Semantically Coherent Business Architecture Models

Integrating Capabilities, Value Streams, and Business Objects

Anonymous

Abstract

In the face of increasing organizational complexity and continuous strategic change, business architecture models are expected to provide a coherent view of how strategy translates into value-creating activities. However, widely used frameworks such as TOGAF struggle to support this coherence as core elements such as capabilities, value streams, and business objects are modeled in isolation, with unclear semantics and weak structural alignment.

We address this gap by developing a Capability-Object-Value Ontology (COVO) and a set of modeling constraints that enforce semantic coherence. Our approach, grounded in the Unified Foundational Ontology (UFO), establishes two core principles: (1) a semantically integrated triad of capabilities, value streams, and business objects, unified by a value commitment, and (2) recursive coherence, ensuring that the same rules apply at every level of granularity. This provides (1) a foundation for models that are semantically sound (2) at every level of detail, enabling architects to seamlessly and reliably zoom between different levels of abstraction. The ontology and modeling constraints are illustrated by comparing a model of common, unconstrained practices with a constraint-compliant model, and are further demonstrated by a real-world example from the Dutch energy sector.

Keywords

Enterprise modeling, Business architecture, Capability modeling, Value streams, Business objects, OntoUML

1. Introduction

In an era of continuous change, business architecture models are essential to provide a stable foundation for an organization, with business capability models [1] serving as a cornerstone of this approach. Their strength comes from operating at the level of a *reference architecture* [2], which is intended to provide a common language for a shared understanding of the business, independent of its specific, evolving *implementation.*

However, this promise of a coherent business architecture language is often not fulfilled. A reference architecture is only effective if its constituent parts are integrated, yet current approaches such as TOGAF [3] treat core perspectives—notably capabilities, value streams, and business object models—in isolation. This fragmentation breaks the common architecture language at its core, hindering the holistic oversight needed to prevent inconsistencies.

For example, if business architecture models do not explicitly show that different value stream stages rely on the same capability, organizations miss opportunities to reuse existing resources, such as systems, staff, or procedures, leading to redundant investments and inconsistent customer experiences. Likewise, when capabilities are not clearly linked to the business objects they are meant to transform, it becomes difficult to determine each capability's value contribution and align responsibilities. This ambiguity extends to the ownership of the data describing these objects, as accountability for data is naturally connected to the capability responsible for transforming the object. In an era where data are a crucial asset for business intelligence and AI, this lack of clarity directly undermines an organization's ability to govern data quality and capitalize on the value of its data assets. These issues weaken the foundation for investment decisions and ultimately threaten organizational cohesion, underscoring the need for coherent enterprise models that support strategic alignment and informed decision-making [4].

To restore the integrity of this architectural language, we argue that a coherent integration of these perspectives is essential. Therefore, our contribution provides an ontological foundation and a set of modeling constraints that enforce the coherence of a central architectural triad: a business capability

Anonymous

as the organization's potential (*how*) to transform business objects (*what*) within a value stream that defines the purpose of this transformation (*why*). Specifically, we develop the Capability-Object-Value Ontology (COVO), based on the Unified Foundational Ontology (UFO) and formalized in OntoUML [5, 6]. The accompanying constraints ensure that the business architecture models remain semantically coherent across perspectives and granularity levels, thus laying a conceptual groundwork that supports practical application in modeling languages such as ArchiMate [7]. We illustrate our contribution by comparing the structural consequences of unconstrained versus constrained modeling and by a real-world capability model from the Dutch energy sector (i.e., NBility model) that conforms to our constraints.

The remainder of this paper is structured as follows. We begin by discussing related work (Section 2) and outlining our methodological approach (Section 3). Subsequently, we present the ontological analysis in Section 4, the resulting COVO ontology with its constraints in Section 5, and an illustrative application in Section 6. We conclude with a discussion of implications and future work (Section 7).

2. Related Work

While capability models are widely used for strategic alignment [1, 8], mainstream frameworks like TOGAF [3] and BIZBOK [9] lack an explicit semantic basis for integrating them with value streams and business objects. Recent formalization efforts, such as OMG's Business Architecture Core Metamodel [10], still lack a deep ontological foundation or the explicit constraints needed to enforce coherence.

Ontological analysis using UFO [5] offers a path towards semantic clarity. Calhau et al. [11] analyzed the capability concept as a UFO `disposition`. Building on their work and insights on value from the Common Ontology of Value and Risk (COVER) [12], we introduce COVO to create a coherent model of all three elements, thus addressing this gap.

