



Modeling Civil Engineered Systems

Ontological Modelling

submission date: 10.10.2025

Author

Mehmet Emin Suntay

Civil Systems Engineering
Institute of Civil Engineering | Technische Universität Berlin

1. Objective

This study aims to develop an ontological knowledge model that comprehensively defines the structural, material, and geotechnical properties of reinforced concrete cantilever retaining walls, as well as the locations where the structure is used. The main objectives of the study are as follows. Firstly, to prepare the ground for establishing logical connections between engineering data by bringing together the parameter relationships defined in the literature, such as structural geometry, concrete strength, backfill type, and friction angle, into a single knowledge model. Second, to ensure the comparability of alternative designs. By representing two different design configurations, Option 1 (lower safety limit) and Option 2 (upper safety limit) within the same ontological structure, the structural and geotechnical differences between these systems can be analyzed using consistent data types and logical queries. Currently, the System Ontology does not directly evaluate these two systems as “safe” or “unsafe”; it only objectively defines the parameter differences. Finally, the goal is to support information sharing and digital engineering processes. By providing a reusable and interoperable information structure compatible with BIM platforms and decision support systems, it aims to enable more effective use of engineering knowledge in pre-construction and post-construction processes.

The need for this study is as follows. In the traditional retaining wall design process, structure, ground, and material data are stored in different sources, and this information is usually compiled manually by engineers. This situation makes it difficult to maintain data integrity and increases the risk of errors in the design process. Furthermore, the lack of organization of structural design criteria and the data used under a single source prevents the systematic and easy comparison of alternative designs. A similar problem persists in the later stages of the structure's life cycle: it is often impossible to transfer construction and design data to maintenance and operation processes, and basic project information is lost over time or cannot be integrated with other digital systems.

2. Scope

This ontology covers the geometric, material, and functional properties of reinforced concrete cantilever retaining walls. The model defines basic physical components such as the stem, heel, toe, base plate, and backfill soil, as well as the structural relationships between them (`hasComponent`, `hasSubstructure`, `hasSuperstructure`). The ontology represents design parameters that affect the wall's behavior, such as the safety factor, drainage system, concrete strength, friction angle, geometric ratios, etc., and enables the comparison of design alternatives through these parameters. However, the model does not include detailed structural analyses, dynamic ground behavior, or cost calculations; it focuses solely on the semantic organization of this data and its logical interconnection. And it is intended for use by engineers, project managers, and maintenance teams involved in the design, construction, and management of RC retaining walls.

3. Literature Review

Physical Components: Goh (1993) examined the behavior of T-type reinforced concrete cantilever retaining walls through the interaction of the stem, base plate, heel, toe, backfill, and foundation soil. The study demonstrated the effects of these components on stiffness, rotation, and soil pressure distribution. In the ontology, this structure is modeled using the `hasComponent`, `hasSubstructure`, and `hasSuperstructure` relationships.

Material and Design Parameters: Akış (2023) defined the engineering parameters for concrete, reinforcing steel, and backfill soil for the cost optimization of reinforced concrete retaining walls. The study assumed a concrete compressive strength of 30 MPa, a reinforcement yield strength of 420 MPa, and backfill friction angles between 30° and 40°. These data established reference values for the data properties (`hasConcreteStrength`, `hasFrictionAngle`, `hasFactorOfSafety`, `hasDrainageSystem`) in the

ontology. Furthermore, the upper and lower limits of the design variables (e.g., heel length, toe length, stem thickness) were taken from this source and used in the creation of the Option 1–2 scenarios.

Functional Purpose and Usage: Brooks & Nielsen (2021) defined the primary functions of reinforced concrete retaining walls as resisting ground pressure, transferring loads to the foundation, and maintaining stability through drainage. They specified application areas such as highways, bridge approaches, basements, and industrial platforms. This information provided the basis for the RetainingWallUse classes in the ontology (e.g., BridgeAbutmentsUse, HighwayEmbankmentsUse, RetainingYardsUse).

4. Ontology Structure

The steps used in this ontology were developed based on the methodology proposed by Noy and McGuinness (2001).

