Neuro-Symbolic Design Patterns: A Semantic Representation of Neuro-Symbolic Architectures

Yashrajsinh Chudasama[1,2,*], Disha Purohit[2,3], Enrique Iglesias[2,3], Mahsa Forghani[2,4] and Maria-Esther Vidal[1,2,3]

[1]TIB-Leibniz Information Centre for Science and Technology, Hannover, Germany
[2]Leibniz University Hannover
[3]L3S Research Center, Hannover, Germany
[4]DigiStrucMed Program, Dean's Office for Academic Career Development, Hannover Medical School, Germany

Abstract

Neuro-symbolic (NeSy) systems are commonly described using taxonomies and design patterns that capture different perspectives of the interaction between neural and symbolic components. However, these representations remain largely human-oriented and do not support interoperability or reasoning across taxonomies. This paper introduces a semantic framework for representing neuro-symbolic design patterns, taxonomies, and architectures in a machine-interpretable form. The proposed approach formalizes existing NeSy taxonomies using Boxology patterns, introduces the NeSy ontology for representing taxonomies, design patterns, and their alignments, and materializes these representations in a Neuro-Symbolic Knowledge Graph (NeSy-KG). Building on this representation, we define an axiomatic reasoning framework that infers implicit pattern hierarchies, equivalences, and cross-taxonomy alignments. An empirical evaluation on a corpus of published neuro-symbolic systems demonstrates that the proposed framework identifies interoperability relationships across alternative taxonomic representations and enriches the knowledge graph with previously implicit knowledge. The results establish a foundation for representing, comparing, and reasoning about NeSy systems using semantic technologies.

Keywords

Neuro-symbolic AI, Symbolic Reasoning, Knowledge Representation, Knowledge Graphs, Design Patterns

CEUR-WS.org/Vol-4313/kgnesy-paper23.pdf

1. Introduction

Neuro-symbolic (NeSy) systems combine neural learning and symbolic reasoning to support explainable, knowledge-aware, and trustworthy artificial intelligence. As the field has matured, several taxonomies and design frameworks have been proposed to classify and describe NeSy systems [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Examples include Kautz's taxonomy of NeSy integration strategies, Kierner's architectural archetypes for clinical decision support systems, and the Boxology design patterns proposed by van Bekkum et al. [5]. While these frameworks provide valuable perspectives on NeSy architectures, they employ different vocabularies, levels of abstraction, and classification criteria, making it difficult to compare systems across taxonomies. The main problem addressed in this paper is the lack of a machine-interpretable representation that enables interoperability and reasoning across existing NeSy taxonomies and design patterns. Current representations are primarily intended for human interpretation and do not provide explicit semantic relationships among categories, architectural patterns, and NeSy systems. Consequently, alignments between taxonomies remain implicit and cannot be automatically exploited for analysis or knowledge discovery. The main idea of this paper is to represent NeSy taxonomies, design patterns, and architectures within a common semantic framework. We formalize existing taxonomies using Boxology patterns, define an ontology representing taxonomies

Third Workshop on Knowledge Graphs and Neurosymbolic AI (KG-NeSy 2026), in ISWC 2026 Workshops Joint Proceedings, October 25–26, 2026, Bari, Italy

*Corresponding author.
yashrajsinh.chudasama@tib.eu (Y. Chudasama); disha.purohit@tib.eu (D. Purohit); iglesias@l3s.de (E. Iglesias); mahsa.forghani.tehrani@stud.uni-hannover.de (M. Forghani); vidal@l3s.de (M. Vidal)
0000-0003-3422-366X (Y. Chudasama); 0000-0002-1442-335X (D. Purohit); 0000-0002-8734-3123 (E. Iglesias); 0009-0001-6000-2636 (M. Forghani); 0000-0003-1160-8727 (M. Vidal)

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

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-05

and pattern alignments, and materialize these representations in a NeSy Knowledge Graph (NeSy-KG). On top of this representation, we define an axiomatic reasoning framework that derives implicit pattern hierarchies, equivalence relationships, and cross-taxonomy alignments.

The main contribution of this paper is a semantic representation and reasoning framework for NeSy design patterns. Specifically, we i) formalize existing NeSy taxonomies using a common pattern-based representation, ii) introduce an ontology for representing taxonomies, design patterns, and their alignments, iii) define a NeSy KG that integrates multiple taxonomic perspectives, and iv) propose an axiomatic reasoning framework for discovering implicit relationships among patterns and taxonomies. An empirical evaluation based on a corpus of NeSy systems [9, 11] demonstrates that the proposed framework can integrate alternative taxonomic viewpoints within a common semantic representation, capture interoperability relationships among taxonomies, and derive new pattern hierarchies and cross-taxonomy alignments through semantic reasoning.

2. Preliminaries

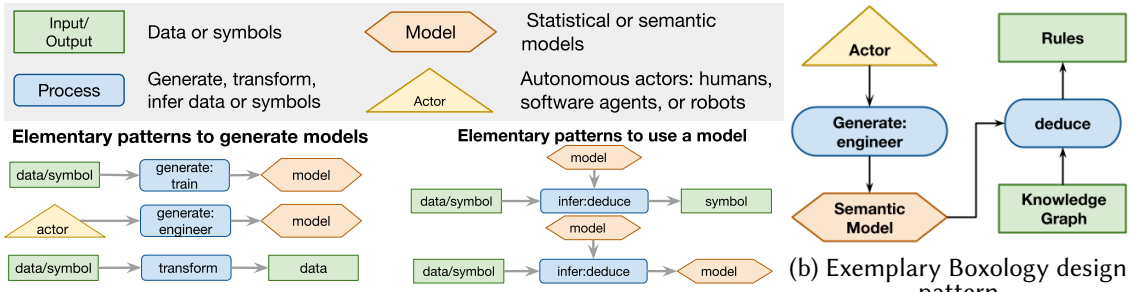
NeSy systems can be described from different perspectives and levels of abstraction. Van Bekkum et al. [5] propose Boxology, a graphical language whose elementary patterns capture the structural building blocks of NeSy architectures. Kierner et al. [9] classify clinical NeSy systems according to recurring architectural archetypes, while Kautz et al. [7] proposes a high-level taxonomy that characterizes the interaction between symbolic and neural components. Although these approaches describe NeSy architectures, they employ different vocabularies, levels of granularity, and classification criteria. As a consequence, the same NeSy system may be represented differently depending on the selected taxonomy, creating interoperability challenges that hinder comparison, integration, and analysis. This section introduces these complementary views and motivates the need for a semantic representation that makes their relationships explicit and machine-interpretable.

2.1. Boxology Language and Elementary Patterns

Van Bekkum et al. [5] proposed the Boxology notation to describe NeSy architectures as typed graphical patterns; Figure 1 illustrates elementary patterns for creating and using models, together with an exemplary Boxology diagram. A boxology pattern is composed of four kinds of elements: *input/output artefacts*, *processes*, *models*, and *actors*. Boxology provides a graphical language whose nodes represent different elements, while edges represent the flow of information between these elements. *Input/output* artefacts denote data or symbolic objects. *Processes* denote operations that generate, transform, or infer data or symbols. *Models* denote statistical, neural, or semantic models. *Actors* denote autonomous entities, such as humans, software agents, or robots. Elementary boxology patterns describe recurrent operations in NeSy systems. They include patterns for generating models, transforming artefacts, engineering artefacts, and using models for deduction. In this paper, these elementary patterns provide the structural layer used to represent higher-level NeSy taxonomies. Valid Boxology diagrams are defined by combining elementary design patterns that represent admissible input–process–output combinations. They support the representation of complex systems through the composition of simpler patterns. The pattern in Figure 1(b) shows how such patterns can be composed to describe a complete NeSy system, from the construction of a semantic model to the deduction of rules from a KG.