3. Methodology

Our methodology consists of three sequential steps. First, we performed an **ontological analysis** to clarify the semantics of business capabilities, value streams, and business objects. We used TOGAF [3] as a pragmatic representation of mainstream practice, which we then refined and anchored using foundational insights from UFO [5], COVER [12], and the work of Calhau et al. [11]. Second, we **formalized** these concepts in OntoUML [6], resulting in COVO. Finally, we used COVO as a basis to **formulate modeling constraints**. These constraints were formalized in the Ampersand rule language [13], which allowed us to automatically verify their correctness and rigorously test their robustness by attempting to construct invalid counterexample models. We then illustrated their practical utility in two ways: first, through a comparative analysis of abstract models representing constrained versus unconstrained practices, and second, through a real-world example using the NBility model [14].

4. Ontological Analysis

Our ontological analysis clarifies the semantics of the three core constructs (business object, business capability, and value stream) by grounding them in UFO [5, 6].

While our focus is on these three elements, the rigor of UFO forces us to 'ground' them by establishing their existential dependencies. This grounding begins with a crucial insight into the nature of the business object. Our analysis conceptualizes them as a UFO `role mixin` [15, 12]. Consider a telecommunications provider that defines a *Customer Product* business object. The `role` aspect of this concept emphasizes that a classification is context-dependent. The classification of a smartphone as a *Customer Product* is not intrinsic; it only acquires this specific role when the organization assigns it to fulfill a commitment to a customer. Within the warehouse, the same physical object may play a different role, such as *Inventory Item*.

This insight has a profound consequence. In UFO, a role cannot exist in isolation; it requires a context. Our analysis revealed that this context is provided by a **value commitment** of the organization to a stakeholder. Without this commitment, there is no foundation for defining relevant business objects or purposeful capabilities.

The `mixin` aspect signifies that the role can be played by entities of fundamentally different kinds. For example, the provider may make distinct value commitments to supply a physical smartphone (a physical good) and to provide a telephony plan (a service). Despite their different ontological nature, the organization may treat both as instances of the *Customer Product* role. The UFO `role mixin` provides the formal mechanism to unify these disparate entities under a single business-relevant concept.

This commitment-centric view allows us to precisely define the fulfillment mechanism. A **business capability** is the organization's reusable potential (a UFO `disposition` [11]) to transform business objects in order to fulfill a value commitment. It represents a deliberate grouping of potential behaviors that the organization chooses to manage as a single, reusable unit. The **value stream**, in turn, is the manifestation of this capability in action (a `perdurant` [6]) enacting this transformation to realize the promised value.

This reusability is the core power of a capability. For example, a provider may define a single capability to *Manage Warehouse & Stock*. This unit of potential can be manifested as the delivery of smartphones to customers, as part of one value stream, while simultaneously being enacted to provide technical equipment for its network infrastructure, as part of another. By managing this potential as a single, reusable unit, the organization creates opportunities for synergy and efficiency across these distinct value streams, all while fulfilling different commitments.

For effective governance, actual behavior (the value stream) must be unambiguously traceable to this potential (the capabilities) to enable learning and improvement. The many-to-many mappings between these elements, as common in practice, obscure this traceability and hinder accountability. To resolve this, we posit that each value stream stage is driven by one *primary* capability. This is the capability directly responsible for achieving the object transformation defined by the stage's goal (its "exit criteria" in TOGAF terms [16, p. 6]), while other capabilities may act in supporting roles. This principle, combined with treating value streams and stages as the same ontological type (a view consistent with ArchiMate [7, p. 53]), enables a recursive decomposition where accountability remains clear at every level of detail.

Together, these concepts form a coherent triad at the core of business architecture. A value commitment is the promise to a stakeholder to transform a **business object**. The **business capability** is the reusable potential to honor that promise. Finally, the **value stream** is the concrete enactment that satisfies the commitment. Their relationships are not only a matter of convention but of ontological precision: the promise to transform an object requires the potential to do so, which is then realized in action.

5. Ontology and Modeling Constraints

Section 5.1 presents the COVO ontology, formalized in OntoUML, which builds on the ontological analysis in Section 4 and forms the foundation for the modeling constraints in Section 5.2.

5.1. Capability-Object-Value Ontology (COVO)

Figure 1 presents COVO, which formalizes the conceptual insights from our analysis. Using OntoUML stereotypes, we make the ontological nature of each element explicit according to UFO, creating a precise and coherent semantic structure.

