

セールスドメインにおけるドメインモデルの獲得とその再利用

平井俊光, 田村恭久, 伊藤潔
上智大学理工学部

本稿では類似したシステム群をより少ない工数で開発するためのドメインモデルの構築法とその利用法について述べる。対象としてセールス業務ドメインを取り上げる。当該ドメインでは、商品、顧客、問屋といったエンティティと、仕入れ、販売のオペレーションが存在する。これらに対応するテーブル仕様と問い合わせ仕様から成るドメインプロダクトモデルを導出するための、語彙の列挙、グループ化、関係付けを行なうドメインプロセスモデルを構成した。これらのセールス業務のドメインモデルは、獲得したプロセスモデルに沿ってテーブル仕様や問い合わせ仕様のモデルに個々のシステム依存の要素を追加・改変することにより、ドメインに含まれるシステムの開発に再利用可能である。

Elicitation and Usage of Domain Model for Sales Business Domain

Toshimitsu Hirai, Yasuhisa Tamura, Kiyoshi Itoh
Faculty of Science and Technology,
Sophia University
Kioi-cho 7-1 Chiyoda-ku Tokyo 102 JAPAN
{to-hirai, ytamura, itohkiyo}@sophia.ac.jp

This paper shows the domain model and its usage of sales business domain for decreasing time and workload to develop systems in the domain. There are common features of the target systems: three major entities of goods, customers and wholesale stores, and two major operations of stocking and selling. The authors developed a domain process model that enables to build a domain product model with specifications of element tables and associated query, and provides enumeration, classification and making relationships of entities and operations. By appending or altering characteristic entities or operations in accordance with the process model, the domain model enable to develop systems in the sales business domain efficiently.

1 はじめに

ソフトウェアやシステムの開発の際の再利用を促進する手段としてドメイン分析・モデリング技術 (Domain Analysis and Modeling :以下DAMと略す) [Prieto-Diaz87] [Arango91] [Tracz92] [田村94] [伊藤94,95a]での重要性が認識されている。ここでは、既存のソフトウェア群をモデル化し(ドメインモデルの獲得)、それを新システムの開発時に再利用しようとする。

筆者らは、DAMを「対象システム自身が本来もつ各種の性質や開発上の多様な知識を十分に分析し認識して組織化し、システムの開発に有効な、共通の対象領域(ドメイン)に属する、用語、問題の捉え方、システムの構造、システムの作り方等の、固有な概念構造を得るプロセスである。この概念構造をドメインモデルと呼ぶ。このドメインモデルを用いて、複雑で大規模な実際のシステム開発での生産性の向上と再利用の促進を図ろうとしている」と定義した[田村94][伊藤95a]。得られたドメインモデルは、そのドメインに属するシステムを繰り返して開発する際のカテゴリモデル、ひな型モデル、標準モデル、ハンドブック等として使われ、ソフトウェア開発の生産性と再利用性を向上させる。システムの開発フェーズに即して述べると、DAM技術とは、複数の類似した業務やソフトウェアが属するドメインを明確にし、当該ドメインが持つ要求仕様や設計仕様のモデルを構築し、あるいはドメインが持つ仕様を導出する共通のプロセスを明確化し、この共通のモデルやプロセスをもとにドメインに属する実際のシステム開発を行なう技術である。個々の業務やソフトウェアは、ドメインという抽象的な対象の実例とみなすことができる。

DAMの目的は、本来(a)ドメインに属する過去の生産物の再利用、にあったが、(b)ドメイン特有/固有の開発プロセス・方法・ツールの体系化/明確化/改良/創生、(c)組織内での開発事例/ノウハウ/経験/方法の収集/体系化/教育/継承、なども重要と認識される[伊藤95a]。

本稿ではセールス業務を例とし、セールスドメインのドメインモデルを導出し、得られたドメインモデルを個々のシステム開発に利用する指針を示す。セールス業務とは、小売店が商品を売買って利潤を得る業務であると捉える。すなわち、商品の流通とそれに伴う金銭の

授受が当該ドメインの本質的な要素であると考えられる。小売店には商品を売買すること以外にも、店員の就労時間の管理や、給料の支払いなどの業務が存在する。しかしこれらはセールスに特有な業務ではなく、雇用に関する業務であるので、セールスドメインモデルを構築するときの考察の対象としない。

商品と金銭の流通に注目して当該ドメインをみると、(1) 取り扱う商品の情報の管理と、その取引である(2) 仕入れと(3) 販売の伝票の管理が必要である。また取引先である(4) 問屋と(5) 顧客の情報の管理が必要である。そこで本稿では、以下のチェックポイントを設けて、これらの情報を管理するテーブル仕様と、テーブルの各々のデータ項目への問い合わせ仕様を導出する、(a) 語彙の列挙、(b) 語彙の分類、(c) アイテムのグループ化、(d) テーブル仕様の決定、(e) アイテムとオペレーションの関係付け、(f) 問合せ仕様の決定、からなるプロセスを提案する。この導出プロセスで、テーブル仕様と問合せ仕様を導出できる。