2.2. NeSy Taxonomies

Several taxonomies classify NeSy systems based on different criteria. Kautz et al. [7] distinguishes six categories based on the way symbolic and neural components interact: i) symbols guiding neural processing (*Symbolic→Neuro*), ii) neural outputs driving symbolic reasoning (*Neuro→Symbolic*), iii) co-operative neural and symbolic inference (*Neuro+Symbolic*), iv) symbolic knowledge influencing neural learning (*Neuro:Symbolic→Neuro*), v) symbolic knowledge embedded in neural models ($Neuro_{\{\mathrm{Symbolic}\}}$),



(a) Elementary patterns to generate and use a model

(b) Exemplary Boxology design pattern

Figure 1: Boxology design patterns. The left diagram presents the elementary patterns, while the right diagram illustrates how these patterns can be combined to describe a complete NeSy workflow. Adapted figure from Bendler et al. [12]

and vi) symbolic reasoning embedded within neural architectures (*Neuro[Symbolic]*). In contrast, Kierner et al. [9] classify NeSy clinical decision support systems according to architectural archetypes: i) rules embedded in machine learning models (REML), ii) machine learning followed by rule-based inference (MLRB), iii) rule-based preprocessing followed by machine learning prediction (RBML), iv) rules guiding machine learning training (RMLT), and v) parallel ensembles of rule-based and machine learning methods (PERML). While these taxonomies describe NeSy systems, they do so from different perspectives and levels of abstraction. Kautz focuses on the conceptual interaction between neural and symbolic components, Kierner classifies recurring architectural archetypes observed in practical systems, and van Bekkum's Boxology provides elementary design patterns that describe the operational building blocks of NeSy architectures. Consequently, the same NeSy system may be assigned different categories depending on the taxonomy used and may simultaneously instantiate multiple Boxology patterns. This heterogeneity creates interoperability challenges, since the relationships among taxonomies are not explicitly represented and must typically be identified manually. Table 1 shows that categories from Kautz's taxonomy can be aligned with Kierner's archetypes through a common Boxology representation. For example, the Kautz category *Neuro+Symbolic* corresponds to Kierner's *PERML* archetype, while $Neuro_{\text{Symbolic}}$ aligns with *REML*. These alignments suggest that different taxonomies often describe the same underlying architectural principles using different vocabularies and levels of abstraction. The remainder of the paper formalizes these relationships through the NeSy ontology and NeSy KG, enabling their representation, integration, and automated reasoning.

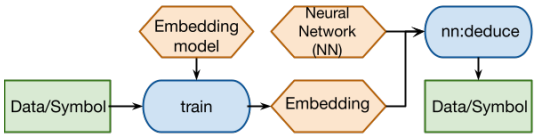
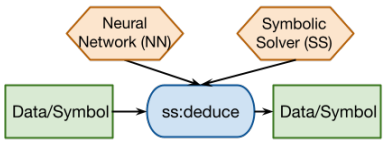
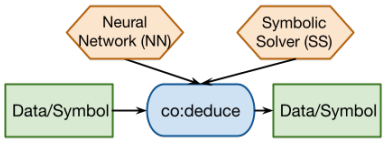
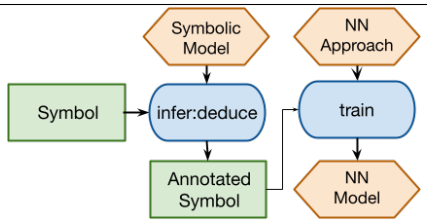
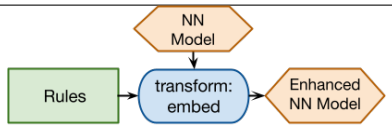
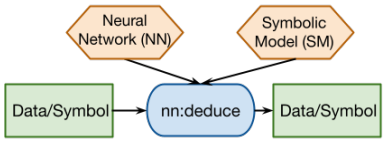
3. Semantic Representation of NeSy Design Patterns

NeSy taxonomies and design patterns provide complementary perspectives for describing NeSy systems. However, these representations are primarily intended for human interpretation and use different vocabularies, levels of abstraction, and classification criteria. Consequently, the same NeSy system may be represented differently across taxonomies, hindering interoperability, comparison, and analysis. To address this challenge, we introduce a semantic representation framework for NeSy design patterns. First, NeSy taxonomies are formalized using a common set of Boxology patterns. Second, an ontology provides a machine-interpretable vocabulary for representing taxonomies, design patterns, and their relationships . Third, the ontology is instantiated as a Neuro-Symbolic Knowledge Graph (NeSy-KG) that captures taxonomies, design patterns, and NeSy systems in a unified semantic representation. This framework establishes explicit alignments across taxonomies and provides the foundation for reasoning over NeSy systems.

3.1. Extensions to Boxology Design Patterns by Van Bekkum et al. [5]

We extend Bekkum's boxology [5] in two directions. First, we include additional elementary patterns required to describe knowledge-intensive NeSy systems, such as model transformation, data–symbol

Table 1
Neuro-symbolic categories represented as boxology patterns.

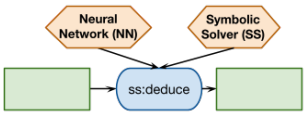
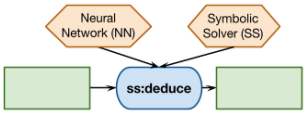
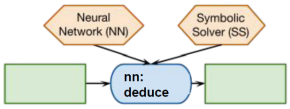
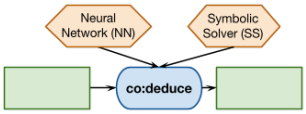
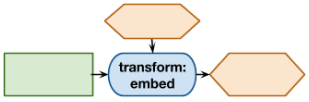
Kautz Category	Kierner Category	Bekkum Boxology Pattern
Symbolic → **Neuro**: Symbolic information guides the neural component, e.g., by embedding symbolic data before it is consumed by a neural model.	**Rules influence ML training (RMLT)**: Rules can help with feature selection or activation, or preprocess training data to build a simpler ML model.	
Neuro → **Symbolic**: A neural model produces outputs that are consumed by a symbolic solver or reasoning component.	**ML pre-processes input data for rule-based inference (MLRB)**: Benchmark, validate or constrain Machine Learning outputs with existing knowledge based on rules.	
Neuro + **Symbolic**: Neural and symbolic components cooperate during inference while remaining distinct modules.	**Parallel ensemble of rules and ML (PERML)**: Ensemble methods improve the predictive performance of a single model by training multiple models and combining their predictions.	
Neuro:Symbolic → **Neuro**: Symbolic rules or annotations influence the learning process of a neural model.	**Rules influence ML training (RMLT)**: Rules can help with feature selection or activation, or preprocess training data to build a simpler ML model.	
Neuro$_{\{\text{Symbolic}\}}$: Symbolic rules are encoded into the internal structure of a neural model.	**Rules embedded in ML architecture (REML)**: Rules are embedded in the ML architectures to enhance the performance of models.	
Neuro[Symbolic]: A symbolic reasoner is embedded inside, or invoked, as a subroutine by, a neural architecture.	**Rules pre-process input data for ML (RBML)**: An inference pipeline with rules solves challenging tasks and provides a preprocessing engine for more relevant features.	

transformation, and the explicit generation of embeddings. Second, we introduce typed process annotations for patterns involving more than one model. In particular, a process may take two models as input. We use a prefix in the process name to indicate which model executes the main operation. For example, `ss:deduce` denotes a deduction process executed by the symbolic solver, which may call a neural model as a subroutine. Analogously, `nn:deduce` denotes a deduction process executed by the neural model, which may call a symbolic model. Finally, `co:deduce` denotes a cooperative process in which the neural and symbolic models act as co-routines. We also use `transform:embed` for a specialized transformation process that generates embeddings from data or symbolic artefacts using a given model. Table 2 defines the newer extension incorporating additional processes and design patterns.

3.2. Formalization of Neuro-Symbolic (NeSy) Taxonomies

The taxonomies introduced in Section 2 describe Neuro-Symbolic (NeSy) systems from complementary perspectives. Kautz's taxonomy categorizes NeSy systems according to their integration strategy, Kierner's taxonomy emphasizes the interaction between machine learning and symbolic reasoning, while van Bekkum's taxonomy characterizes elementary architectural patterns. Although these taxonomies employ different conceptual abstractions and terminology, they describe related classes of NeSy systems. To provide a unified semantic representation, the proposed ontology separates taxonomy definitions from the architectural structures that realize them. This separation enables multiple taxonomies to coexist without introducing unintended logical equivalences between their categories.