The ontology is anchored in the *Organization*, modeled as a `<<kind>>`, which serves as the bearer of capabilities. The **Business Capability** is modeled as a `<<mode>>`, the OntoUML stereotype for intrinsic properties such as UFO `dispositions`, formally capturing its nature as the organization's persistent and reusable potential. Its actual manifestation is the **Value Stream**, an `<<event>>` (a `perdurant`

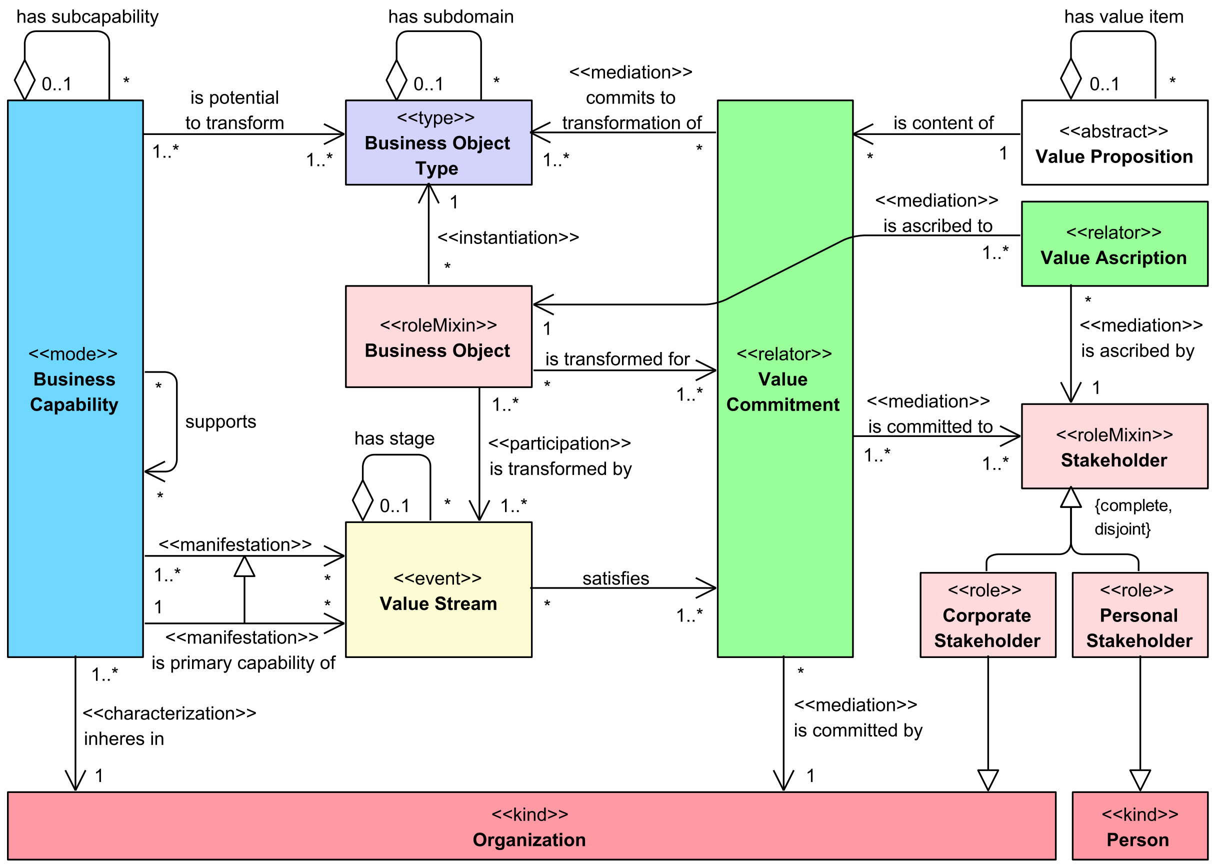

Figure 1: The proposed Capability-Object-Value Ontology (COVO)

in UFO), which is driven by exactly one *primary* capability to ensure unambiguous traceability from action back to potential.

The **Business Object** (`<<roleMixin>>`) represents a real-world entity playing a specific role defined by the context of a *Value Commitment*. This commitment is modeled as a `<<relator>>` that formally mediates the socio-economic promise between the *Organization* and a *Stakeholder*. This promise can be to create or enhance an object with positive value, or to mitigate one with negative value (a risk), reflecting the specific concerns of the stakeholder.

Finally, the ontology structures these core elements through three parallel hierarchical relations: *has subcapability*, *has stage*, and *has subdomain*. These relations are not mere refinements but are governed by specific design principles to enable the recursive refinement discussed earlier. For capabilities, we adopt a teleological refinement inspired by KAOS goal modeling [17]. In this view, a subcapability is not just a part, but a means to achieve the purpose of its parent capability. Furthermore, we define these hierarchical refinements to be mutually exclusive and collectively exhaustive (MECE) [18], ensuring that refinements are complete and non-overlapping. This structural discipline is what guarantees that model semantics remain consistent when zooming across different levels of granularity.

