

From structure to behavior

AI Development Harness / Episode 4

1. Episode 3: static models

Order and product structures and relationships

2. Episode 4: dynamic models

Sequence, state transitions, and failure handling

3. Dynamic model harness

Constrain AI-generated and modified behavior

Business rules for behavior

1. Before confirming an order

Reserve inventory and authorize payment

2. While processing

Define waiting deadlines and retry conditions

3. After failure

Define cleanup and human handoff

CML rules in checking and execution

1. Define rules in CML

Model workflows and state machines

2. Check generated changes

Check models and supplementary implementation

3. Execute in Textus

Follow permitted progression and transitions

Domain and application models

1. Domain model

Order structures, relationships, and business rules

2. Application model

Workflows that realize use case goals

Workflow progression

1. Example order

O-101: one unit of P-1 for JPY 3,000

2. Confirmation sequence

Reserve inventory, authorize payment, confirm

3. Workflow rules

Branching, parallel work, waiting, completion

States and transition guards

1. Draft

Zero order lines allowed

2. Start confirmation

At least one order line required

3. Transition to confirmed

Inventory reserved AND payment authorized

Static structure and behavioral contracts

1. Static contracts

Identity, product references, and amount types

2. Operation contracts

Connect preconditions and postconditions

3. Commands and events

Requested changes versus established facts

Compensation after failure

1. Payment declined

Leave the order unconfirmed and release inventory

2. Compensation fails

Define retry and human-handoff conditions

Unknown outcomes and failures

1. Response timeout

Query the payment status

2. Waiting and retries

Set deadlines and retry limits

3. Duplicate-payment protection

Use idempotency; escalate unknown outcomes

Models as executable programs

1. Executable description

CML workflows and state machines

2. Execution platform

Textus interprets and executes the models

3. AI-supplied implementation

Check it against model types and contracts

User goals in use cases

1. Use case

Who wants to achieve what

2. Order-confirmation scenario

Reserve inventory, authorize payment, confirm

3. Requirement check

Does the behavior achieve the user's goal?

Requirements and realization

1. Requirements and realization

Connect use cases to application workflows

2. Business-rule correspondence

Trace to operations and state machines

3. Paths to check

Normal execution, rejection, and waiting

Examples and general rules

1. Inductive model: use case

Check requirements through concrete scenarios

2. Deductive model: dynamic model

Execute behavior under general rules

3. Examples to compare

Normal, declined payment, unknown outcome

Checking and semantic review

1. Mechanical checks

Types, references, and model connections

2. Review and execution checks

Requirement gaps and concrete results

3. Human judgment

Acceptable operations and failure responses

Model Up for stakeholder review

1. Model Up

Abstract an executable model for review

2. Example

Derive a flow diagram from a state-machine workflow

3. Review feedback

Preserve correspondence and revise the source model

AI judgment and rule-based execution

1. AI's role

Interpret requirements, propose models, select actions

2. Execution's role

Apply rules to inputs, events, and current state

Workflows as business operations

1. Business operation contract

Inputs, preconditions, outcomes, failure handling

2. Capability

An intermediate representation of what can be done

3. Execution

Associated operations, workflows, state machines

Managing behavioral changes

1. Start from the models

Revise requirements and behavioral rules

2. Check the impact

Run checks and revisit supplementary implementation

3. In-progress work

Define model-version and state migration rules

Dynamic Model Harness

Apply business rules to AI generation, changes, and execution

