

# Domain Modeling

2026-09-07  
SimpleModeling

ドメインモデリング (domain modeling) は、開発対象となる問題領域について、意味、構造、規則、境界をドメイン・モデル (domain model) として組織する活動です。SimpleModeling では、ドメイン・モデルをオブジェクト・モデル、知識モデル、文芸モデルと並ぶ第四の構成要素にはしません。同じ対象を表す三つの構成要素から、問題領域を理解し、判断し、開発するために必要なモデル要素を選び、対応付けた適用モデルとして扱います。

ドメイン・モデルは、整った図や一貫した用語だけで完成するものではありません。現実の業務に対して妥当であり、重要な規則と境界を正しく表している必要があります。そのため、ドメインエキスパート、開発者、生成 AI が、用語、根拠、形式構造、説明を共有し、モデルの妥当性を反復的に高めます。

## 1 シリーズにおける位置付け

第 5 回から第 7 回では、SimpleModeling モデルを構成する三つの側面を扱いました。

- オブジェクト・モデル (object model): 構造と振る舞いを形式的に表す実行可能なモデルと実行例
- 知識モデル (knowledge model): 概念、分類、意味関係、規則、根拠を生成 AI が探索、関連付け、説明するためのモデル
- 文芸モデル (literate model): 文脈、意図、要求、判断、シナリオなどを自然言語中心で表すモデル

第 8 回からは、三つの構成要素を開発上の目的に沿って組織します。ドメインモデリングは、問題領域に何が存在し、何を意味し、どのような構造と規則を持つかを中心に扱います。状態や振る舞いも、それが対象そのものの意味を表す場合はドメイン・モデルに含まれます。次のアプリケーションモデリング (application modeling) は、利用者の目的とユースケースを起点に、UI と動的な協調、相互作用によってその目的をどう実現するかを中心に扱います。

前回の記事: 文芸モデリング

## 2 ドメインモデリングが組織するもの

ドメインモデリングでは、問題領域について少なくとも次の四つを明らかにします。

| 観点 | 問い                       |
|----|--------------------------|
| 意味 | その用語や概念は、この業務では何を意味するのか  |
| 構造 | 概念はどのように分類され、関係し、所有されるのか |

|    |                         |
|----|-------------------------|
| 規則 | 何が許され、何が禁止され、何が常に成立するのか |
| 境界 | どの範囲で、その意味、構造、規則が有効なのか  |

この四つは別々に決めるものではありません。たとえば「顧客」という用語の意味は、販売、契約、請求、サポートで同じとは限りません。どのコンテキストで用語を使うかが決まると、必要な属性、関係、規則も変わります。逆に、同じ用語に異なる規則が結び付いていることから、コンテキストを分ける必要が見える場合もあります。

### 3 ドメイン・モデルは目的別に構成する適用モデル

オブジェクト・モデル、知識モデル、文芸モデルは、同じ対象の異なる側面を表す SimpleModeling モデルの構成要素です。ドメイン・モデルは、それらと同列の構成要素ではなく、問題領域を理解し開発するという目的に必要なモデル要素を三つの構成要素から選び、対応付けたモデルです。

この区別により、ドメイン・モデルをクラス図だけに限定せず、同時に何でも含む巨大な文書にすることも避けられます。形式構造はオブジェクト・モデル、意味関係と根拠は知識モデル、文脈と説明は文芸モデルに保持したまま、共通の用語識別子とモデル要素の対応によって一つのドメイン・モデルとして扱います。

### 4 用語から概念を共有する

ドメインモデリングの入口は用語です。用語集 (glossary) は、名称を並べる一覧ではなく、概念を識別し、その定義、適用範囲、同義語を共有するための基点です。一般の日本語と重なる「関係」「関連」「所有」「合成」のような語も、用語集の識別子を通してモデル上の概念として区別します。

用語を定義するときは、単独の一文だけで固定するのではなく、次の情報を対応付けます。

- どのコンテキストで使う用語か
- 何を含み、何を含まないか
- 近い用語や対立する用語とどう違うか
- どの規則、文書、モデル要素、事例が根拠になるか

同じ識別子をオブジェクト・モデル、知識モデル、文芸モデルから参照することで、人間と生成 AI が同じ概念をたどれます。

### 5 コンテキストと境界

コンテキストは、用語と規則を解釈するための背景です。組織、業務目的、責任、利用者、時間、法制度などによって、同じ言葉の意味が変わります。コンテキストを明示しないまま一つの定義へ統合すると、異なる業務上の関心が衝突し、モデルが不自然に複雑になります。

境界づけられたコンテキストは、モデルの用語、構造、規則が一貫した意味を持つ範囲です。境界の目的は、システムを機械的に細分化することではありません。内部では意味を一貫させ、外部とは意味の違いを明示して接続できるようにすることです。

一つのドメイン・モデルは、必要に応じて複数の境界づけられたコンテキストを含みます。それぞ

れの内部で意味を一貫させ、コンテキスト間では同じ用語や情報がどのように対応するかを明示します。

境界は組織図や既存システムの分割と常に一致するわけではありません。業務上の責任、変更理由、規則のまとまり、用語の意味を手掛かりに候補を作り、実際の事例と照らして調整します。

## 6 ユビキタス言語

ユビキタス言語は、境界づけられたコンテキストの中で、ドメインエキスパート、開発者、生成 AI、モデル、プログラムが共有する言語です。用語集に登録するだけでは成立しません。会話、要求、モデル、CML、レビュー、テスト、ドキュメントで同じ意味を使い、意味のずれが見つければモデルと用語を一緒に更新します。