In summary, this logic can be captured in one integrative sentence: An *Organization* has a *Value Commitment* to its *Stakeholders* to manifest its **Business Capabilities** through **Value Streams** in order to transform **Business Objects**, thereby realizing its *Value Proposition*.

5.2. Modeling Constraints for Semantic Coherence

The COVO ontology provides a conceptual foundation for coherent business architecture models. To translate this foundation into actionable model guidance, we introduce a set of formal constraints. This list provides a self-contained specification of all constraints. It formalizes and extends the structural rules visible in Figure 1. These constraints serve as principles to ensure coherence, traceability, and governability as business architecture models are refined, acting as the specification for possible implementation in modeling languages such as ArchiMate. Some constraints are also formalized in first-order logic (FOL), where this adds precision or supports formal verification beyond what is evident from natural language.

The constraints are organized into two sets. The first governs the hierarchical refinement within each perspective, and the second ensures the alignment between them.

Constraints for Consistent Zooming

This first set of constraints (C1–5) governs the hierarchical structures that enable consistent zooming across different levels of granularity. We refer to elements in these hierarchies as parents, children, and ancestors.

- **C1. Unique parent**: Each element has at most one parent. *Rationale:* This ensures a single, unambiguous position for every element in the hierarchy.
- **C2. Acyclicity**: An element cannot be its own ancestor. *Rationale:* This prevents ill-defined, circular refinement structures.
- **C3. Consistent refinement depth**: All leaf elements (elements without children) must have the same number of ancestors. *Rationale:* This prevents incomplete levels of detail, which create both structural gaps and semantic ambiguity. An unbalanced model leaves the meaning of its most detailed elements unclear, as their defining peer group is incomplete.
- **C4. Upward coherence**: A non-hierarchical relationship between two elements requires a corresponding relationship between their parents (if any), provided the parents are distinct. *Exception:* The relationship does not need to be propagated if the parent elements are both primary capabilities within the same top-level value stream. *Rationale:* This ensures that low-level relationships are reflected at higher levels of abstraction. The exception allows lower-level support relations to remain implicit at higher levels. This aligns with the principle that value streams represent simplified views of value creation rather than detailed process models [16, p. 7]. In FOL (for the *transforms* relationship): $\forall o, o_p, c, c_p . \texttt{hasSubdomain}(o_p, o) \land \texttt{hasSubcapability}(c_p, c) \land \texttt{transforms}(c, o) \rightarrow \texttt{transforms}(c_p, o_p)$.
- **C5. Downward coherence**: A relationship between two parent elements requires that at least one pair of their respective children (if any) is also related. *Rationale:* This ensures that high-level relationships are grounded in more detailed, concrete relations.

Together, constraints C4 and C5 implicitly enforce a crucial principle: non-hierarchical relationships — the subject of the next set of constraints — may only occur between elements at the same granularity level. This prevents semantically incoherent configurations, such as directly linking a fine-grained capability (e.g., *Manage Customer Location*) to a coarse-grained object (e.g., *Customer* instead of *Customer Location*).

Constraints for Cross-Perspective Alignment

The second set (C6–10) ensures that the three perspectives remain aligned, forming the coherent triad established in our analysis.

- **C6. Capability impact**: Each business capability must transform exactly one business object. *Exception:* At the leaf-level, a business capability may transform multiple objects. *Rationale:* This ensures that every capability has a well-defined, non-overlapping impact on value creation. The exception prevents artificial fragmentation of what the business considers a single cohesive capability. It applies when a capability transforms multiple distinct business objects, each of which is justified by its role in triggering a different downstream process.

- **C7. Object relevance**: Each business object must be transformed by exactly one business capability. *Exception:* At the leaf-level, an object may be transformed by multiple capabilities. *Rationale:* This ensures clear relevancy and accountability for the object in value-creating activities. The exception prioritizes the conceptual stability of business objects as recognized by stakeholders. It avoids the need to decompose a familiar business object into numerous, fine-grained lifecycle states (e.g., *Submitted Order*, *Validated Order*), which would compromise the model's readability.
- **C8. Capability purpose**: Each business capability must either manifest as a primary capability in a value stream or support another capability that does. *Rationale:* This guarantees that all potential is ultimately linked to a value-creating purpose. In FOL: $\forall c.\exists c', s.\texttt{supports}^*(c, c') \wedge \texttt{isPrimaryCapabilityOf}(c', s)$.
- **C9. Traceability**: Each value stream must manifest exactly one primary business capability. *Rationale:* This constraint ensures traceability and governability. It establishes a clear, unambiguous link from value-creating action back to the accountable capability.
- **C10. Exclusive manifestation**: Each capability may manifest only once per top-level value stream as a primary capability. *Exception:* This constraint does not apply at the leaf-level. *Rationale:* This prevents a granularity mismatch between value streams and business capabilities. The exception at the leaf-level avoids the artificial discrimination between near-identical capabilities.