- (1) The scope and domain were defined with the aim of ensuring that the ontology represents the physical, material, and functional components of reinforced concrete cantilever retaining walls and supports design comparisons.
- (2) Existing knowledge sources were reused; structural components from Goh (1993), material and geometric parameters from Akış (2023), and functional definitions from Brooks & Nielsen (2021) were integrated into the model.
- (3) Key terms have been listed and
- (4) These terms have been organized within a class hierarchy using “is-a” and “part-of” relationships
- (5) Class properties (object and data properties) have been defined,
- (6) Constraints have been added to these properties
- (7) In the final step, instances were created; RetainingWallOption1 and Option2 were modeled with different safety, drainage, and friction values.
- (8) The ontology was validated using the Pellet reasoner, which confirmed logical consistency and correct classification of design options under the designed wall class.

4.1 Class Hierarchy

This hierarchical classification defines the conceptual structure of the ontology, showing the division of reinforced concrete cantilever retaining walls into their physical, material, and functional components.

Reinforced_concrete_retaining_wall

- └─ **DesignedReinforcedConcreteRetainingWall**
- └─ RetainingWallOption1
- └─ RetainingWallOption2

RetainingWallDomain

- └─ **SubstructureRetainingWall**
- └─ BackfillSoilSubstructure
- └─ HeelSubstructure
- └─ ToeSubstructure
- └─ **SuperstructureRetainingWall**
- └─ StemSuperstructure

RetainingWallMainMaterial

- └─ ConcreteMaterial
- └─ Reinforcing_SteelMaterial
- └─ Backfill_SoilMaterial
- └─ OtherMaterials

RetainingWallUse

- └─ BridgeAbutmentsUse
- └─ HighwayEmbankmentsUse
- └─ RetainingYardsUse
- └─ RailwayEmbakmentsUse
- └─ BasementAndUnderUndergroundStructuresUse

4.2 Logical Axioms

Table.1 summarizes the semantic relationships between classes (e.g., Reinforced_concrete_retaining_wall), object properties (hasSubstructure, hasComponent, etc.), data properties (hasFactorOfSafety, hasDrainageSystem, etc.), and engineering design alternatives (RetainingWallOption1&2). Each axiom explains what engineering knowledge the ontological model represents and how this knowledge contributes to the system's internal reasoning processes.

Table 1: Logical Axioms of the Ontological Model (Representative Overview)

Logical Axiom	Effect in the Ontological Model
$\text{hasSubstructure} \sqsubseteq \text{hasComponent}$	Substructure is treated as a component of the retaining wall.
$\text{inverse}(\text{hasComponent}) = \text{isComponentOf}$	Allows a two-way relation between the wall and its component parts.
$\text{Transitive}(\text{hasComponent})$	Enables multi-level component reasoning
$\text{Domain}(\text{hasSubstructure}) = \text{Reinforced_concrete_retaining_wall}$	Restricts the use of hasSubstructure property to reinforced concrete retaining walls.
$\text{Range}(\text{hasSubstructure}) = \text{SubstructureRetainingWall}$	Specifies that the target of hasSubstructure must be a substructure class.
$\text{DisjointClasses}(\text{Reinforced_concrete_retaining_wall}, \text{RetainingWallDomain}, \text{RetainingWallUse}, \text{RetainingWallMainMaterial})$	Ensures that the wall class, its domain, use, and main material categories are mutually exclusive.
$\text{RetainingWallOption1} \sqsubseteq (\text{hasBackfillSoil value BackfillSoilOption1}) \sqcap (\text{hasHeel value HeelOption1}) \sqcap (\text{hasToe value ToeOption1})$	Defines Option 1 as a retaining wall variant with its specific backfill, heel, and toe components.
$\text{RetainingWallOption1} \sqsubseteq (\text{hasSubstructure some BackfillSoilSubstructure}) \sqcap (\text{hasSubstructure some DrainageSystemSubstructure}) \sqcap (\text{hasSubstructure some HeelSubstructure}) \sqcap (\text{hasSubstructure some ToeSubstructure}) \sqcap (\text{hasSuperstructure some StemSuperstructure})$	Ensures that Option 1 includes all required substructure elements and a stem as its superstructure.
Datatype definitions: hasFactorOfSafety, hasConcreteStrength, hasFrictionAngle, hasBaseThickness, hasToeLength : xsd:double; hasDrainageSystem: xsd:boolean	Defines numerical and Boolean data types for structural and geotechnical design parameters.
$\text{Reinforced_concrete_retaining_wall} \sqsubseteq \exists \text{hasSubstructure.SubstructureRetainingWall}$	Defines that each retaining wall must include a substructure.
$\text{Reinforced_concrete_retaining_wall} \sqsubseteq (\text{hasSubstructure some SubstructureRetainingWall}) \sqcap (\text{hasSuperstructure some SuperstructureRetainingWall}) \sqcap (\text{hasMainMaterial some RetainingWallMainMaterial})$	States that every RC retaining wall must have a substructure, a superstructure, and a main material.
$\text{DesignedReinforcedConcreteRetainingWall} \sqsubseteq \exists \text{hasMainMaterial.ConcreteMaterial} \sqcap \exists \text{hasMainMaterial.Reinforcing_SteelMaterial}$	Connects the designed retaining wall class with its main materials (concrete and steel).
$\text{Domain}(\text{hasDrainageSystem}) = \text{DesignedReinforcedConcreteRetainingWall}$; Range = xsd:boolean	Restricts the drainage system property to designed wall instances, represented as Yes/No.

