

Schema Semantics

[Go up to Terms and Definitions](#)

Discussion

Schema Semantics defines meaning through information elements, structures, permitted values, relationships, cardinalities, constraints, validation rules, and representation contexts.

A [Schema](#) expresses semantic content when its element names, datatypes, constraints, relationships, and documentation preserve [traceability](#) to defined domain meaning. The schema constrains representation. The associated vocabulary, [conceptual model](#), rule set, [ontology](#), report model, or other authoritative semantic source establishes the intended meaning.

Schema Semantics differs from schema syntax. Schema syntax defines how the schema appears in a selected language or technology. Schema Semantics defines what the schema elements, constraints, and relationships mean in the domain.

Schema Semantics also differs from [datatype](#) semantics. Datatype semantics constrains value space, representation, and permitted operations. Schema Semantics constrains how information elements, datatypes, relationships, and rules combine within a defined representation context.

Schemas appear in many forms, including XML Schema, JSON Schema, [RDF Schema](#), SQL DDL, OpenAPI schemas, Avro schemas, Protobuf schemas, DDS IDL, and message specifications.

Schema semantic content includes:

- Information elements
- Element names
- Attributes
- Relationships
- Cardinalities
- Required elements
- Optional elements
- Permitted values
- Datatypes
- Value constraints
- Pattern constraints
- Range constraints
- Enumeration values
- Default values
- Structural rules
- Validation rules
- Representation contexts
- Traceability to the [Domain](#) meaning

A governed architecture preserves traceability from schema Semantics to the conceptual model, vocabulary, type system, report model, ontology, rule set, policy source, implementation artifact, or

conformance test that establishes or validates the intended meaning.

Definition

meaning defined through schema elements, structures, relationships, permitted values, constraints, and validation rules within a defined representation context

Source

DIDO Solutions usage is informed by schema-language practice, data modeling, information modeling, software architecture, semantic modeling practice, and model-driven architecture principles.

Note

Schema Semantics does not define the full meaning of a domain. A schema expresses selected semantic commitments in a representation context. Domain meaning remains governed by the conceptual model, vocabulary, ontology, rule set, report model, policy source, or other authoritative semantic source.

A schema can validate structure while still failing to preserve meaning. For example, a schema can verify that a value is a valid date, but it does not, by itself, determine whether the date represents a trade date, settlement date, effective date, or termination date.

Example

An FX trade schema defines selected information elements for a trade message.

Schema element	Datatype	Constraint	Schema semantic meaning
tradeIdentifier	string	required	identifies the reported trade
counterpartyLEI	string	required; LEI pattern	identifies a counterparty using a legal entity identifier
currencyPair	string	required; currency-pair pattern	identifies the ordered pair of currencies exchanged in the trade
notionalAmount	decimal	required; greater than zero	identifies the amount used to calculate settlement obligations
tradeDate	date	required	identifies the date on which the trade occurs
settlementDate	date	required	identifies the date on which settlement occurs
lifecycleState	string	enumeration	identifies the governed lifecycle state of the trade

The schema constrains the representation of the FX trade message. Schema Semantics arises from

the relationship between schema elements and the meanings of the domain they govern. The schema treats `tradeDate` and `settlementDate` as different information elements even though both use the same date datatype.

A validation rule that rejects a message when `settlementDate` precedes `tradeDate` adds rule semantics to the Schema Semantics. The schema identifies the fields and datatypes. The rule defines a domain constraint across those fields.

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:

<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:

<https://wiki.didosolutions.com/dido/05-semantics/01-kinds-of-semantics/12-schema-semantics>

Last update: **2026/07/18 12:33**