Table 2
Extension incorporating additional processes and design patterns

Boxology Process	Extension
	We introduce typed process annotations for patterns involving more than one model. A process may take two models as input, e.g., **Neural Network (NN)** and **Symbolic Solver (SS)** and indicate the model it uses to derive the output.
	The **ss:deduce** signifies a deduction process carried out by the symbolic solver, which may invoke a neural model as a subroutine.
	The **nn:deduce** signifies a deduction process carried out by the neural network, which may invoke a neural model as a subroutine.
	The **co:deduce** denotes a cooperative process in which the neural and symbolic models act as co-routines.
	The **transform:embed** function is a specialized transformation process that generates embeddings from data or symbolic artifacts using a given model.

Design Pattern: A design pattern is represented as a typed directed graph $P = (N, E, \tau)$, where N is a finite set of nodes, $E \subseteq N \times N$ is a set of directed edges, and τ assigns a semantic type to every node. Node types correspond to the Boxology primitives introduced in Section 2, namely data and symbolic artefacts, processes, models, and actors. Consequently, each design pattern captures the structural organization of a reusable architectural component independently of any particular taxonomy.
Given two design patterns $P_1 = (N_1, E_1, \tau_1)$ and $P_2 = (N_2, E_2, \tau_2)$, a *pattern composition* is a design pattern $P = (N, E, \tau)$ satisfying $N_1 \subseteq N$, $N_2 \subseteq N$, $E_1 \subseteq E$, and $E_2 \subseteq E$. In this case, P_1 and P_2 are called subpatterns of P. More generally, a design pattern may contain a collection of subpatterns $\{P_1, \ldots, P_n\}$, allowing complex NeSy architectures to be represented hierarchically. For example, let P_1 denote the pattern that transforms symbolic representations into embeddings and P_2 denote a neural deduction pattern. A composite design pattern representing a Symbolic→Neuro architecture may contain both P_1 and P_2, making them subpatterns of the composite architecture. This hierarchical organization motivates the introduction of the `nesy:subPatternOf` relation in the ontology.
Neuro-Symbolic Taxonomy: A NeSy taxonomy is represented as a pair $T = (C, R)$, where $C = \{c_1, \ldots, c_n\}$ is a finite set of taxonomy categories and R defines semantic relationships

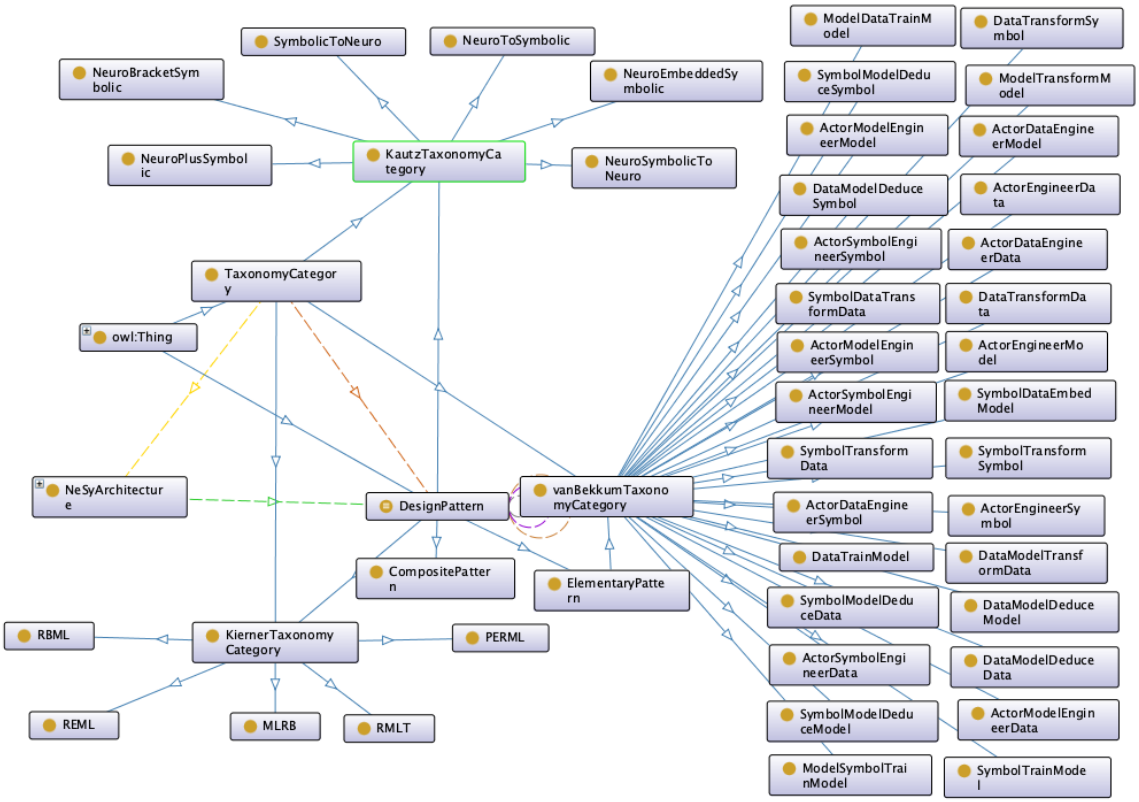


Figure 2: Visualization of the NeSy ontology generated with Protégé [13]. Grey boxes denote ontology classes. Solid blue arrows represent `rdfs:subClassOf` relations, whereas dashed arrows denote additional OWL axioms, such as `owl:equivalentClass` and domain/range constraints of object properties. Instance-level relations, including `nesy:equivalentPattern`, `nesy:subPatternOf`, and `nesy:alignedWith`, are not shown in this class hierarchy visualization.

among these categories within the taxonomy. Examples include the categories `Symbolic→Neuro`, `Neuro→Symbolic`, `Neuro+Symbolic`, and `Neuro[Symbolic]`. Each taxonomy is represented independently in the ontology, allowing multiple taxonomies to coexist without asserting logical equivalence among their categories. Rather than directly equating categories from different taxonomies, each taxonomy category is associated with one or more design patterns through a mapping $f : C \to 2^P$, where P denotes the set of design patterns. The mapping associates a conceptual category with the architectural structures that realize it. Categories belonging to different taxonomies may therefore correspond to similar or overlapping sets of design patterns while remaining semantically independent. Expert-established correspondences between taxonomy categories are explicitly documented using the annotation property `nesy:categoryAlignedWith`, which records conceptual alignments without introducing logical entailments.

Example: Let c_{SN} denote the category `Symbolic→Neuro`. The mapping $f(c_{SN}) = \{P_{\text{embed}}, P_{\text{deduce}}\}$ associates this category with the design patterns `transform:embed` and `infer:deduce`. Similarly, the corresponding category in Kierner's taxonomy is linked to the same architectural structures through the annotation property `nesy:categoryAlignedWith`, while both categories remain distinct ontology classes. Consequently, taxonomies employing different terminology and levels of abstraction can be compared through their associated design patterns without asserting class equivalence. Table 1 summarizes these mappings and provides the semantic foundation for the NeSy ontology and the resulting knowledge graph.

Table 3
Main classes and predicates introduced by the NeSy ontology.

Ontology Element	Description
Classes	
`nesy:Taxonomy`	Abstract class representing a Neuro-Symbolic taxonomy.
`nesy:TaxonomyCategory`	Abstract class representing a category of a Neuro-Symbolic taxonomy.
`nesy:DesignPattern`	Reusable NeSy architectural design pattern (equivalent to `t4b:BoxologyPattern`).
`nesy:ElementaryPattern`	Minimal reusable design pattern representing an elementary Boxology pattern.
`nesy:CompositePattern`	Design pattern composed of one or more elementary patterns.
`nesy:KautzTaxonomy`, `nesy:KiernerTaxonomy`, `nesy:vanBekkumTaxonomy`	Represent the three Neuro-Symbolic taxonomies considered in this work.
`nesy:KautzTaxonomyCategory`, `nesy:KiernerTaxonomyCategory`, `nesy:vanBekkumTaxonomyCategory`	Represent the category hierarchies of the corresponding taxonomies.
`nesy:SymbolicToNeuro`, `nesy:REML`, `nesy:SymbolTrainModel`, …	Examples of taxonomy categories represented as subclasses of their corresponding taxonomy-category class.
Predicates	
`nesy:subPatternOf`	Relates a design pattern to another design pattern that structurally contains it.
`nesy:equivalentPattern`	Relates design-pattern instances that exhibit equivalent architectural organization.
`nesy:alignedWith`	Relates design-pattern instances that correspond across different taxonomy descriptions.
`nesy:categoryAlignedWith`	Annotation property documenting expert-established correspondences between taxonomy categories without asserting logical equivalence.
`nesy:isCategoryOf`	Annotation property linking a taxonomy-category class to the taxonomy to which it belongs.