Logical Axiom	Effect in the Ontological Model
Reasoner inference example results: DesignedReinforcedConcreteRetainingWall \sqsubseteq (hasSubstructure some SubstructureRetainingWall) \sqcap (hasSuperstructure some SuperstructureRetainingWall) \sqcap (hasMainMaterial some RetainingWallMainMaterial) and RetainingWallOption1, RetainingWallOption2 \sqsubseteq DesignedReinforcedConcreteRetainingWall	Validating the ontology's logical coherence.
Annotation: rdfs:comment "An OWL ontology for reinforced concrete retaining wall design."	Describes the ontology's purpose and scope for documentation and clarity.

A complete version of the Logical Axioms Table, including all subclass definitions, inverse properties, and detailed component-level relations, is provided in Appendix

4.3 Data Properties

The data properties as shown in Table 2. defined in the ontology represent the fundamental numerical parameters used in retaining wall design. These parameters are classified as geometric ratios (stem height, base thickness, heel and cap lengths), material properties (concrete and steel strengths and safety factor), drainage conditions, and friction angle. The value ranges are taken from Flow (2023)'s experimental and optimization studies. Design options Option 1 – low safety limit and Option 2 – high safety limit are represented and exemplified with these parameters. Reasoner-based class inference, such as SafeWall, DrainedWall, and OptimalWall, has not been formally added to the ontology. Therefore, the system does not produce classification on individuals. However, since the data properties and class relationships are defined, the model maintains its logical consistency.

Table 2. Data properties and their defined data types

Property	Type	Description	Example Values (Option 1-2)
hasbasethickness	xsd:double	Base plate thickness (m)	0.56 – 0.30
hasconcretestrength	xsd:double	Concrete compressive strength (MPa)	30
hasreinforcementstrength	xsd:double	Reinforcement yield strength (MPa)	420
hasstemheight	xsd:double	Stem height (m)	8.0 – 3.0
hasstemwidth	xsd:double	Stem width (m)	0.2 – 0.3
hasheellength	xsd:double	Heel length (m)	0.64 – 0.30
hastoelength	xsd:double	Head length (m)	0.80 – 0.693
hasfrictionangle	xsd:double	Backfill internal friction angle (°)	30 – 40
hasfactorofsafety	xsd:double	Safety factor (FS)	1.5 – 2.5
hasdrainagesystem	xsd:boolean	Drainage system presence	0 / 1

4.4 Ontological Map

The ontology model shown in Figure 1 semantically relates the structural components, material types, and application areas of reinforced concrete retaining walls. Classes such as SubstructureRetainingWall, SuperstructureRetainingWall, and BackfillSoilSubstructure define the physical sections of the wall, while classes such as ConcreteMaterial, Reinforcing_SteelMaterial, and Backfill_SoilMaterial represent the relevant material types. Object properties such as hasSubstructure, hasSuperstructure, and hasMainMaterial establish relationships between these components, enabling connections to the structural and material properties of individuals such as RetainingWallOption1 and

RetainingWallOption2. This structure allows the ontology to show a holistic analysis of design alternatives, material properties, and areas of use.

