

Literate Modeling

2026-08-31
SimpleModeling

A SimpleModeling Model has three constituents that express different aspects of the same subject: the Object, Knowledge, and Literate Models. A Literate Model uses primarily human-comprehensible natural language to express subjects, purposes, situations, requirements, decisions, scenarios, intent, assumptions, and history. Conventional documents that explain Domain Model concepts, rules, and relationships are also important Literate Model constituents. Developers, non-developers, and generative AI share it, and it retains as part of the Model information that cannot, or should not, be expressed by reducing it to the Object or Knowledge Model.

A Story is the semantic content of what happens, comprising actors, purposes, situations, events, causality, and outcomes. A Narrative is a Literate Model that selects, orders, and explains that Story from a particular viewpoint, Purpose, and Context. It enables inductive understanding of meaning from concrete situations. A Use Case Scenario is a representative Narrative constructed from the user's perspective.

Before generative AI, developers read Scenarios and manually transformed them into analysis, design, and implementation. In the AI era, Use Case Scenarios—the core of requirements modeling—can be supplied directly to generative AI. Generative AI principally generates, updates, and manages the Object Model and CML, while people review visualized results, differences, and evidence and return approval or findings to the AI. The essential shift is that a requirements specification becomes an input that drives system development while preserving human judgment of meaning, rather than remaining only something people interpret.

1 Position in the Series

Part 5 treated the Object Model as a formal constituent shared by developers, programming languages, execution platforms, and generative AI. Part 6 treated the Knowledge Model as a constituent through which generative AI explores concepts, rules, semantic relationships, and evidence and explains them to people.

Part 7 examines the Literate Model. A Literate Model is neither supplemental explanation for a formal structure nor documentation written after implementation. Its natural-language content is itself part of the Model and becomes input to requirements, design, decisions, validation, and document composition.

Previous article: Knowledge Modeling for AI Collaboration

2 The Literate Aspect of a SimpleModeling Model

- Object Model: Executable formal Models and execution examples express structure and Behavior. They are shared by developers, programming languages, execution platforms, and generative AI.
- Knowledge Model: It expresses connections among concepts, rules, semantic relationships, and evidence. It is used primarily by generative AI, with people using it through AI interaction.
- Literate Model: It expresses natural-language-centered context, intent, requirements, decisions, and Scenarios. It is shared by developers, non-developers, and generative AI.

The three do not describe unrelated subjects; they divide responsibility for the required aspects of the same subject. Formal structure alone cannot fully retain who and what situation a requirement serves, why a decision was made, or which assumptions it depends on. A Literate Model keeps this information within the Model and relates it to the Object and Knowledge Models. Narratives and Use Case Scenarios are highlighted as important forms, but Vision, requirements, decision records, existing specifications, and conventional natural-language explanations of the Domain Model also constitute Literate Models.

3 Story and Narrative

A Story is semantic content comprising actors, purposes, situations, events, causality, and outcomes. A Narrative selects the elements required for a Purpose from that Story and orders and explains them according to a particular viewpoint and Context. In this sense, a Narrative functions as a Literate Model that expresses the meaning of change. The same Story can be organized into different Narratives for users, operators, and developers.

- Induction through Narrative: Derives meaning and patterns in requirements and decisions from concrete situations, events, and outcomes.
- Deduction through formal Models: Uses generalized rules and structures to check the conditions that individual Behavior must satisfy.

People often derive meaning more easily when a concrete Story is presented as a Narrative than when they follow only abstract rules deductively. One Narrative alone, however, cannot define every permitted Behavior. Using inductive Narratives and deductive formal Models together connects “what happens and why it matters” with “what must hold generally.”

4 Use Case Technology as a Direct Input to AI Development

Use Case technology is a core requirements-modeling approach that describes system Behavior through a Narrative constructed from the user’s perspective. A Use Case Scenario is a Narrative

that organizes one Story for achieving an Actor’s Goal as events, actions, branches, and successful or failed outcomes between the Actor and the subject system.

Traditionally, a Use Case Scenario was a requirements description that developers interpreted and transformed into analysis, design, and implementation. Generative AI can read the natural-language Scenario directly, consult related terms, rules, and Model elements, and generate, update, and manage the Object Model and CML (Cozy Modeling Language).

