

## かみ合わない議論を解消するために一ユビキタス言語

土屋 正人

Masato Tsuchiya

議論していて、話がかみ合わないことがあります。様々な状況があるでしょうが、大別すると、

- 抽象度(レイヤ)が異なる場合
- 視点(ドメイン)が異なる場合
- 両者が混在している場合

に分けられると思います。同じテーマについて議論しても、

- ビジョン
- 目的/理由
- 目標
- 実現手段

など、さまざまな抽象度が入り乱れることがあります。抽象度が高すぎると、「曖昧で何を言っているのかわからない」状態に、抽象度が低すぎると「すなわち具体的すぎると、「木を見て森を見ず」状態に、それぞれなりかねません。

### ◆レイヤを合わせる

抽象度をレイヤと見なすことで、同じレイヤで議論する必要性が見えてきます。声を高くしても異なる階にいる人には届きません。お互い同じ階で話す必要があります。システム開発でも同様でしょう。

- 要望のレイヤ
- 要求のレイヤ
- 仕様のレイヤ
- 設計のレイヤ
- 実装のレイヤ

今、どのレイヤの話をしているのかを参加者が把握していなければ議論はかみ合いません。意識していても次第に混在していくことがあるので、適切なファシリテーションが必要です。ファシリテーションングラフィクスツールを使って可視化することが有効です。ファシリテーションングラフィクスとしては、先に挙げたビジョンや目的、目標、手段などの議論の際には Vol.83 で紹介したリーニキャンパスが有効です。

要件定義においては、要求と仕様のレイヤを明確にする表記法である USDM (Universal Specification Describing Manner) があります。要求とその理由を明らかにして、階層構造で仕様を定義して行きます。

要求仕様	
要求	UC001 商品を受注する
理由	顧客がオンライン注文登録したい
説明	電話、FAX、インターネットで受け付け、顧客からの注文登録する
要求	UC001-01
理由	注文内容登録する
説明	仕様の詳細を記述して、参照できるようにしたい
□	UC001-01-1 受注日/受注日付(年月日)
□	UC001-01-2 受注日/受注時刻(秒)の日付とする
□	UC001-01-3 商品コードを受注日付する
□	UC001-01-4 商品名/商品名(最大4桁)
□	UC001-01-5 顧客コードを受注日付する
□	UC001-01-6 受注日、商品コード、商品名/数量、顧客コードを受注日付する
要求	UC001-02
理由	受注日の入力/参照したい
説明	
□	UC001-02-1 受注日/受注時刻(秒)の日付/年表示、日付/時刻とする
□	UC001-02-2 月/日/年の別欄(秒)は、入力されていない日付とする
□	UC001-02-3 日付が入力されていない場合は、現在の日付とする
要求	UC001-03
理由	商品コードを受注日付したい
説明	
□	UC001-03-1 登録されている商品名を受注日付する
□	UC001-03-2 一覧表示する商品名(最大4桁)に、検索キーワードで検索できる
□	UC001-03-3 商品名/検索キーワードが登録される商品名を受注日付する
□	UC001-03-4 商品名/検索キーワードが登録される商品名を受注日付する
要求	UC001-04
理由	顧客コードを受注日付したい
説明	
□	UC001-04-1 登録されている顧客名を受注日付する
□	UC001-04-2 一覧表示する顧客名(最大4桁)に、検索キーワードで検索できる
□	UC001-04-3 顧客名/検索キーワードが登録される顧客名を受注日付する
□	UC001-04-4 顧客名/検索キーワードが登録される顧客名を受注日付する
要求	UC001-05
理由	営業在庫の不足を確認したい
説明	在庫確認、数量の取りたいこと/確認したい
□	UC001-05-1 営業在庫(在庫)に、商品名/数量の不足を確認する
要求	UC001-06
理由	不足数量の補充を希望したい
説明	
□	UC001-06-1 営業在庫(在庫)に、商品名/数量の不足を確認する
□	UC001-06-2 営業在庫(在庫)に、商品名/数量の不足を確認する

図 1 USDM サンプル

### ◆視点を合わせる

異なる視点が混在することもあります。

- 顧客の視点
- ユーザの視点
- 開発者の視点

などが入り乱れると、やはり議論はかみ合わなくなります。ここでは言葉が問題になります。業務領域(問題領域)とシステム領域(ソリューション領域)の言葉が混在すると理解できずに何度も確認が必要になります。確認する