ユビキタス言語は、業務で使われている言葉が無条件に採用することでも、実装上の名前を業務へ押し付けることでもありません。業務上の区別を保ち、モデルとして扱える精度を持つ言語を協業によって育てます。

## 7 三つの構成要素でドメイン・モデルを表す

一つのドメイン・モデルに対して、三つの構成要素は次の役割を持ちます。

| 構成要素       | ドメイン・モデルにおける役割                     |
|------------|------------------------------------|
| オブジェクト・モデル | 概念の型、分類、関係、責務、制約、状態、振る舞いを形式的に表す    |
| 知識モデル      | 用語、概念、分類、意味関係、規則、根拠、出典を探索可能に結ぶ     |
| 文芸モデル      | 業務の文脈、意図、判断理由、事例、既存仕様、説明を自然言語で保持する |

たとえば業務規則は、文芸モデルの説明だけに置くのではなく、知識モデルで根拠と適用範囲をたどれ、必要な部分はオブジェクト・モデルの制約や振る舞いとして実行可能に表します。三つへ同じ文章を複製するのではなく、それぞれに適した表現を対応付けます。

## 8 ドメイン・モデルを反復的に作る

ドメインモデリングは、一度の分析で完成させる直線的な工程ではありません。次の活動を行き来しながら、モデルの妥当性を高めます。

1. 既存文書、会話、事例、規則、用語を文芸モデルとして収集する。
2. 用語と概念を整理し、コンテキストごとの意味を区別する。
3. 境界づけられたコンテキストの候補と相互関係を作る。
4. 概念、分類、関係、規則をオブジェクト・モデルとして形式化する。
5. 意味関係、根拠、出典を知識モデルとして接続する。
6. 生成 AI がオブジェクト・モデルを生成、更新する。
7. 開発者が textus-cbd-support でモデル、差分、根拠、影響範囲、検証結果をレビューする。
8. レビュー結果をフィードバックとして生成 AI へ返し、必要に応じてドメインエキスパートと業務上の妥当性を確認する。

順序は固定ではありません。形式化したことで用語の曖昧さが見つかり、事例を確認したことで境界が変わり、根拠をたどったことで規則の例外が見つかることがあります。更新の理由と影響を追跡できることが重要です。

## 9 生成 AI と開発者のモデル開発ループ

生成 AI は、大量の文書と用語を読み、概念の候補、重複、矛盾、境界案、形式構造を提示できます。SimpleModeling では、textus-bok が文芸モデルを知識モデル化して生成 AI へ接続し、生成 AI がオブジェクト・モデルを主として生成、更新します。オブジェクト・モデルを人間が最初から手作業で記述することよりも、生成 AI による生成と更新を開発の通常ループに組み込むことを重視します。

そのために textus-cbd-support が、生成されたモデルの可視化、前回との差分、根拠、影響範囲、検証結果を開発者へ提示します。開発者はそれらを確認し、モデル構造、設計上の意図、実現可能性、変更影響を判断して、修正方針をフィードバックとして生成 AI へ返します。業務上の意味や例外について判断が必要な場合は、ドメインエキスパートの確認もこのループへ接続します。

この開発ループでは、生成 AI の回答をそのまま正本にしません。生成 AI を主要なモデル作成主体として使いながら、開発者がレビューとフィードバックを継続できることが重要です。用語、根拠、モデル要素、変更理由、レビュー結果の対応を残すことで、AI による更新を追跡可能な開発作業として扱います。

## 10 「注文を確定する」の例

「注文を確定する」を例に考えます。販売コンテキストでは、注文確定は、利用者が注文内容を確認し、以後の変更に制約が加わる業務上の出来事かもしれません。一方、決済コンテキストでは支払いの承認、在庫コンテキストでは在庫の引当が重要です。三つを一つの「確定」という操作へ押し込むと、用語と責任が曖昧になります。

- 文芸モデルは、利用者の目的、注文確定のシナリオ、例外、業務上の判断理由を記述します。
- 知識モデルは、「注文確定」の意味、適用するコンテキスト、関連する規則、根拠、決済や在庫との意味関係を結びます。
- オブジェクト・モデルは、注文、注文明細、状態、制約、出来事など、販売コンテキストで必要な形式構造を表します。

この整理により、販売、決済、在庫がそれぞれのモデルを保ちながら、境界を越える情報と責任を明示できます。次回は、このドメイン・モデルを使い、利用者の目的をユースケースとして捉え、UI、サービス、操作、イベントなどがどのように協調して目的を実現するかをアプリケーション・モデルとして組織します。

## 11 まとめ

- ドメインモデリングは、問題領域の意味、構造、規則、境界をドメイン・モデルとして組織します。
- ドメイン・モデルは第四の構成要素ではなく、オブジェクト・モデル、知識モデル、文芸モデルから目的に必要な要素を組織した適用モデルです。
- 用語集、コンテキスト、境界づけられたコンテキスト、ユビキタス言語によって、意味と有効範囲を共有します。
- 三つの構成要素を共通の用語識別子とモデル要素の対応で接続します。
- 生成 AI がオブジェクト・モデルを主として生成、更新し、開発者が textus-cbd-support で差分、根拠、影響、検証結果をレビューしてフィードバックを返します。
- ドメイン・モデルは、事例、根拠、形式化、AI 生成、開発者レビューを往復する開発ループで反復的に改善します。

## 12 次回

第 9 回「アプリケーションモデリング」では、利用者の目的とユースケースを起点に、UI と動的な協調、相互作用によって目的を実現するアプリケーション・モデルを構成します。ドメイン・モデルの状態や振る舞いを土台として利用しながら、UI、サービス、操作、イベントなどの協調を扱います。