The leaf-level exceptions in rules C6, C7, and C10 should only be applied in exceptional cases and when justified.

6. Illustrative Application

This section demonstrates the practical impact of our constraint-based approach. We first use the abstract models in Figure 2 to compare the structural consequences of unconstrained versus constrained modeling. We then illustrate our constraints in a real-world context using a fragment from the NBility model in Figure 3.

6.1. Semantic Incoherence in Unconstrained Models

The top model in Figure 2 is a stylized representation of common, unconstrained practices based on BIZBOK and TOGAF. The bottom model illustrates our proposed, constrained approach. Because the constrained model respects granularity levels, its relationships can be conveyed through clean visual alignment and nesting (in addition to explicit *support* links). However, the unconstrained model requires connecting lines because its relationships arbitrarily cross these levels.

Although frameworks such as TOGAF and BIZBOK encourage hierarchical structures and relating perspectives, the top model in Figure 2 reveals how the lack of formal constraints leads to semantic incoherence. For example:

- Value stream stage `V2` is ambiguously linked to three capabilities, violating *Traceability* (C9) and obscuring accountability.
- A detailed capability (`C1.2.1`) maps directly to a high-level object (`O1.1`), violating *Upward and downward coherence* (C4 & C5).
- The perspectives are refined to different depths, violating *Consistent refinement depth* (C3).

The resulting model is semantically incoherent, failing to provide the reliable foundation required for architectural analysis.

6.2. Semantic Coherence Through Constraints: The NBility Model

The constrained model in Figure 2 avoids these issues. Its clarity stems from two core principles embedded in our approach.

