

UML によるビジネスモデリングの試み

児玉公信†
Kiminobu KODAMA

1. はじめに

1.1 ビジネスモデル

企業が競争社会で生き抜くためには、ビジネス環境の変化に受動的に対応するだけでは十分でなく、ビジネスの変化を能動的に生み出す戦略も必要である。妥当な戦略を導く前提として、ビジネスの仕組みや構造を明示的に表現したい。こうしたビジネスの仕組みや構造を、一定の表記ルールで記述したものをビジネスモデルと呼ぶ。

1.2 ビジネスモデリングの目的

ビジネスモデルを書く目的は、第 1 に、ビジネスそのものを理解することである。一旦、モデルとして記述されれば、その仕組みの改善点が見えてきたり、プロセスを変えたらどうなるかといったシミュレーションが行える。

第 2 に、情報システムを開発する根拠とすることである。残念なことに、開発者は本質的なビジネスニーズを把握しないまま情報システムを開発することがある。何のために開発するのが意識されないと、的はずれな機能を作ってしまうたり、過剰品質で開発費ばかり増えたりすることになる。

1.3 ビジネスモデリングのレベル

モデルのレベルは、問題ドメイン、アプリケーションドメイン、ソリューションドメインの 3 層とらえる(図 1)。問題ドメインは観察者が実世界だと解釈^[1]している領域、アプリケーションドメインは問題ドメインのうち解決しようとする対象領域、ソリューションドメインは対象に機能を引き当てた領域である。ビジネスモデルは、問題ドメインのレベルで記述する。

ビジネスモデルの具体的な記述方法は、以下に述べるように模索中である。

1.4 UML によるビジネスモデリング

UML (Unified Modeling Language)^[2]はソフトウェアの仕組みや情報の構造を図式的に記述するためのルールである。

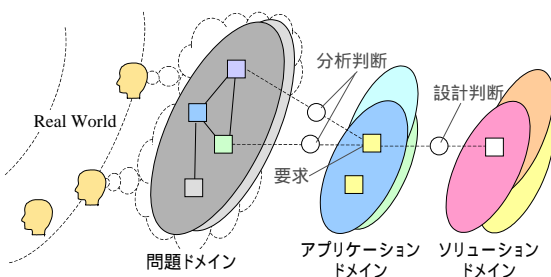


図 1 モデリングのレベル

オブジェクト指向技術に関する標準化団体である OMG がモデル記法の標準として採用し、ISO でも国際標準化の作業が進められている (ISO/IEC 19501-1 FDIS)。

ビジネスモデルを UML で記述することの意義は、厳格に記述できること、統一された記法を用いることによって、多くの関係者が理解できること、情報システムのモデルとの対応づけが容易なことなどである。一方で、UML がビジネスモデルを記述するために制定されたものではないことから、そのための何らかの工夫を必要とする。

本稿では、そうしたモデリングの試みの例として、Marshall および Eriksson & Penker を取り上げて概説する。

2. Marshall の方法

Marshall^[3]の方法は、ビジネス要素を抽象的または総称的なビジネスオブジェクトとしてコンポーネント化しておき、個別のビジネスに合わせてこれらを継承して拡張することによって、情報システムを構築しようとした。その手法の有用性はともかく、それ以前に、ビジネスをどのようにとらえるかについての議論、特にその中で例示されるビジネスプラクティスの分析が示唆に富む。

2.1 ビジネスモデルの要素

ビジネスモデルを構成する要素として、エンティティ、プロセス、目標値、組織 (パーティ) という 4 つを挙げ、それらを UML の図を使って表す。ここでいうエンティティとは、顧客や従業員や、装置のようにアクタになりうるもの、資産、製品などのビジネスの対象を指す。プロセスとは取引などのビジネス活動のこと、目標値とはエンティティの状態であり、組織とは、そのビジネスを行う主体のことである。