3.3. Ontology of NeSy Design Patterns

Figure 2 presents the NeSy ontology, which provides a semantic representation of Neuro-Symbolic (NeSy) taxonomies, taxonomy categories, and architectural design patterns. The ontology explicitly distinguishes conceptual classifications from the architectural structures that realize them. While multiple taxonomies describe NeSy systems from different perspectives, reusable design patterns provide a common structural representation that enables these taxonomies to be related without introducing unintended logical equivalences. This separation allows several taxonomies to coexist while preserving their individual semantics. To represent these concepts in a machine-interpretable form, the ontology is implemented using RDF, RDFS, and OWL, and extends the T4B ontology [12]. The T4B ontology provides a semantic representation of the original Boxology language, including artefacts, processes, models, actors, elementary patterns, and composite patterns. The proposed ontology extends this vocabulary with concepts required to represent NeSy taxonomies, taxonomy categories, architectural design patterns, and the relationships among them. By relying on standard Semantic Web technologies [14], the ontology supports semantic integration, querying, and rule-based reasoning over heterogeneous NeSy architectures. Table 3 summarizes the principal classes and predicates introduced by the ontology. The NeSy ontology is expressed in the $\mathcal{SHI}$ Description Logic, supporting transitive object properties,

role hierarchies, and inverse-role semantics. This expressivity is sufficient to capture the structural organization of neuro-symbolic design patterns and their taxonomic relationships while ensuring decidable reasoning under OWL 2 DL. Reasoning beyond standard OWL semantics is realized through a custom GraphDB rule set, which operationalizes additional pattern relations, including reflexivity, equivalence propagation, and mutual subsumption. This hybrid reasoning approach combines the formal semantics of Description Logics with rule-based inference to support the classification and analysis of NeSy architectures.

Classes for Taxonomies, Categories, and Design Patterns: The ontology introduces the abstract classes `nesy:Taxonomy` and `nesy:TaxonomyCategory` to represent NeSy taxonomies and their conceptual categories, respectively. Three specialized taxonomy classes, namely `nesy:KautzTaxonomy`, `nesy:KiernerTaxonomy`, and `nesy:vanBekkumTaxonomy`, capture the three taxonomies considered in this work. Their corresponding taxonomy categories are represented through the classes `nesy:KautzTaxonomyCategory`, `nesy:KiernerTaxonomyCategory`, and `nesy:vanBekkumTaxonomyCategory`. The individual categories of each taxonomy, such as `nesy:SymbolicToNeuro`, `nesy:REML`, and `nesy:SymbolTrainModel`, are modeled as subclasses of their corresponding taxonomy-category class. Architectural structures are represented by the class `nesy:DesignPattern`, which is defined as equivalent to `t4b:BoxologyPattern`. Consequently, every NeSy design pattern is represented using the Boxology structural model provided by T4B while remaining compatible with the NeSy-specific vocabulary. The class `nesy:DesignPattern` is further specialized into `nesy:ElementaryPattern` and `nesy:CompositePattern`, enabling complex NeSy architectures to be represented hierarchically as compositions of reusable structural patterns. Taxonomy-category classes classify design-pattern instances, thereby establishing the semantic connection between conceptual taxonomies and their architectural realizations.

Relationships among Taxonomies and Design Patterns: The ontology introduces three complementary relationships to represent NeSy knowledge. First, the annotation property `nesy:categoryAlignedWith` records expert-established correspondences between taxonomy categories belonging to different taxonomies. Unlike `owl:equivalentClass`, this annotation documents conceptual similarities without asserting logical equivalence or equality of class extensions. Consequently, different taxonomies remain semantically independent while their corresponding categories can still be related. Second, the object property `nesy:subPatternOf` represents structural composition among design patterns. If a design pattern P_1 forms part of another design pattern P_2, then P_1 `nesy:subPatternOf` P_2. This relationship enables complex NeSy architectures to be represented as hierarchical compositions of reusable elementary and composite patterns. Third, the object property `nesy:equivalentPattern` relates design-pattern instances that exhibit equivalent architectural organization, regardless of the taxonomy used to classify them. In addition, the symmetric property `nesy:alignedWith` records correspondences between design-pattern instances originating from different taxonomy descriptions while preserving their individual classifications. Together, these relationships separate conceptual taxonomy alignment from structural architectural reasoning. RDF, RDFS, and OWL provide the structural semantics of the ontology, whereas the semantics of the NeSy-specific predicates, including `nesy:subPatternOf`, `nesy:equivalentPattern`, and `nesy:alignedWith`, are defined through the axiomatic system presented in section 4 and implemented as custom inference rules in GraphDB (section 5); implementation details are available in our repository[1].

3.4. Neuro-Symbolic Knowledge Graph – NeSy KG

The ontology presented in the previous subsection provides the semantic schema for representing Neuro-Symbolic taxonomies, taxonomy categories, architectural design patterns, and NeSy systems. Based on this ontology, we construct the Neuro-Symbolic Knowledge Graph (NeSy-KG), which instantiates these concepts as interconnected entities and enables semantic querying and rule-based reasoning over heterogeneous NeSy architectures.

Knowledge Graph (KG) [14]. A knowledge graph is a directed labeled graph $KG = (V, E, L)$, where V is a set of entities, L is a vocabulary of predicates, and $E \subseteq V \times L \times V$ is a set of labeled relation-

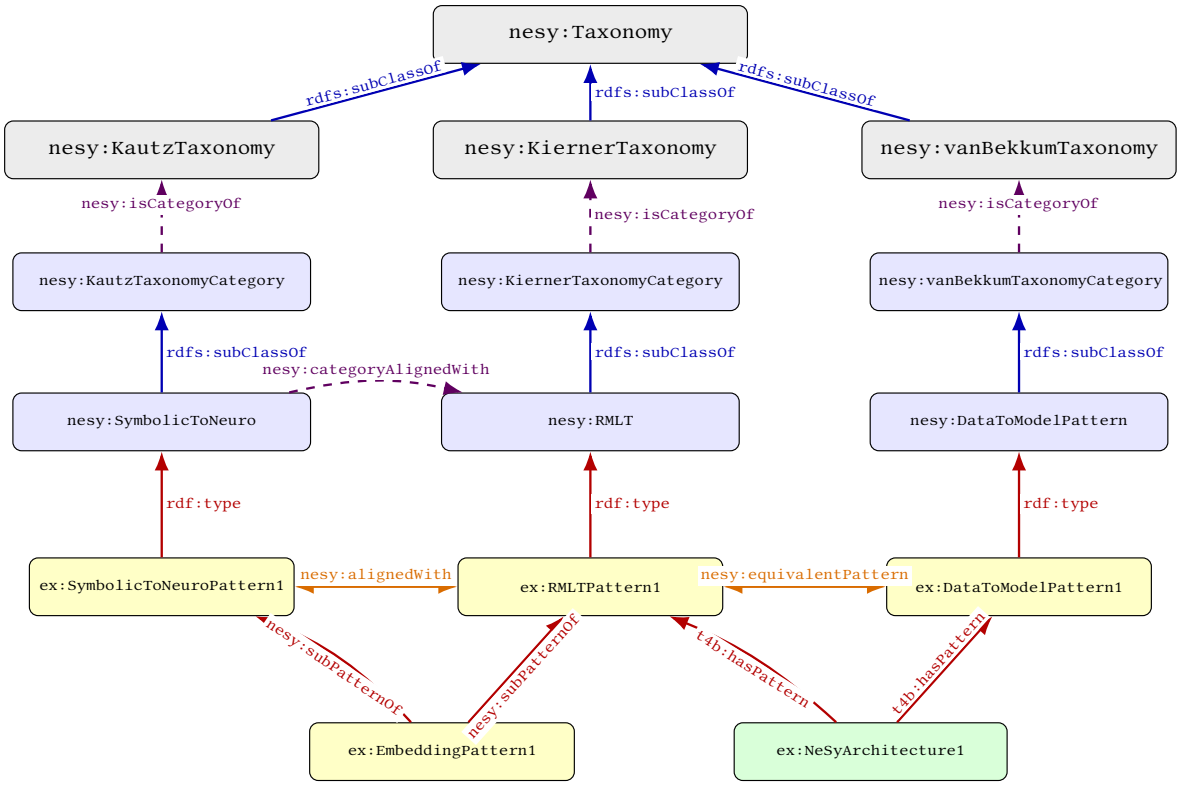


Figure 3: Illustrative fragment of the NeSy-KG. Grey boxes represent taxonomy classes, blue boxes represent taxonomy-category classes, yellow boxes represent design-pattern instances, and the green box represents a NeSy architecture instance. Solid blue arrows denote class-specialization relations, dashed violet arrows denote taxonomy annotations, and red and orange arrows denote instance-level structural and alignment relations.

