

Rule Semantics

[Go up to Terms and Definitions](#)

Discussion

Rule semantics defines meaning through rules, conditions, constraints, decisions, obligations, prohibitions, permissions, derivations, validations, and consequences.

A rule expresses semantic content when it constrains what an artifact, system, actor, process, message, [report](#), [schema](#), [ontology](#), or implementation behavior means in a defined context. Rule semantics identifies the meaning of the condition, the required outcome, the prohibited outcome, the permitted outcome, and the consequence of rule satisfaction or violation.

Rule semantics differs from rule syntax. Rule syntax provides the form of the rule. Rule semantics defines what the rule means and how the rule constrains interpretation or behavior.

Rule semantics also differs from implementation logic. Implementation logic executes or enforces selected rules. Rule semantics establishes the meaning that implementation logic preserves, validates, derives, or enforces.

Rules appear in several forms, including natural-language business rules, decision tables, validation rules, policy rules, conformance rules, [Semantic Business Vocabulary Model \(SBVM\)](#) vocabulary constraints, [Standard Business Report Model \(SBRM\)](#) reporting rules, schema constraints, [OWL](#) axioms, runtime checks, and executable rule engines.

Rule-processing architectures express rule semantics through defined responsibilities. A policy administration point records or manages governed rules. A policy information point supplies facts, attributes, or context used during rule evaluation. A policy decision point evaluates rules and produces decisions. A policy enforcement point enforces decisions.

Rule semantic content includes:

- Rule subjects
- Rule conditions
- Rule conclusions
- Business vocabulary terms
- Fact types
- Obligations
- Prohibitions
- Permissions
- Constraints
- Derivations
- Decision inputs
- Decision outputs
- Validation outcomes
- Decision outcomes
- Exceptions

- Violation conditions
- Rejection reasons
- Evidence requirements
- Policy administration points
- Policy information points
- Policy decision points
- Policy enforcement points
- Traceability to semantic sources

A governed architecture preserves [traceability](#) from rule semantics to the [conceptual model](#), vocabulary, policy source, schema, ontology, report model, implementation artifact, or conformance test that expresses or validates the rule meaning.

Definition

meaning defined through rules, conditions, constraints, decisions, obligations, prohibitions, permissions, validations, and consequences within a defined context

Source

DIDO Solutions usage, informed by business rule practice, policy modeling, validation practice, conformance testing, semantic modeling practice, and model-driven architecture principles.

Note

Rule semantics is not equivalent to rule execution. A rule engine, validation function, policy decision point, policy enforcement point, or runtime check expresses selected rule semantics in executable or operational form. The governed rule source establishes the intended meaning.

A rule violation carries semantic meaning when the violated rule traces to a defined concept, business vocabulary, policy, requirement, report model, schema, ontology, lifecycle, or other authoritative semantic source.

Example

An FX Demo specification defines the following business rule:

A confirmed FX trade has a settlement date that does not precede the trade date.

The rule contains semantic content because it defines a valid relationship between the trade date, settlement date, and confirmed lifecycle state.

Rule element	Example value	Rule semantic meaning
Rule subject	ConfirmedFXTrade	the rule applies to trades in the confirmed lifecycle state
Business vocabulary terms	confirmed FX trade, settlement date, trade date	the rule uses governed business meanings from the SBVM or another authoritative vocabulary source
Condition	settlementDate precedes tradeDate	the trade contains an invalid temporal relationship
Prohibition	settlement date before trade date	the lifecycle interpretation is not permitted
Expected validation outcome	Reject	the system rejects the trade message or state transition
Rejection reason	Settlement date precedes trade date	the rejection records the semantic rule violation
Semantic source	FX trade lifecycle rule	the rule traces to the governed lifecycle meaning
Conformance interpretation	non-conforming trade lifecycle interpretation	the failed rule identifies a semantic non-conformance

The rule-processing architecture assigns responsibilities for the same rule:

Rule-processing role	Example component	Rule semantic responsibility
Policy administration point	FX Policy Console	records and governs the settlement-date rule
Policy information point	Trade Attribute Service	supplies tradeDate, settlementDate, and lifecycleState
Policy decision point	FX Validation Decision Service	evaluates the rule and returns Accept or Reject
Policy enforcement point	Commit Guard	prevents commitment of a trade state when the decision is Reject

The same rule appears as executable validation logic in an implementation:

```
if settlementDate < tradeDate:
    reject("Settlement date precedes trade date")
```

The code expresses rule semantics only because the implementation traces to the governed rule. Without traceability, the code represents an undocumented implementation decision rather than a governed semantic constraint.

© 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/11-rule-semantics>

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