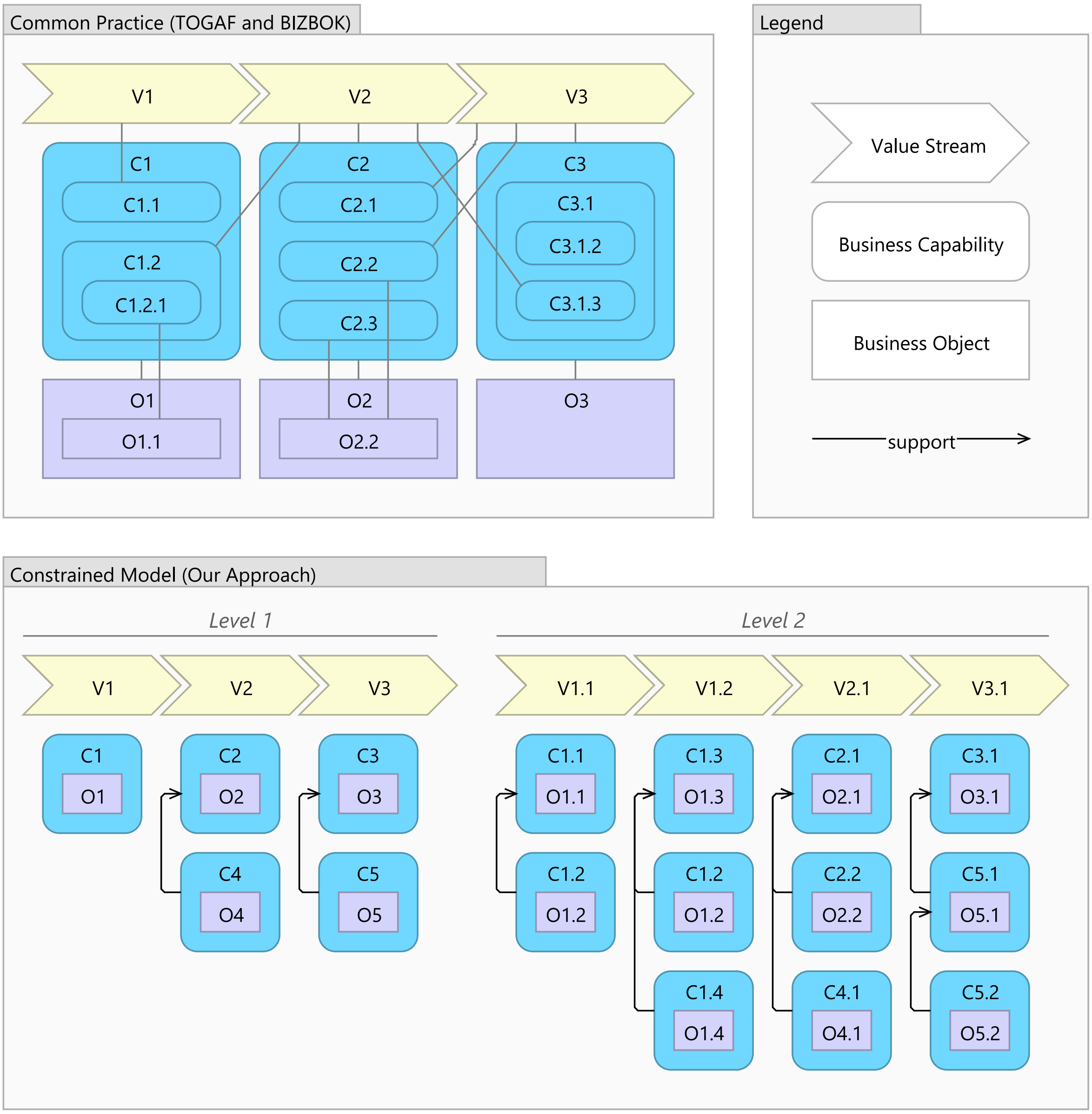

Figure 2: Comparison between modeling approaches

First, a **semantically integrated triad**. Unlike mainstream approaches that treat perspectives in isolation, our model is built on the logically necessary relationship between a business object, a capability, and a value stream, which are modeled as a coherent triad to fulfill a value commitment. Figure 3 illustrates this principle within a fragment of the real-world NBility model [14]. The figure's top diagrams present NBility's level 1 and 2 core capabilities and objects as hierarchical refinements, while the bottom diagram rearranges these same elements into one of NBility's value streams. The value stream *Install and change connections* is not merely mapped to capabilities; it is the concrete manifestation of them, transforming specific business objects to fulfill a clear purpose. For example, the stage *Integrate connection into energy grid* is a specific manifestation of the capability to *Expand, replace and renew energy grids*. This capability transforms the *Energy grid* to a desired state ('expanded with connection', not visualized), directly contributing to the value stream's proposition. This integration provides a holistic and unambiguous view of how value is created.

Second, **recursive coherence across granularity levels**. Our constraints, particularly C4 and C5, enforce a 'fractal-like' structure. As demonstrated in Figure 2, this ensures that the abstract model at