4 つのモデル要素のうち、エンティティ、プロセス、目標値についてももう少し詳しく見ていく。

2.2 エンティティ

エンティティはそれを扱うプロセスごとに、値とプロパティを持つと考える (図 2)。プロパティとはロールまたは

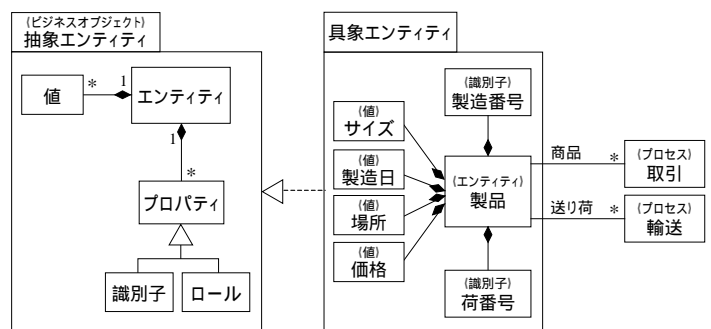


図 2 抽象エンティティ

† (株) エクサ



図3 プロセスの分類

識別子である。たとえば、「生産物」は取引プロセスにおいて「商品」というロール、「製造番号」という識別子、「価格」という値を持つ。また、輸送プロセスにおいて「送り荷」というロール、「荷番号」という識別子、荷物の「サイズ」や「重量」という値を持つ。

このことは、エンティティはそれ単独では認識されることはなく、プロセスを通して認識されるということの意味する。道ばたに落ちているゴミはゴミにすぎず、エンティティとして取り上げる意味はないが、それをリサイクルの取引に使えば商品というエンティティのロールを持ち、そのときはじめて「ゴミ」というエンティティが正統に認識されることになる。

2.3 プロセス

プロセスは1つ以上のプロセスステップからなる階層(コンポジット^[4])構造をしている。各プロセスステップはエンティティに働きかけて、値の変化を促す。たとえば、製造プロセスは複数の組立や加工のプロセスステップからなる。組立では部品のエンティティが有形の製品のエンティティに変換され、その値である「サイズ」や「数量」が決定する。

プロセスは図3のように分類できる。この分類は、エンティティの変化の意味を整理するのに役に立つ。

2.4 目標値

エンティティの値をどのように決定したり変化させたりするかが、プロセスの目標値である。目標値と実績値を比較し、その差異を評価することで、ビジネス環境の変化を早期に察知できる。

目標値は、管理レベルに合わせて階層構造をしている。製造業の例で言えば、需要予測に基づいて全社で何をいくつか作るかという目標を立てる。各工場では、その目標を達成するために設備をどのくらい増強したり、原料をどのくらい手配したらよいかを計画したりする。

意識されることは少ないが、目標値の階層構造は、ビジネスモデルの至るところに存在する。たとえば、プロジェクトのWBS(Work Breakdown Structure)の階層構造は、典型的なコンポジット構造をしているし、勘定科目体系も目標値の階層構造の1つである。

3. Eriksson & Penker の方法

Eriksson & Penker^[5]は、Marshallのビジネスモデルよりマクロな見方を提供している。彼らは、ビジネスモデルに書くべきビジネス要素として、エンティティ(オブジェクト)、ゴール(目的)、プロセス、ルールの4つを挙げている。エンティティはMarshallと同様の意味だが、ゴールをマクロ

な目標値、プロセスもマクロな活動としてとらえる。ルールは概念レベルの制約記述で、OCL(オブジェクト制約言語)を用いる。その上で、エンティティやプロセスをソフトウェアパターン^[6]として記述しようとしている点が特徴的である。

3.1 ゴールどうしの依存関係

Marshallの目標値はプロセスの成果(パフォーマンス)を数値化したものだから、大きな目標値を階層構造に沿ってブレイクダウンしたり、実績値をロールアップしたりすることには意味がある。ただし、目標値を設定できるのは、かなり具体的な戦略が与えられた場合である。