ships among entities. Knowledge graphs provide a machine-interpretable representation of domain knowledge that supports semantic integration, querying, and automated reasoning.

NeSy Knowledge Graph. A NeSy-KG is a knowledge graph $KG = (V, E, L)$ whose entities represent NeSy architectures, design-pattern instances, taxonomy entities, and taxonomy-category classes. Relationships among entities are expressed using predicates from the NeSy ontology, the T4B ontology, and the RDF, RDFS, and OWL vocabularies, including `nesy:subPatternOf`, `nesy:alignedWith`, `nesy:equivalentPattern`, `t4b:hasPattern`, `rdf:type`, and `rdfs:subClassOf`. The annotation properties `nesy:categoryAlignedWith` and `nesy:isCategoryOf` record expert-established correspondences among taxonomy-category classes and associate each taxonomy-category class with its corresponding taxonomy, respectively. The predicate `rdf:type` associates design-pattern and architecture instances with ontology classes, whereas `rdfs:subClassOf` represents specialization relationships among ontology classes. The predicates `nesy:subPatternOf`, `nesy:alignedWith`, and `nesy:equivalentPattern` capture complementary relationships among design-pattern instances. The property `nesy:alignedWith` represents interpretative semantic correspondences between patterns classified according to independently defined taxonomies (e.g., Kautz and Kierner), whereas `nesy:equivalentPattern` relates pattern instances that represent the same architectural organization (e.g., Kierner's `RMLT` and van Bekkum's `DataToModelPattern`). Together, these predicates separate conceptual taxonomy alignment from architectural reasoning, allowing multiple taxonomies to coexist while preserving their individual semantics. Throughout the paper, the prefixes `nesy:`, `t4b:`, `rdf:`, `rdfs:`, and `owl:` identify the namespaces defining the corresponding classes and predicates. Figure 3 illustrates a fragment of the NeSy-KG; it contains taxonomy-category classes, design-pattern instances, and a NeSy architecture together with representative relationships, including `rdf:type`, `t4b:hasPattern`, `nesy:subPatternOf`, `nesy:alignedWith`, and `nesy:equivalentPattern`. Unlike the ontology, the NeSy-KG represents concrete entities and their relationships. This separation enables architectures described using different NeSy taxonomies to be compared through their structural design patterns while preserving the semantics of the original taxonomies.

Table 4
Axiomatic semantics of NeSy design-pattern relationships.

Axiom	Rule	Meaning
SP1 (Reflexivity)	IF $(P, \texttt{rdf:type}, \texttt{nesy:DesignPattern})$ THEN $(P, \texttt{nesy:subPatternOf}, P)$	Every design pattern is a subpattern of itself.
SP2 (Transitivity)	IF $(P_1, \texttt{nesy:subPatternOf}, P_2)$ AND $(P_2, \texttt{nesy:subPatternOf}, P_3)$ THEN $(P_1, \texttt{nesy:subPatternOf}, P_3)$	Indirect structural containment is inferred transitively.
SP3 (Mutual Subsumption)	IF $(P_1, \texttt{nesy:subPatternOf}, P_2)$ AND $(P_2, \texttt{nesy:subPatternOf}, P_1)$ THEN $(P_1, \texttt{nesy:equivalentPattern}, P_2)$	Patterns that mutually contain each other are structurally equivalent.
EP1 (Reflexivity)	IF $(P, \texttt{rdf:type}, \texttt{nesy:DesignPattern})$ THEN $(P, \texttt{nesy:equivalentPattern}, P)$	Every design pattern is equivalent to itself.
EP2 (Symmetry)	IF $(P_1, \texttt{nesy:equivalentPattern}, P_2)$ THEN $(P_2, \texttt{nesy:equivalentPattern}, P_1)$	Architectural equivalence is bidirectional.
EP3 (Transitivity)	IF $(P_1, \texttt{nesy:equivalentPattern}, P_2)$ AND $(P_2, \texttt{nesy:equivalentPattern}, P_3)$ THEN $(P_1, \texttt{nesy:equivalentPattern}, P_3)$	Architecturally equivalent patterns form equivalence classes.
EP4 (Preservation of Superpatterns)	IF $(P_1, \texttt{nesy:equivalentPattern}, P_2)$ AND $(P_1, \texttt{nesy:subPatternOf}, P_3)$ THEN $(P_2, \texttt{nesy:subPatternOf}, P_3)$	Equivalent patterns inherit the same containing patterns.
EP5 (Preservation of Subpatterns)	IF $(P_1, \texttt{nesy:equivalentPattern}, P_2)$ AND $(P_3, \texttt{nesy:subPatternOf}, P_1)$ THEN $(P_3, \texttt{nesy:subPatternOf}, P_2)$	Equivalent patterns inherit the same contained patterns.

4. Reasoning over NeSy Systems

The NeSy ontology defines the classes and predicates required to represent NeSy taxonomies, taxonomy categories, design patterns, and NeSy architectures. However, several structural relationships among design-pattern instances cannot be derived solely through the class hierarchy. We therefore define an axiomatic system that specifies the semantics of the predicates `nesy:subPatternOf` and `nesy:equivalentPattern`. The resulting inference rules enable the NeSy-KG to derive implicit pattern hierarchies and architectural equivalences while preserving the conceptual independence of the taxonomies represented in the ontology.

The reasoning process operates on design-pattern instances rather than on taxonomy-category classes. Correspondences between taxonomy categories are explicitly documented using the annotation property `nesy:categoryAlignedWith` and are not interpreted as logical equivalences. Similarly, `nesy:alignedWith` records explicit correspondences between design-pattern instances without implying that the related patterns are structurally equivalent. By contrast, `nesy:equivalentPattern` has formal inferential semantics and can therefore trigger additional structural relationships. Table 4 summarizes the axiomatic system used to derive these implicit statements.

The axioms SP1–SP3 define the semantics of structural containment among design patterns. SP1 establishes reflexivity for instances of `nesy:DesignPattern`, ensuring that each pattern belongs to its own structural hierarchy. SP2 enables transitive traversal of nested pattern compositions. For example, if an embedding pattern is a subpattern of a learning pattern and the learning pattern is a subpattern of a composite NeSy architecture, then the embedding pattern is also inferred to be a subpattern of that architecture. SP3 connects structural containment with equivalence: if two patterns are subpatterns of each other, they describe the same structural organization and are therefore related through `nesy:equivalentPattern`.

The axioms EP1–EP5 define the semantics of architectural equivalence among design-pattern instances. EP1–EP3 make `nesy:equivalentPattern` reflexive, symmetric, and transitive. Consequently, patterns representing the same architectural organization form equivalence classes, even when they have

different identifiers or are classified according to different taxonomy categories. EP4 and EP5 propagate structural containment across equivalent patterns. EP4 ensures that equivalent patterns share their superpatterns, whereas EP5 ensures that they share their subpatterns. These propagation rules allow equivalent representations of an architecture to participate consistently in the same structural hierarchy. The predicates `nesy:alignedWith` and `nesy:categoryAlignedWith` have a different purpose. The former is a symmetric object property used to record explicit correspondences between design-pattern instances, whereas the latter is an annotation property used to document expert-established correspondences between taxonomy categories. Neither predicate entails `nesy:equivalentPattern`. Thus, an alignment indicates that two resources are related or comparable, but it does not assert that they have the same architectural structure. This distinction prevents conceptual mappings between taxonomies from generating unintended structural equivalences.