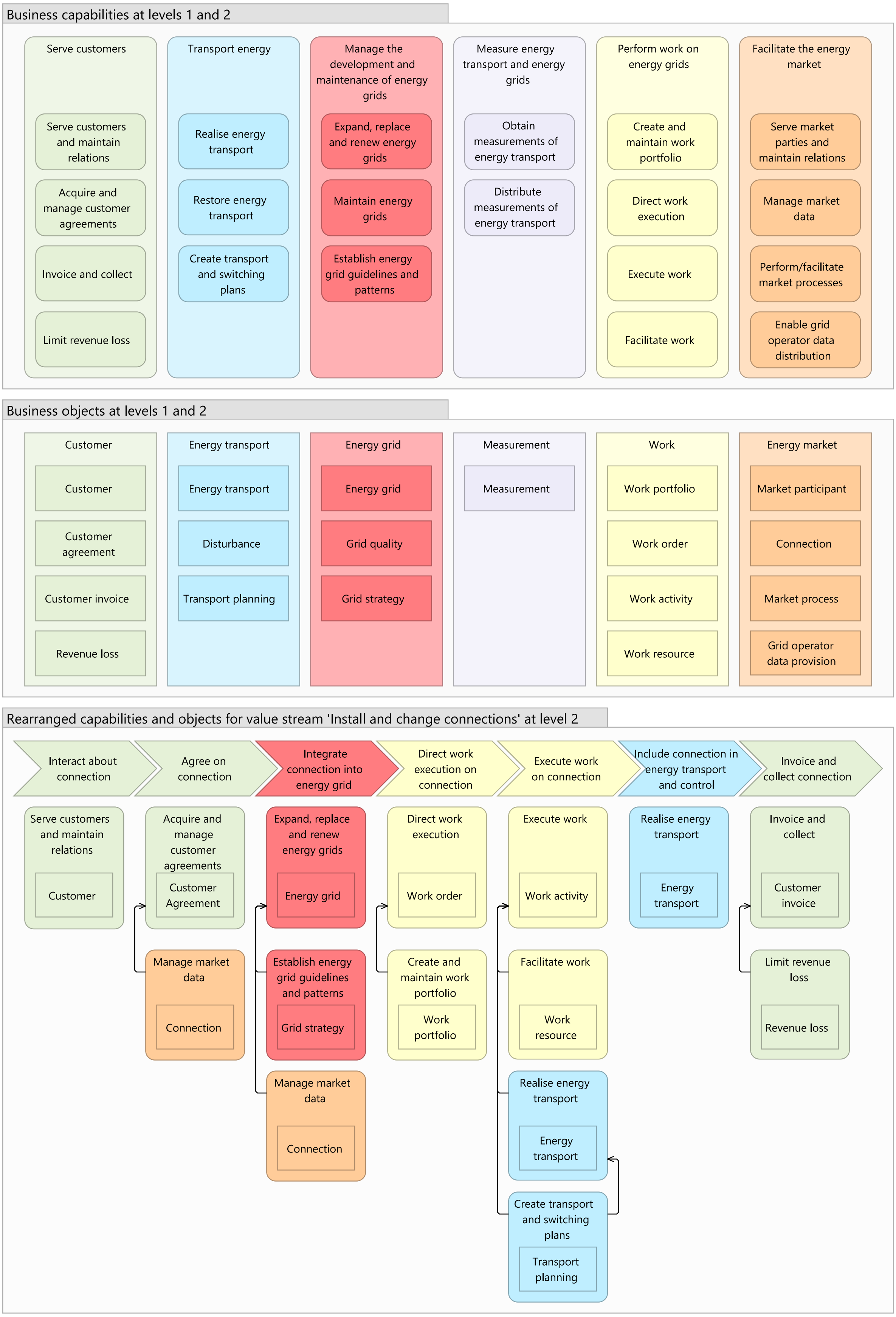

Figure 3: Overview of NBility level 1 and 2 core capabilities and objects (top), and their rearrangement into one of NBility's value streams (bottom)

Level 1 is fully and formally derivable from the refined model at Level 2. This principle guarantees that the semantic integrity of the triad holds true at every level of detail, enabling architects to seamlessly and reliably zoom between different levels of abstraction.

7. Discussion and Conclusion

This paper addressed the critical issue of semantic incoherence in business architecture modeling. Frameworks like TOGAF and BIZBOK promote well-intentioned modeling principles, but their lack of formal constraints often leads to models that are structurally inconsistent and semantically ambiguous.

Our primary contribution is a formal, constraint-based modeling approach that enforces semantic coherence. Building on UFO [5, 6] and COVER [12], we established two core principles: (1) a semantically integrated triad, and (2) recursive coherence across granularity levels. A key aspect of this contribution is the shift from an IT-centric to a business-centric perspective. In contrast to traditional enterprise data models (EDMs) [19, p. 105], which often lack guidance for defining business-relevant entities, our approach explicitly links each object to value streams and capabilities, thus clarifying why it matters for value creation. This provides a crucial advantage over traditional EDMs. Although EDMs define data entities, they often leave the business context and ownership ambiguous. Our approach makes this context explicit: by inextricably linking each business object to the transforming capability, we establish clear ownership of the corresponding data. This solves a fundamental challenge for data-driven organizations by providing a solid foundation for data governance, thereby creating a critical prerequisite for achieving the data quality required for AI applications and enabling strategic management of data as a valuable asset. Although constraint-based modeling may seem restrictive, in practice it provides guidance and clarity.

Our approach has several limitations. First, the evaluation in this paper is only illustrative. Second, manually applying the constraints is labor intensive and error-prone, requiring dedicated tool support to be effective in practice. Third, the generalizability of our approach beyond the energy sector requires further investigation. On a more fundamental level, our approach ensures the semantic coherence of a model, but does not yet help to select the right value streams that correspond to the value propositions chosen in the business model [20].

To tackle these limitations, future work should focus on bridging the gap between our conceptual foundation and modeling practice. A crucial first step is to translate the COVO ontology and its constraints into widely adopted languages such as ArchiMate [7]. Building on that translation, tool support can be developed to automate constraint validation, for which technologies like Ampersand [13] are promising. A complementary and high-impact avenue is the development of practical modeling aids. Research into guided patterns, in particular, could bridge the gap between formal coherence and strategic alignment, helping practitioners create business architectures that are not only internally consistent, but also genuinely aligned with the value propositions of the business model.