これよりも大きい粒度であるゴールを扱う場合は、階層的にサブゴールに分割することが合理的とは限らない。サブゴールが互いに矛盾することもあるからである。たとえば、収益を向上するというゴールに対するサブゴールとしてレイオフを実施したとしても、人員の削減が顧客サービスの低下につながって、かえって長期的には収益を低下させるかもしれない。このように、ゴールやサブゴールは、階層構造というよりも、依存関係にあり、場合によっては、ループになっていることもあると考える。

3.2 プロセス図

Eriksson & Penkerは、マクロなビジネスプロセスを記述するために、UMLの活動図を拡張したプロセス図とアセンブリライン図を提案した。

プロセス図は、プロセス内部の詳細を隠蔽して、処理内容を連想させる名前を付けたものである。プロセスには、論理的なオブジェクト(情報)や物理的なオブジェクト(エンティティ)が投入され、何らかの変換や交換の結果として情報やエンティティが生成される。プロセスは、プロセスのゴール(たとえば、3日以内に完了するとか、生産歩留まりを98%以上にするなど)と出力オブジェクトのゴール(たとえば、製品品質の分散がいくつか以下であるなど)を持つ。

プロセスシンボルはUMLの活動図を拡張した図法である。プロセスに投入されるオブジェクトは破線の矢印で示され、活動図のオブジェクトフローに対応する。

プロセスをある大ききで切り出して順序づけて記述することで、ビジネスプロセスの特長を浮かび上がらせることができる。図4は、筆者が試みに、プロセス図を使って2

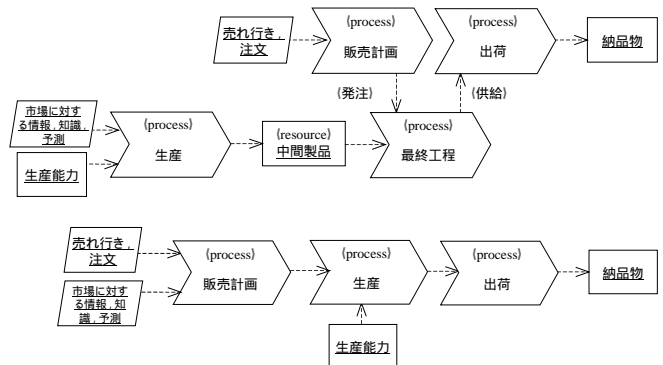


図4 プロセス図の例

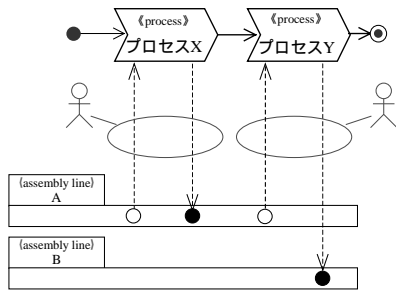


図5 アセンブリライン図

つのビジネスプロセスを記述したものである。

図4上は Assemble-to-order(受注組立)型の生産ビジネスプロセスの記述例である。まず市場動向に合わせて中間製品を大量生産して在庫しておく。これと並行して、実際の店頭での売れ行きなどを見ながら、中間製品を売れ行きの良い製品に組み立てて出荷するというプロセスである。こうすることで、製品在庫を作りすぎたり、売れない製品を作らないようにする。

図4下は Make-to-stock(見込み生産)型ビジネスプロセスの記述例で、市場動向を見て最終製品まで作って在庫して、在庫から販売する。コンスタントに売れる製品ならこのモデルでよいが、付加価値で売ろうとする製品では、このプロセスはふさわしくない。

ここでどちらのモデルがよいかを議論しているのではない。2つのビジネスプロセスの違いが、プロセス図を使うことで目に見える形で明確に表現できる点が重要である。定量的な優位性までは表記できないが、プロセス改善を検討するにはよい表現である。