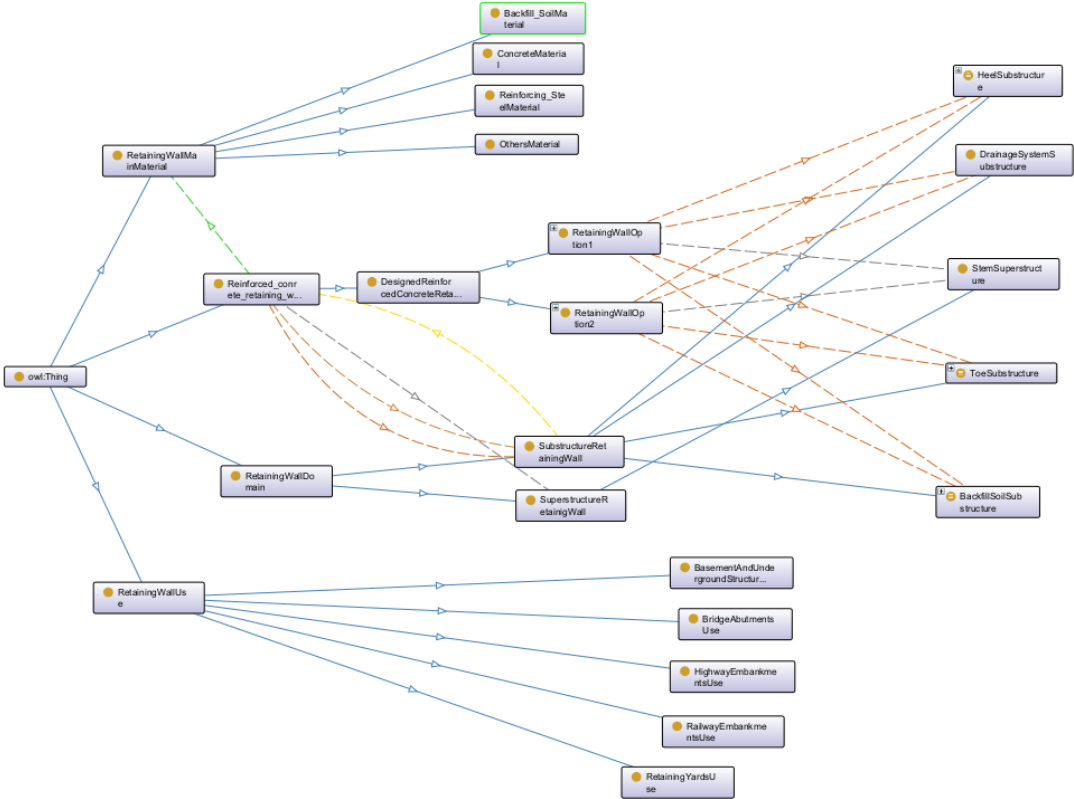


Figure1. Ontological Map of RC Cantilever Retaining Wall

5. Engineering Examples

Scenario 1:

Cracks have been observed in the existing reinforced concrete cantilever retaining wall along a railway corridor due to water accumulation behind the wall body. The physical and ground parameters of the wall have been transferred to the ontology to determine the cause of the problem.

Use Case

Based on the parameters entered into the ontology (hasDrainageSystem = 0, hasFrictionAngle = 30°), the system defines that there is no drainage layer and that the fill material has a low friction angle. In accordance with defined engineering rules, this combination is classified as a potential instability condition. Following this diagnosis, engineers can test possible solution scenarios in a digital environment: adding a drainage system (hasDrainageSystem = 1) and selecting fill material with a higher friction angle (hasFrictionAngle = 40°). In this way, the ontology analyzes the performance of the existing structure using semantic rules, reveals the cause of instability, and enables the evaluation of alternatives.

Scenario 2:

During the construction of a reinforced concrete retaining wall in a new infrastructure project, discrepancies were noticed between the material test results obtained on site and the approved design values. The measurements for concrete strength and reinforcement class were below the values specified in the project. To determine whether this situation was repeated in other walls in the project, the data for all structural components was transferred to the ontology.