Actor’s Goal → Use Case Scenario → Knowledge Model construction by textus-bok → Generative AI generates, updates, and manages the Object Model and CML → Visualizations, differences, and evidence from textus-cbd-support → Human review → Return approval or findings to generative AI

“Direct input” does not mean that a correct implementation is produced automatically and unconditionally from a Scenario. It means that generative AI manages the Object Model while people evaluate correspondence to requirements, meaning, and impact and return approval or findings. The fundamental breakthrough is turning interpretation formerly held in developers’ heads into a collaborative process that can be visualized, traced, and approved.

5 From Literate Programming to Literate Models

Literate Programming treats a program and its explanation in one document and presents them in an order that supports human understanding rather than the computer’s execution order. Literate Modeling applies this idea to modeling: it is the activity and method of organizing Model information and explanation together in human-comprehensible form. A Literate Model is the subject and result described and organized through that activity.

Literate Programming → Literate Modeling: the activity and method applied to Models → Literate Model: its subject and result

CML is also a concrete practice of Literate Modeling. Formal elements in a CML document express an executable Object Model, while natural-language explanations, intent, and assumptions in the same document express Literate Model content. A CML document therefore spans both the Object and Literate Models. This does not mean that CML expresses the whole Literate Model. In the AI era, Vision, requirements, decisions, Scenarios, and other documentary information outside CML are also handled consistently as Literate Models.

6 Conditions for Becoming a Literate Model

Natural-language text alone does not make a document a Literate Model. At minimum, its subject and Purpose must be clear, its meanings and identities must be traceable, and it must connect to design, decisions, or validation.

Condition	Question
Subject and Purpose	What does it describe, and for what Purpose?
Meaning and identity	Which concept, requirement, decision, or Scenario does it denote?

Model correspondence	Which Model elements, rules, and evidence does it relate to?
Development use	How is it used for design, decisions, validation, or explanation?

7 Content Expressed by Literate Models

Content	Meaning
Subject	What is being described.
Purpose	What should be achieved or understood.
Situation	The context in which it applies.
Requirement	What must be satisfied.
Decision	What was chosen and why.
Scenario	How events proceed.
Intent	The intent behind a form or structure.
Assumption	What is treated as known or valid.
History	How understanding and decisions changed.

Not every item must be placed in one document. Literate Modeling selects the content required by the Purpose and relates it to the relevant terms, knowledge, and Object Model elements.

Literate Models do not consist only of purpose-specific documents such as Vision statements and Use Cases. Conventional explanatory documents that describe Domain Model concepts, rules, and relationships in ordinary human-readable form are also important constituents. Relating them to the formal Model gives people and generative AI a foundation for understanding domain meaning and evaluating differences.

8 Literate Models that Direct Development

Vision, Goal, and Context are also important Literate Models that direct development. A Vision states the desired future and reason for the project's existence. A Goal states intended outcomes, priorities, and success conditions. Context identifies stakeholders, scope, environment, constraints, and assumptions used in decisions.

Keeping these as Models lets people evaluate requirements, architectural decisions, priorities, and validation results against the project's Purpose and boundaries. Generative AI can also create candidates using the context that explains why a requirement is needed, rather than using an isolated requirement alone.

9 Sharing Context through SmartDox

SmartDox is the full-spec primary language for describing Literate Models, while Markdown is also allowed. SmartDox is a descriptive foundation that preserves natural-language readability while organizing documents through headings, paragraphs, lists, tables, diagrams, and term

references. Changes can be tracked as plain text, and Glossary identities connect Literate Model descriptions to other Model elements.

Writing something in SmartDox does not by itself make it a valid Model. Its subject, Purpose, meaning, identity, evidence, and approval state must be organized. textus-bok turns Literate Models into Knowledge Models and connects them to generative AI. The AI uses this context to generate, update, and manage the Object Model. People examine the meaning and validity of the results through the visualizations, differences, and evidence provided by textus-cbd-support, then return approval or findings to the AI.

10 Authoritative Sources for Documents

Developing requirements specifications, design explanations, review materials, user guides, and reference manuals as independent authoritative sources duplicates facts and creates inconsistencies with every update. Literate Modeling organizes background, Purpose, requirements, decisions, Scenarios, assumptions, and history as authoritative documentary information and composes or generates purpose- and audience-specific documents from the required sources.