With these practical implementations in place, the benefits need to be empirically validated. This calls for case studies to assess whether our approach yields a higher return on modeling effort [21], which is achieving significantly improved model quality relative to the effort invested. Such a validation could also explore the added value of semantic rigor in automated settings by comparing the quality of models generated by Large Language Models (LLMs) with and without our constraints; recent benchmarking work provides a useful context for such experiments [22]. Ultimately, this research path can pave the way for the creation of business architecture models that provide the clarity and semantic coherence required for rapidly evolving and structually complex enterprises.

References

[1] P. Aleatrati Khosroshahi, M. Hauder, S. Volkert, F. Matthes, M. Gernegroß, Business capability maps: Current practices and use cases for enterprise architecture management, Proceedings of the 51st Hawaii International Conference on System Sciences 51 (2018) 4603–4612.

[2] R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole, M. Bone, The concept of reference architectures, Systems Engineering 13 (2010) 14–27. doi:10.1002/sys.20129.
[3] The Open Group, Togaf® standard, 10th edition, 2022. URL: https://publications.opengroup.org/c220.
[4] U. Frank, Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges, Software & Systems Modeling 13 (2014) 941–962. doi:10.1007/s10270-012-0273-9.
[5] G. Guizzardi, Ontological foundations for structural conceptual models, 2005.
[6] G. Guizzardi, A. Botti Benevides, C. M. Fonseca, D. Porello, J. P. A. Almeida, T. Prince Sales, Ufo: Unified foundational ontology, Applied ontology 17 (2022) 167–210. doi:10.3233/AO-210256.
[7] The Open Group, Archimate 3.2, 2022. URL: https://publications.opengroup.org/c226.
[8] S. Kotusev, A. Alwadain, Modeling business capabilities in enterprise architecture practice: the case of business capability models, Information Systems Management 41 (2024) 201–223. doi:10.1080/10580530.2023.2231635.
[9] Business Architecture Guild, A guide to the business architecture body of knowledge, version 14.0, 2025. URL: https://www.businessarchitectureguild.org/page/BIZBOK.
[10] Object Management Group (OMG), Business architecture core metamodel (bacm) version 1.0, 2024. URL: https://www.omg.org/spec/BACM/1.0.
[11] R. F. Calhau, J. P. A. Almeida, G. Guizzardi, Ontological analysis of advanced capability modeling in archimate: A first step towards language revision, in: International Conference on Enterprise Design, Operations, and Computing, Springer, 2024, pp. 82–100. doi:10.1007/978-3-031-79059-1_6.
[12] T. P. Sales, F. Baião, G. Guizzardi, J. P. A. Almeida, N. Guarino, J. Mylopoulos, The common ontology of value and risk, in: Conceptual Modeling: 37th International Conference, ER 2018, Xi'an, China, October 22–25, 2018, Proceedings 37, Springer, 2018, pp. 121–135. doi:10.1007/978-3-030-00847-5_11.
[13] S. Joosten, E. Roubtsova, Enterprise modeling with conventions, in: International Symposium on Business Modeling and Software Design, Springer, 2023, pp. 56–73. doi:10.1007/978-3-031-36757-1_4.
[14] Netbheer Nederland, Nbility model, 2025. URL: https://nbility-model.github.io/.
[15] N. Guarino, On the semantics of ongoing and future occurrence identifiers, in: Conceptual Modeling: 36th International Conference, ER 2017, Valencia, Spain, November 6–9, 2017, Proceedings 36, Springer, 2017, pp. 477–490. doi:10.1007/978-3-319-69904-2_36.
[16] The Open Group, Togaf® series guide: Value streams, 2022. URL: https://publications.opengroup.org/g178.
[17] A. v. Lamsweerde, Requirements engineering: from system goals to UML models to software specifications, John Wiley & Sons, Ltd, 2009.
[18] B. Minto, The pyramid principle: logic in writing and thinking, Pearson Education, Harlow, 2009.
[19] Dama International, DAMA-DMBOK: Data management body of knowledge, Technics Publications, LLC, 2017.
[20] A. Osterwalder, Y. Pigneur, Business model generation: a handbook for visionaries, game changers, and challengers, John Wiley & Sons, 2010.
[21] H. A. Proper, G. Guizzardi, Modeling for enterprises;: Let's go to rome via rime, in: 15th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modelling, PoEM 2022, CEUR, 2022, pp. 4–15.
[22] I. Samarasekara, M. Bandara, F. Rabhi, B. Benatallah, Benchmarking llms for business architecture modelling with hierarchical capability maps, in: International Conference on Advanced Information Systems Engineering, Springer, 2025, pp. 37–53. doi:10.1007/978-3-031-94569-4_3.