Use Case:

The measurements obtained on site are entered into the system as the data properties hasConcreteStrength, hasReinforcementStrength, and hasBaseThickness and are automatically compared with the approved design values. The system identifies deviating values and flags them as

non-conformities. For example, a stem thickness 10% lower than design or a low reinforcement strength class. This accelerates the quality management process and makes it more error-free.

Scenario 3:

In a reinforced concrete retaining wall project, several years after the structure entered service, the maintenance team wanted to evaluate the wall's drainage performance. However, most of the data created during the project phase, such as concrete class, reinforcement ratio, foundation type, or drainage system, could not be accessed because it was stored on different platforms. Therefore, the maintenance team had to rely on incomplete or estimated information about the current state of the structure.

Use Case:

The RC Retaining Wall Ontology combines all data from the design, construction, use, and maintenance phases under a single semantic structure to prevent such information loss.

The wall's design parameters, material composition, construction date, current condition, and past interventions are linked to each other through relationships such as `hasDesignParameter`, `hasMaterialComposition`, `hasConstructionDate`, `hasConditionState`, and `hasMaintenanceActivity`.

6. Conclusion

The RC Retaining Wall Ontology developed within the scope of this study brings together the structural, material, and usage characteristics of reinforced concrete retaining walls under a single semantic information model. The model ensures data integrity, comparability, and information sharing by linking the data generated during the design phase. As a result, this study has established a fundamental structure that reduces information loss throughout the life cycle and provides sustainable decision support for digital engineering processes.

7. References

- Akış, E. (2023). *Optimum Cost Prediction of Reinforced Concrete Cantilever Retaining Walls*. Department of Civil Engineering, Atılım University.
- Brooks, H., & Nielsen, J. P. (2021). *Basics of Retaining Wall Design* (10th ed.). Civil & Geotechnical Engineers.
- Goh, A. T. C. (1993). Behavior of Cantilever Retaining Walls. *Journal of Geotechnical Engineering*, 119(8), 1250–1266.
- Noy, N. F., & McGuinness, D. L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05.

8. Appendices

Table 3. Logical axioms and their semantic effects

Logical Axiom	Effect in the Ontological Model
$\text{hasSubstructure} \sqsubseteq \text{hasComponent}$	Substructure is treated as a component.
$\text{hasBackfillSoil} \sqsubseteq \text{hasSubstructure}$	Backfill soil is part of the substructure.
$\text{hasHeel} \sqsubseteq \text{hasSubstructure}$	Heel is a substructure element.
$\text{hasToe} \sqsubseteq \text{hasSubstructure}$	Toe is a substructure element.
$\text{hasSuperstructure} \sqsubseteq \text{hasComponent}$	Superstructure is treated as a component.
$\text{inverse}(\text{hasComponent}) = \text{isComponentOf}$	Allows two-way relation between wall and parts.
$\text{inverse}(\text{hasSubstructure}) = \text{isSubstructureOf}$	Enables reverse query for substructure.
$\text{inverse}(\text{hasSuperstructure}) = \text{isSuperstructureOf}$	Enables reverse query for superstructure.
$\text{inverse}(\text{hasMainMaterial}) = \text{isMainMaterialOf}$	Links wall and its materials both ways.
$\text{inverse}(\text{hasUse}) = \text{isUseOf}$	Links wall and its use area in both directions.
$\text{Transitive}(\text{hasComponent})$	Allows multi-level component relations to be inferred.
$\text{Transitive}(\text{isComponentOf})$	Enables reasoning over nested component hierarchies in the inverse direction.
$\text{Domain}(\text{hasSubstructure}) = \text{Reinforced_concrete_retaining_wall}$	The property is used only by RC retaining walls.
$\text{Range}(\text{hasSubstructure}) = \text{SubstructureRetainingWall}$	The target is always a substructure class.
$\text{DisjointClasses}(\text{Reinforced_concrete_retaining_wall}, \text{RetainingWallDomain}, \text{RetainingWallUse}, \text{RetainingWallMainMaterial})$	Ensures that the wall class, its domain, use, and main material categories are mutually exclusive.
$\text{DisjointClasses}(\text{SubstructureRetainingWall}, \text{SuperstructureRetainingWall})$	Prevents overlap between substructure and superstructure component classes.
$\text{RetainingWallOption1} \sqsubseteq (\text{hasBackfillSoil value BackfillSoilOption1}) \sqcap (\text{hasHeel value HeelOption1}) \sqcap (\text{hasToe value ToeOption1})$	Defines Option 1 as a retaining wall variant with its own backfill, heel, and toe components.
$\text{RetainingWallOption1} \sqsubseteq (\text{hasSubstructure some BackfillSoilSubstructure}) \sqcap (\text{hasSubstructure some DrainageSystemSubstructure}) \sqcap (\text{hasSubstructure some HeelSubstructure}) \sqcap (\text{hasSubstructure some ToeSubstructure}) \sqcap (\text{hasSuperstructure some StemSuperstructure})$	Ensures that Option 1 includes all required substructure parts and a stem as its superstructure.
$\text{RetainingWallOption1} \sqsubseteq (\text{hasSuperstructure some StemSuperstructure})$	Ensures that Option 1 also includes a stem element as its superstructure part.
$\text{RetainingWallOption2} \sqsubseteq (\text{hasBackfillSoil value BackfillSoilOption2}) \sqcap (\text{hasHeel value HeelOption2}) \sqcap (\text{hasToe value ToeOption2})$	Corresponding component bindings for Option 2.
$\text{RetainingWallOption2} \sqsubseteq (\text{hasSubstructure some BackfillSoilSubstructure}) \sqcap (\text{hasSubstructure some DrainageSystemSubstructure}) \sqcap (\text{hasSubstructure some HeelSubstructure}) \sqcap (\text{hasSubstructure some ToeSubstructure}) \sqcap (\text{hasSuperstructure some StemSuperstructure})$	Ensures that Option 2 also includes the full set of substructure components and a stem as its superstructure.

