Conf. Ser.*, volume 2069, IOP Publishing, 2021, p. 012109. doi:10.1088/1742-6596/2069/1/012109.