This does not mean that every document is merely derived. A natural-language document that explains Domain Model concepts, rules, and relationships can itself be an authoritative Literate Model source referenced by other purpose-specific documents.

- User guide: Use Cases and Use Case Scenarios are its primary authoritative sources; Literate Models supply usage situations, purposes, terms, assumptions, and cautions.
- Reference manual: Program specifications and the executable Object Model are its primary authoritative sources; Literate Models supply background, Purpose, design intent, and usage context.
- Requirements and design explanation: Requirements, decisions, Object Models, and Knowledge Models are their primary authoritative sources; Literate Models supply Scenarios, rationale, assumptions, and history.

A generated document is not a new authoritative source. It is a View composed from the required Model elements for a particular Purpose and audience. Authoritative sources are updated, and documents are recomposed according to consistent organization rules.

11 Use Case Scenarios as an Important Example

A Use Case Scenario is an important example of a Literate Model. It presents a Story comprising an Actor's Goal, events, actions, branches, and outcomes as a user-centered Narrative and forms the core of a requirements specification. It is an important source for user guides and a direct input to Model creation and update in AI-assisted development.

Literate Models are not limited to Use Case Scenarios. Vision, Goals, Context, requirements, decision records, design intent, assumptions, and history, as well as conventional documents that

explain the Domain Model, are important Literate Models when they relate to Model meanings and identities and become inputs to design, understanding, or validation.

12 Connecting Scenarios to Executable Models

Jumping directly from a Use Case Scenario to CML would make it difficult to evaluate which Scenario statements correspond to which participants, responsibilities, Operations, Events, and State changes. SimpleModeling places a reviewable Use Case Realization Model between a Scenario and executable elements.

Use Case Scenario → Knowledge Model construction by textus-bok → Generative AI generates and updates the Use Case Realization Model and executable elements → Visualizations, differences, and evidence from textus-cbd-support → Human review → Return approval or findings to generative AI → Update CML

Generative AI extracts participants, Roles, responsibilities, Collaborations, and Interactions from Scenarios and principally generates, updates, and manages the Use Case Realization Model and executable elements such as Services, Operations, Events, and StateMachines. People evaluate whether the requirements' meaning and mappings are preserved and return approval or findings to the AI. The intermediate Model makes the AI's interpretation and Object Model updates traceable. Part 9 will examine the detailed Application Modeling that uses Collaborations, Interactions, and StateMachines.

13 A Compact Order-Confirmation Example

Examining the same subject, “confirm an order,” through the three constituents makes their respective roles clear.

- Literate Model: It expresses the buyer's Goal of confirming an order, inventory and payment conditions, and successful and failed Scenarios and outcomes.
- Knowledge Model: It expresses the meaning of “order confirmation,” business rules, evidence, sources, and mappings to related terms and Model elements.
- Object Model: It expresses approved meaning executably through formal structures such as Services, Operations, Events, and StateMachines.

The three descriptions are related through shared Glossary identities. textus-bok turns the Literate Model into a Knowledge Model and supplies it to generative AI, which generates, updates, and manages the Object Model. Through textus-cbd-support, people evaluate whether the Scenario's Goal and outcomes are preserved in the executable Model.

14 Summary

- A Story is the semantic content of what happens, while a Narrative is a Literate Model that organizes it from a particular viewpoint, Purpose, and Context. Vision, requirements, decision records, existing specifications, and natural-language documents explaining the Domain Model also constitute Literate Models.
- Literate Modeling is the activity and method of organizing natural-language Model information and explanation in human-comprehensible form.
- A Literate Model retains subjects, purposes, situations, requirements, decisions, Scenarios, intent, assumptions, and history.
- textus-bok turns Literate Models into Knowledge Models and connects them to generative AI, which principally generates, updates, and manages the Object Model.
- People review results through the visualizations, differences, and evidence provided by textus-cbd-support and return approval or findings to the AI.
- Literate Models are organized as authoritative sources for facts from which purpose-specific documents are composed.

15 Next

Part 8, “Domain Modeling,” will examine how the meanings, structures, rules, and boundaries of a problem domain are organized from Object, Knowledge, and Literate Models.