Logical Axiom	Effect in the Ontological Model
ToeSubstructure) \sqcap (hasSuperstructure some StemSuperstructure)	
RetainingWallOption2 \sqsubseteq (hasSuperstructure some StemSuperstructure)	Ensures that Option 2 also includes a stem as its superstructure element.
Datatype definitions: hasFactorOfSafety, hasConcreteStrength, hasFrictionAngle, hasBaseThickness, hasReinforcementStrength, hasStemHeight, hasStemWidth, hasHeelLength, hasToeLength : xsd:double; hasDrainageSystem: xsd:boolean	Defines numeric and Yes/No data types for structural and geotechnical parameters.
Reinforced_concrete_retaining_wall \sqsubseteq \exists hasSubstructure.SubstructureRetainingWall	Every retaining wall must have a substructure.
Reinforced_concrete_retaining_wall \sqsubseteq (hasSubstructure some SubstructureRetainingWall) \sqcap (hasSuperstructure some SuperstructureRetainingWall) \sqcap (hasMainMaterial some RetainingWallMainMaterial)	Every RC retaining wall must have a substructure, a superstructure, and a main material.
SubstructureRetainingWall \sqsubseteq \exists isSubstructureOf.Reinforced_concrete_retaining_wall	Enables inverse reasoning from a substructure to its retaining wall.
DesignedReinforcedConcreteRetainingWall \sqsubseteq \exists hasMainMaterial.ConcreteMaterial \sqcap \exists hasMainMaterial.Reinforcing_SteelMaterial	Links the designed wall class with its main materials (concrete + reinforcing steel).
DesignedReinforcedConcreteRetainingWall \sqsubseteq \exists hasUse.RetainingWallUse	Connects each designed retaining wall to a defined use type (bridge abutment, highway embankment, etc.).
Domain(hasDrainageSystem) = DesignedReinforcedConcreteRetainingWall Range(hasDrainageSystem) = xsd:boolean	Restricts drainage system property to wall instances, represented as binary (No /Yes).
Reasoner inference example results: DesignedReinforcedConcreteRetainingWall \sqsubseteq (hasSubstructure some SubstructureRetainingWall) \sqcap (hasSuperstructure some SuperstructureRetainingWall) \sqcap (hasMainMaterial some RetainingWallMainMaterial) and RetainingWallOption1, RetainingWallOption2 \sqsubseteq DesignedReinforcedConcreteRetainingWall	Validating the ontology's logical coherence.
Annotation: rdfs:comment "An OWL ontology for reinforced concrete retaining wall design."	Describes the ontology's purpose and scope for documentation and clarity.