Let G be a NeSy-KG and let α be an RDF statement. We write $G \models_{\mathcal{R}} \alpha$ when α can be derived from G by applying the rule set $\mathcal{R}$ defined in Table 4, together with the RDFS and OWL entailments provided by the underlying RDF store. The deductive closure of G is defined as:

$$\text{Closure}_{\mathcal{R}}(G) = \{\alpha \mid G \models_{\mathcal{R}} \alpha\}. \tag{1}$$

The closure contains the statements explicitly represented in the NeSy-KG and all statements inferred through the ontology and the NeSy-specific rules. The reasoning process therefore enriches the graph with implicit pattern hierarchies and architectural equivalences without changing the definitions or alignments of the taxonomy categories.

Example. Consider a NeSy-KG containing the following statements:

$$(\texttt{ex:EmbeddingPattern1}, \texttt{nesy:subPatternOf}, \texttt{ex:RMLTPattern1})$$
$$(\texttt{ex:RMLTPattern1}, \texttt{nesy:equivalentPattern}, \texttt{ex:SymbolicToNeuroPattern1})$$

The first statement asserts that `ex:EmbeddingPattern1` is structurally contained in `ex:RMLTPattern1`. The second asserts that `ex:RMLTPattern1` and `ex:SymbolicToNeuroPattern1` represent equivalent architectural structures. Applying EP5 yields:

$$(\texttt{ex:EmbeddingPattern1}, \texttt{nesy:subPatternOf}, \texttt{ex:SymbolicToNeuroPattern1})$$

Assume that the graph additionally contains:

$$(\texttt{ex:SymbolicToNeuroPattern1}, \texttt{nesy:subPatternOf}, \texttt{ex:NeSyArchitecture1})$$

By applying SP2, the following statement is inferred:

$$(\texttt{ex:EmbeddingPattern1}, \texttt{nesy:subPatternOf}, \texttt{ex:NeSyArchitecture1})$$

The reasoning process therefore reveals that the embedding pattern is an indirect component of `ex:NeSyArchitecture1`, even though this relationship was not explicitly asserted. Moreover, the inference remains independent of the taxonomy categories used to classify `ex:RMLTPattern1` and `ex:SymbolicToNeuroPattern1`. The example illustrates how the axiomatic system supports structural comparison and traversal across alternative representations of NeSy architectures while keeping taxonomy alignment separate from architectural equivalence.

5. Evaluation of Representation and Reasoning Capabilities

We evaluate whether the proposed NeSy ontology and knowledge graph can represent heterogeneous Neuro-Symbolic classifications and support structural reasoning over design-pattern instances. The evaluation addresses the following research questions: **RQ1)** Can the NeSy-KG represent NeSy taxonomies, taxonomy categories, design patterns, and systems within a unified semantic model? **RQ2)** Can the NeSy-KG explicitly represent correspondences among taxonomy categories and design-pattern instances without conflating conceptual alignment with architectural equivalence? **RQ3)** Can the proposed reasoning framework derive implicit structural hierarchies and architectural equivalences

Table 5
NeSy-KG statistics before and after applying the revised reasoning rules.

Metric	Explicit Graph	Deductive Closure
Explicit triples	*2584*	*2584*
Inferred triples	0	*3264*
Total triples	*2584*	*5848*
Expansion ratio	1.00×	*2.26*×
Inferred `nesy:subPatternOf`	—	*1008*
Inferred `nesy:equivalentPattern`	—	*2256*

among design-pattern instances? These research questions distinguish two complementary capabilities. RQ1 and RQ2 assess the representational expressiveness of the ontology and the NeSy-KG, whereas RQ3 assesses the deductive consequences of the axiomatic system introduced in Section 4.

Evaluation Configuration. We evaluate the proposed framework using a corpus of 93 Neuro-Symbolic clinical decision-support systems collected by Kierner et al. [9] and Ng et al. [11]. Their architectural archetypes and Boxology representations were obtained from the T4B-KG [12]. Based on these resources, we constructed a NeSy-KG containing the analyzed systems, their architectural design patterns, and the categories defined by the Kautz, Kierner, and van Bekkum taxonomies.

The NeSy-KG distinguishes taxonomy-level and pattern-level correspondences. Expert-established correspondences between taxonomy categories are represented through the annotation property `nesy:categoryAlignedWith`. Correspondences between design-pattern instances are represented through `nesy:alignedWith`, whereas structural containment and architectural equivalence are represented through `nesy:subPatternOf` and `nesy:equivalentPattern`, respectively. Thus, an explicit alignment does not automatically entail architectural equivalence.

The NeSy-KG was deployed in GraphDB. The eight axioms presented in Table 4 were implemented as custom inference rules. These rules provide reflexive and transitive closure for `nesy:subPatternOf`, reflexive, symmetric, and transitive closure for `nesy:equivalentPattern`, identification of equivalence through mutual structural containment, and preservation of subpattern relationships across equivalent patterns. GraphDB additionally applies the selected RDF, RDFS, and OWL entailment regime. All evaluation resources, including the NeSy ontology, the NeSy-KG, the GraphDB rules, and the evaluation scripts, are available in the FigShare repository[1].

5.1. Representation Analysis

The first part of the evaluation examines whether the NeSy-KG can represent the three taxonomies without collapsing their conceptual distinctions. The NeSy ontology models Kautz's, Kierner's, and van Bekkum's proposals as separate taxonomy entities, while their categories are represented as specialized subclasses of `nesy:TaxonomyCategory`. NeSy architectures and their constituent design patterns are represented as instances of the corresponding ontology classes. This organization enables taxonomy definitions, taxonomy-category classes, design-pattern instances, and NeSy architectures to coexist within a unified knowledge graph while preserving their distinct semantic roles. Furthermore, taxonomy-category classes belonging to different taxonomies are not connected using `owl:equivalentClass`, preventing independently defined classifications from being unintentionally merged. These observations demonstrate that the NeSy-KG provides a unified yet semantically faithful representation (**RQ1**). The second part of the representation analysis evaluates how the NeSy-KG captures interoperability across taxonomies. Correspondences between taxonomy-category classes are explicitly documented using the annotation property `nesy:categoryAlignedWith`, which records expert-established conceptual relationships without introducing logical equivalence. At the design-pattern level, `nesy:alignedWith` represents interpretative semantic correspondences between pattern instances classified according

[1] https://figshare.com/s/6653f09e68afc5307846

to different taxonomies, whereas `nesy:equivalentPattern` is reserved for pattern instances that represent the same architectural organization despite originating from different modeling perspectives. This distinction allows the NeSy KG to represent different strengths of correspondence while restricting formal reasoning to structurally equivalent patterns. Overall, the NeSy-KG separates conceptual taxonomy alignment from architectural reasoning by distinguishing taxonomy-category correspondences, pattern-level semantic alignments, and formal architectural equivalences. Consequently, architectures described using different NeSy taxonomies can be compared through their structural design patterns while preserving the semantics of the original taxonomies (**RQ2**).