3.3 アセンブリライン図

アセンブリライン図(図5)は、プロセス図に主要なオブジェクト(エンティティ)との関わりを加えたものである。アセンブリラインとは、元々ベルトコンベアー式に流れる製造組立ラインのことだが、製造プロセスそのものを表わしているわけではない。プロセスの流れに合わせてオブジェクトをラインから取り出したり、ライン上に戻したりする操作になぞらえたものである。この図によって、プロセスとオブジェクトの状態遷移が直感的に表現できている。

構文的には、UMLのパッケージを横長にして、その中にオブジェクトのシンボル(と)を書き、それをプロセスに渡すオブジェクトフローの記法を組み合わせている。オブジェクトの状態遷移に着目することで、プロセスの粒度をある程度統制できる。

プロセスがオブジェクトとやりとりする一群の機能にアクタを割り付けることで、ユースケースの素を取り出すことができる。

4. 検討

4.1 ビジネス要素

両方のビジネスモデルで共通しているのが、ビジネス要素としてエンティティ、プロセス、目的を挙げている点である。一方、Marshall はそれらに加えて組織を取り上げ、

Eriksson & Penkerはルールを取り上げる点が異なる。前者は組織間の相互作用を重視し、後者はビジネスパターンとして記述することを重視している。

4.2 UMLに不足する点

Eriksson & Penker がビジネスモデルを記述するためにUMLを拡張したのは、UMLがオリジナルのままでは、ビジネスモデルのような抽象的なモデルを書きにくいからである。典型的なのが、UMLクラス図における汎化の記号である。プログラムレベルでは、スーパークラスを継承してサブクラスを作るという操作イメージを、白三角矢印でスーパークラスとサブクラスをつなぐことで直感的に表現している。しかし、概念レベルでは、部分集合の関係を別々の矩形で表現するために、1つのオブジェクトが2つの集合に属しているように見えたりして、直感に合わない。

また、ゴールどうしのダイナミックな関連を表すUMLの拡張はない。たとえば、漁業で魚を捕りすぎると資源が枯渇するから、漁師の漁獲高を制限する。しかし、その効果が表れるのは数年後であり、さらにその効果が保証されない場合に、漁師の行動を制御できるのか。抜け駆けしてしまう恐れもある。こうした状況をどう記述すべきだろうか^[7]。これは、システムダイナミクス^{[8][9]}として扱われるテーマであるが、ビジネスモデリングでも重要なテーマである。

さらに、プロセス自身を変えるとといった自己書き換えのプロセスモデルも表現できない(これはUMLに限ったことではないが)。このように、UMLをどう使うか以前に、ビジネスモデルをどうとらえるかについて、十分な議論がされていない状況である。

参考文献

- [1] Wilson, B.: 根来監訳「システム仕様の分析学」, 共立出版, 1996
- [2] OMG:OMG Japan SIG 翻訳委員会 UML 作業部会訳「UML仕様書」, アスキー, 2001
- [3] Marshall, C.: 児玉監訳「企業情報システムの一般モデル」, ピアソンエデュケーション, 2001
- [4] Gamma, E., et al, *Design Patterns*. Reading, MA, Addison-Wesley, 1995. 本位田真一ら(監訳), 「デザインパターン(改訂版)」, ソフトバンクパブリッシング, 1999
- [5] Eriksson, H. and Penker, M.: 本位田ほか監訳「UMLによるビジネスモデリング」, ソフトバンク, 2002
- [6] Hillside Group, <http://www.hillside.net/patterns>
- [7] Kim, D. and Anderson, V.: ニューチャーネットワークス監訳: 「システム・シンキングトレーニングブック」, JMAM, 2002
- [8] Senge, P.: 守部信之訳「最強組織の法則」, 徳間書店, 1995
- [9] 森田道也編著: 「経営システムのモデリング学習」, 牧野書店, 1997