場合はまだ良いほうで、誤解したまま進行する危険があります。用語の認識の不一致はプロジェクトを破綻させます。視点を統一すれば良いのですが、統一することで漏れてしまう要素があつては困ります。

エリック・エヴァンスは『ドメイン駆動設計』(略称DDD)の中で、「ユビキタス言語」を提唱しています。ユビキタス言語は、顧客、ユーザ、業務の専門家、開発者の間で使うための共通言語です。ドメインモデルをユビキタス言語の骨格として使い、チームはコミュニケーションとコードでユビキタス言語を厳格に用いることを約束します。モデリングにおいては、声に出してモデルを読んでもることが推奨されています。モデルの要素を結びつけながら、シナリオを声に出して描写します。その際、例えば業務のエキスパートがモデルを理解できないとしたら、モデルに何らかの問題があることになります。

## ◆ドメインモデル

ドメインモデルはUMLのクラス図で描かれることが多いと思いますが、UMLの記法にこだわりすぎないほうが良いでしょう。UMLの仕様は2.Xになって肥大化しているので、それにこだわると、モデルのためのモデルになりかねません。手段が目的になってしまうわけで、これでは本末転倒です。

ドメインモデルで定義する概念を識別するための方法として、名詞抽出法や動詞に着目する手法がありますが、まずは業務(ドメイン)のエキスパートが使う言葉に耳を傾ける必要があるでしょう。

エリック・エヴァンスは前掲書の中で、モデルにとって有益になりうる概念の手がかりを挙げています。

- 何か複雑なものを簡潔に述べている用語がないだろうか？
- ドメインエキスパートに言葉の選び方を正されていないか？

- あなたが特定のフレーズを使ったときに、ドメインエキスパートたちの困惑した表情が消えることはないか？

概念は、ドメインエキスパートの関心事と関連知識を提供するもので、表面的な側面は捨て去る必要があります。

例えば運送業務であれば、トラックや船舶、飛行機が主要な関心事ではありません。何のために必要かを考えると、把握する対象である「輸送機器の運行(運航)」という概念が浮かび上がります。保温ポット制御であれば、ポットの構成要素が主要な関心事ではありません。制御対象は「水(お湯)」です。音楽管理であれば、物理的なCDやCDタイトル、配信方法が主要な関心事ではありません。管理対象は「演奏」です。

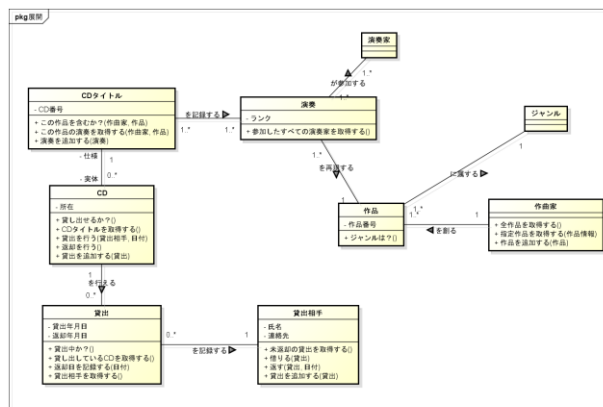


図2 ドメインモデルサンプル

このような形のないものは、モデルを洗練して行くときに見つかることが多いと思います。これらを見出す思考は、**オントロジ**に近いかもしれません。

## ◆名前を創る

概念の名づけに迷う場合は、**意味がない名前でもいい**ので「創ってみる」というのもいいと思います。チームで共有することができれば、ユビキタス言語として成立すると思います。

夢を。

GSLetterNeo Vol. 87  
 2015年10月20日発行  
 発行者●株式会社SRA 先端技術研究所  
 編集者●土屋正人

バックナンバーを公開しています●<http://www.sra.co.jp/gslletter>  
 ご感想・お問い合わせはこちらへお願いします●[gsneo@sra.co.jp](mailto:gsneo@sra.co.jp)



株式会社SRA

〒171-8513 東京都豊島区南池袋2-32-8

夢を。Yawaraka Innovation  
 やわらかいのバージョン