5.2. NeSy Reasoning Analysis

The reasoning analysis evaluates the structural knowledge inferred by the axiomatic system. The GraphDB rules operate exclusively on design-pattern instances connected through `nesy:subPatternOf` and `nesy:equivalentPattern`. They do not derive new `nesy:categoryAlignedWith` or `nesy:alignedWith` statements. Consequently, all inferred triples represent structural consequences rather than automatically discovered taxonomy correspondences. The rules for `nesy:subPatternOf` derive reflexive and transitive subpattern relationships. For example, if one design pattern is a subpattern of a second pattern, and the second is a subpattern of a third, transitivity infers that the first is also a subpattern of the third. Mutual subpattern relationships additionally imply structural equivalence between the corresponding design-pattern instances. The rules for `nesy:equivalentPattern` derive its reflexive, symmetric, and transitive closure. They also propagate structural relationships across equivalent design-pattern instances. Consequently, subpattern relationships asserted for one representation are automatically inferred for all structurally equivalent representations. Table 5 reports the numbers of explicit and inferred triples after reconstructing the NeSy-KG with the revised ontology and executing the GraphDB rules. The analysis distinguishes inferred `nesy:subPatternOf` statements from inferred `nesy:equivalentPattern` statements. The resulting deductive closure reveals implicit pattern hierarchies and architectural equivalences that are not explicitly asserted in the graph. An increase in `nesy:subPatternOf` statements reflects the discovery of indirect structural decompositions, whereas an increase in `nesy:equivalentPattern` statements results from reflexivity, symmetry, transitivity, mutual subpattern relationships, and equivalence propagation (**RQ3**). Overall, the evaluation demonstrates two complementary contributions. First, the NeSy ontology and NeSy-KG provide a unified yet semantically differentiated representation of taxonomies, taxonomy-category classes, design-pattern instances, and NeSy architectures. Second, the axiomatic system enriches this representation by deriving implicit structural knowledge. The framework therefore supports interoperability through explicit semantic correspondences while enabling controlled structural reasoning without treating all cross-taxonomy correspondences as logical equivalences.

6. Related Work

Table 6 presents a structured comparison of 8 prior works across 14 features organized into two major categories — *Architectural Patterns* and *Integration Patterns* within the neuro-symbolic AI landscape. The works span from [15] and [1] as earlier foundational contributions, through [3], [5], and [8] as more structurally oriented frameworks, to more recent works by [16], [10], and [12]. A foundational pillar to structuring the design space of NeSy systems comes from the notion of design patterns represented in software and ontology engineering. Gangemi et al. [15] introduced ODPs as task-oriented ontological components with explicit design documentation. This pattern-oriented thinking established a reusable vocabulary for describing structured knowledge systems and has been adopted as a methodology in neuro-symbolic system design. van Bekkum et al. [5] proposed a set of modular design patterns for hybrid systems, defining them as elementary and compositional patterns, demonstrating that a large number of hybrid system architectures can be described by composing only a small set of such patterns. Moreover, ten Teije et al. [8] proposed a refined set of compositional patterns for NeSy AI, validated

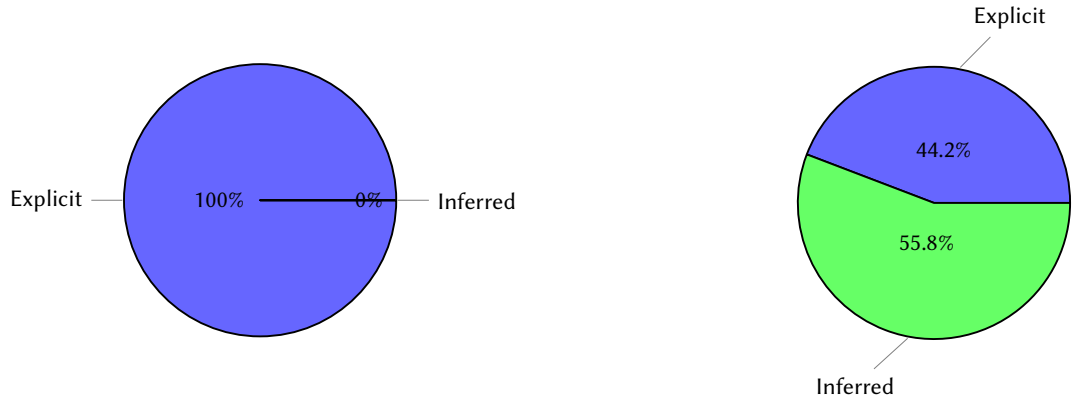


Figure 4: The composition of the NeSy-KG before and after closure is computed. Before reasoning, the graph contains only explicitly asserted triples. After deductive closure, the graph expands from 2,584 explicit triples (blue color) to 5,848 total triples through the derivation of 3,264 inferred statements (green color). Consequently, explicit knowledge represents approximately 44.2% of the resulting graph, while inferred knowledge accounts for 55.8%. The comparison illustrates the substantial contribution of deductive reasoning in enriching the NeSy-KG with implicit semantic relationships.

over recent scientific literature in knowledge and ontology engineering. Alnusair et al. [3] proposed an approach that uses semantic reasoning over OWL representations of source code to infer the presence of GoF design pattern instances on open-source application frameworks. This work demonstrates how ontology-based formalization can be used not only to specify patterns but also to validate whether concrete systems conform to them. More recently, Ellis et al. [10] introduced a composable glyph-based visual notation to facilitate multidisciplinary understanding and communication of AI system architectures. Similarly, Bendler et al. [12] introduced Tool4Boxology, a semantic toolbox that models architectural patterns for NeSy systems, enabling both the construction and semantic validation of NeSy architectures over knowledge graphs—going beyond visual notation towards machine readability. The classification of neuro-symbolic systems according to their mode of integration has been a persistent concern in the field. Bader et al. [1] analyzed surveys of neural-symbolic integration, presenting a classification of systems according to their logic representational abilities within neural networks. The aforementioned classification, organized along eight dimensions grouped into three aspects, was later contrasted with Kautz's taxonomy [7], proposing six categories for NeSy AI systems reflecting different degrees of integration between the neural and symbolic traditions. However, these taxonomic efforts remained largely descriptive. van Bekkum et al. [5] addressed this gap by proposing a modular, composable design pattern language—the boxology—grounded in a structured taxonomy of processes and data structures, enabling hybrid systems to be described, compared, and composed of elementary patterns. Further, Waltersdorfer et al. [16] identified SWeML as a distinct NeSy sub-area and conducted a systematic mapping study characterizing the space of such systems, their integration patterns, and research trends across 497 annotated papers. This work is complemented by Fajar et al. [6], which applied these system patterns specifically to characterize how semantic web and machine learning components are combined in practice across a broad range of published systems. These approaches provide design patterns [5, 8, 12] and taxonomies [7, 9] for describing NeSy systems, but they do not explicitly represent or reason over the relationships among them. In contrast, several older works show partial support (∼) or no coverage (—) across multiple integration dimensions, particularly in semantic violation detection and open or reusable design. Overall, the table illustrates a clear progression toward more holistic, semantically grounded, and practically validated neuro-symbolic frameworks in recent years, with [12] standing out as the most feature-complete contribution relative to the evaluated criteria. However, such frameworks do not provide reasoning axioms that enable reasoning across different taxonomies and represent them as *equivalent patterns* or a pattern can be *sub-pattern* of another. Our work addresses this gap through a NeSy ontology and a NeSy-KG that formalize taxonomies, design patterns, and their alignments, enabling the discovery of implicit cross-taxonomy relationships.

Table 6
Comparison of prior works across architectural and integration pattern features in neuro-symbolic AI. ✓: supported, ∼: partially supported, — : not addressed.

Feature	Gangemi & Presutti (2009)	Alnusair & Zhao (2010)	Bader & Hitzler (2005)	Van Bekkum et al. (2021)	ten Teije & van Harmelen (2023)	Waltersdorfer et al. (2023)	Ellis & Shimizu (2024)	Bendler et al. (2025)
Architectural Patterns								
Ontology-based pattern definition	✓	✓	—	∼	✓	∼	✓	✓
NeSy-specific architectural patterns	—	—	—	✓	✓	✓	✓	✓
Compositional/modular patterns	✓	—	—	✓	✓	✓	✓	✓
Visual/graphical notation	—	—	—	✓	✓	✓	✓	✓
Formal constraint validation (SHACL)	—	—	—	—	—	∼	—	✓
Machine-actionable pattern encoding	—	✓	—	—	—	✓	—	✓
Knowledge graph representation	—	—	—	—	—	✓	—	✓
Integration Patterns								
Taxonomy of NeSy integration types	—	—	∼	∼	∼	∼	∼	∼
Neural–symbolic coupling description	—	—	✓	✓	✓	✓	✓	✓
Semantic Web/ontology integration	✓	✓	—	—	∼	✓	✓	✓
Validated against real systems	∼	✓	✓	✓	✓	✓	—	✓
Semantic violation detection	—	✓	—	—	—	∼	—	✓
Domain application (clinical/biomedical)	—	—	—	—	✓	∼	—	✓
Open/reusable	—	—	—	—	—	✓	∼	✓

7. Conclusions and Future Work

This paper introduced the NeSy ontology, the NeSy Knowledge Graph (NeSy-KG), and an axiomatic reasoning framework for representing Neuro-Symbolic (NeSy) taxonomies, architectural design patterns, and NeSy systems within a unified semantic framework. Unlike previous approaches that implicitly merge heterogeneous taxonomies, the proposed ontology explicitly distinguishes taxonomies, taxonomy categories, and architectural design patterns, thereby separating conceptual taxonomy alignment from structural reasoning. The proposed axiomatic system operates over design-pattern instances, enabling the inference of implicit pattern hierarchies and architectural equivalences while preserving the semantic independence of the underlying taxonomies. The experimental evaluation demonstrates that the NeSy ontology and NeSy-KG can consistently represent multiple NeSy taxonomies and their associated systems, while the reasoning framework automatically derives implicit structural relationships among architectural design patterns. These results indicate that the proposed framework provides a semantic foundation for comparing, querying, and analyzing heterogeneous NeSy architectures without conflating conceptual correspondences with logical equivalence. We expect this work to facilitate the systematic organization, reuse, and evolution of Neuro-Symbolic architectures and to contribute toward shared semantic resources for the growing NeSy research community.

Future work will focus on extending the ontology and the NeSy-KG with recently proposed Neuro-Symbolic taxonomies and conceptual frameworks, including the definitions proposed by Desmet et al. [17], Lappin [18], and Schmid et al. [19]. We also plan to investigate AI-assisted methods for automatically extracting NeSy architectures from scientific publications, identifying architectural design patterns, discovering structural correspondences among patterns, and continuously constructing and enriching NeSy knowledge graphs at scale. Finally, we intend to exploit the resulting knowledge graphs to support architecture recommendation, comparative analyses of Neuro-Symbolic systems, and the systematic study of the evolution of NeSy design patterns across scientific domains.

Acknowledgements

Work at TIB was supported by the “Leibniz Best Minds: Programme for Women Professors,” through funding of the “TrustKG-Transforming Data in Trustable Insights” project (Grant P99/2020), and by the Lower Saxony Ministry of Science and Culture (MWK) with funds from the Volkswagen Foundation’s zukunft.niedersachsen program (CAIMed - Lower Saxony Center for AI and Causal Methods in Medicine; GA No. ZN4257). Mahsa Forghani was supported by the DigiStrucMed Program (Else Kröner

Promotionskolleg Hannover – „Digitale Transfomation in der Medizin"), funded by Hannover Medical School and Else Kröner-Fresenius Foundation (2020_EKPK.20).

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT (GPT-4, OpenAI) and Grammarly in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] S. Bader, P. Hitzler, Dimensions of neural-symbolic integration - A structured survey, CoRR abs/cs/0511042 (2005). URL: http://arxiv.org/abs/cs/0511042. arXiv:cs/0511042.

[2] J. Dietrich, C. Elgar, A formal description of design patterns using OWL, in: 16th Australian Software Engineering Conference (ASWEC 2005), IEEE Computer Society, 2005, pp. 243–250. URL: https://doi.org/10.1109/ASWEC.2005.6. doi:10.1109/ASWEC.2005.6.

[3] A. Alnusair, T. Zhao, Using ontology reasoning for reverse engineering design patterns, in: S. Ghosh (Ed.), Models in Software Engineering, Workshops and Symposia at MODELS 2009, volume 6002 of *LNCS*, Springer, 2009, pp. 344–358. URL: https://doi.org/10.1007/978-3-642-12261-3_32. doi:10.1007/978-3-642-12261-3_32.

[4] V. Lakshmanan, S. Robinson, M. Munn, Machine learning design patterns: solutions to common challenges in data preparation, model building, and MLOps, first edition ed., O'Reilly, Sebastopol, CA, 2020.

[5] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, A. ten Teije, Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases, CoRR (2021).

[6] F. Ekaputra, M. Llugiqi, M. Sabou, A. Ekelhart, H. Paulheim, A. Breit, A. Revenko, L. Waltersdorfer, K. Farfar, S. Auer, Describing and organizing semantic web and machine learning systems in the swemls-kg (2023), URL: http://arxiv. org/abs/2303.15113 (????).

[7] H. A. Kautz, The third AI summer: AAAI robert s. engelmore memorial lecture, AI Mag. 43 (2022) 93–104. URL: https://doi.org/10.1609/aimag.v43i1.19122. doi:10.1609/AIMAG.V43I1.19122.

[8] A. ten Teije, F. van Harmelen, Architectural patterns for neuro-symbolic AI, Frontiers in Artificial Intelligence and Applications, IOS Press, United States, 2023, pp. 64–76. doi:10.3233/FAIA230135, publisher Copyright: © 2023 The authors and IOS Press. All rights reserved.

[9] S. Kierner, J. Kucharski, Z. Kierner, Taxonomy of hybrid architectures involving rule-based reasoning and machine learning in clinical decision systems: A scoping review, Journal of Biomedical Informatics 144 (2023) 104428. URL: http://dx.doi.org/10.1016/j.jbi.2023.104428. doi:10.1016/j.jbi.2023.104428.

[10] A. Ellis, B. Dave, H. Salehi, S. Ganapathy, C. Shimizu, EASY-AI: sEmantic And compoSable glYphs for representing AI systems, in: F. Lorig, J. Tucker, A. Dahlgren Lindström, F. Dignum, P. Murukannaiah, A. Theodorou, P. Yolum (Eds.), Frontiers in Artificial Intelligence and Applications, IOS Press, 2024. URL: https://ebooks.iospress.nl/doi/10.3233/FAIA240187. doi:10.3233/FAIA240187.

[11] C. Ng, A. ten Teije, F. van Harmelen, A boxology-based analysis of design patterns for neuro-symbolic medical decision making systems 15734 (2025). doi:10.1007/978-3-031-95838-0_33.

[12] J. E. Bendler, Y. Chudasama, M. Forghani, E. Iglesias, D. Purohit, J. Roney, A. ten Teije, F. van Harmelen, M. Vidal, Tool4boxology: A semantic toolbox for constructing and analysing neuro-symbolic architectures, in: The Semantic Web - 23rd European Semantic Web Conference, ESWC, 2026. doi:10.1007/978-3-032-25159-6_11.

[13] M. A. Musen, The protégé project: a look back and a look forward, AI Matters 1 (2015) 4–12. URL: https://doi.org/10.1145/2757001.2757003. doi:10.1145/2757001.2757003.

[14] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A. N. Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. F. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, CoRR abs/2003.02320 (2020). URL: https://arxiv.org/abs/2003.02320. arXiv:2003.02320.

[15] A. Gangemi, V. Presutti, Ontology Design Patterns, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 221–243. URL: http://link.springer.com/10.1007/978-3-540-92673-3_10. doi:10.1007/978-3-540-92673-3_10.

[16] L. Waltersdorfer, A. Breit, F. J. Ekaputra, M. Sabou, A. Ekelhart, A. Iana, H. Paulheim, J. Portisch, A. Revenko, A. ten Teije, F. van Harmelen, Semantic web machine learning systems: An analysis of system patterns, in: Compendium of Neurosymbolic Artificial Intelligence, 2023. URL: https://api.semanticscholar.org/CorpusID:264290034.

[17] L. D. Smet, L. D. Raedt, Defining neurosymbolic ai, 2025. URL: https://arxiv.org/abs/2507.11127. arXiv:2507.11127.

[18] S. Lappin, Neuro-symbolic models of ai, Journal of Applied Logics 12 (2025) 1631–1642.

[19] E. van Krieken, S. Badreddine, R. Manhaeve, E. Giunchiglia, Uller: A unified language for learning and reasoning, in: Neural-Symbolic Learning and Reasoning: 18th International Conference, NeSy 2024, Barcelona, Spain, September 9–12, 2024, Proceedings, Part I, Springer-Verlag, Berlin, Heidelberg, 2024, p. 219–239. URL: https://doi.org/10.1007/978-3-031-71167-1_12. doi:10.1007/978-3-031-71167-1_12.