著者らは、ドメインモデルを対象問題、導出プロセス、成果物の3つの側面からとらえるTriadic Domain Modelが有効であると考え[伊藤95b]。セールスドメインでは、セールス業務の定義と語彙を併せた対象問題があり、これをドメインプロブレムモデルとよぶ。また、得られたテーブル仕様および問合せ仕様などの成果物をドメインプロダクトモデルとよぶ。さらに示した導出プロセスをドメインプロセスモデルとよぶ。

個々のシステム開発にこのプロダクトモデルを利用するためには、まず個々のシステムで用いられる語彙を列挙し、ドメインプロダクトモデルの持つ語彙と対応付ける。すべての語彙が対応付けられるなら、ドメインプロダクトモデルのテーブル仕様と、問合せ仕様の語彙をシステム固有の語彙で置き換えることによって、所望のテーブル仕様と問合せ仕様を得られる。対応付けられなかった語彙は上述のドメインプロセスモデルに沿ってテーブル仕様と問合せ仕様へ導く。

本稿の成果の第一はセールス業務ドメインのドメインモデルを構築し、それを実際のシステム開発に利用するための指針を示したことである。このセールスドメインモデルには、上記のテーブル仕様や問い合わせ仕様の他に、セールス業務ドメインを分析する手順やプロセス

も含まれており、これらはセールスドメインに含まれる個々のシステムを開発する際に再利用できる。第二の成果は個々のシステムのテーブル仕様や問い合わせ仕様を導出するために、このドメインモデルに含まれる分析の手順やプロセスを組み込んだ計算機による分析支援環境の実現可能性を示したことである。

以下、2節ではセールス業務におけるテーブル仕様と問い合わせ仕様からなるプロダクトモデルと、その導出のためのドメインプロセスモデルを示す。3節では、個々のシステム開発でのドメインモデルの利用と、そのための計算機による分析支援環境について考察する。4節にまとめを示す。

2 セールスドメインモデルと導出プロセス

本節では、セールス業務におけるテーブル仕様と問い合わせ仕様からなるプロダクトモデルと、その導出のためのドメインプロセスモデルを示す。これらは、筆者らが複数のセールス業務システムを比較分析し、それらの共通要素を抽出した結果得られたものである。

本節で述べるプロダクトモデルは、セールス業務ドメ

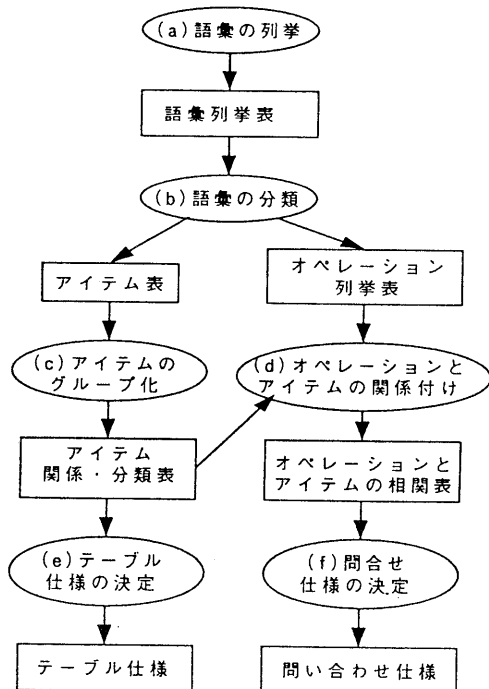


図1 セールスドメインのプロセスモデル

インに含まれるシステムの開発において共通に用いることができる。このため、説明で用いる語彙などは、個々のシステムに依存しない用語を用いている。ここで述べるプロダクトモデルを実際のシステム開発に適用する場合には、ここで用いている用語を個々のシステム固有のものに置き換えるなどの作業が必要になる。具体的な作業は3節で述べる。

図1にプロダクトモデルと導出プロセスの全体図を示す。以下、図1に従って各々の手順と得られるプロダクトモデルについて説明する。

2.1 (a) 語彙の列挙

はじめに、セールス業務の専門家や類似システムの開発経験者へのインタビュー、あるいは実際に使われている台帳や伝票などをもとに、個々のシステムで用いられる語彙を列挙する。この時点では語彙を階層化、分類しない。

2.2 (b) 語彙の分類

列挙した語彙を、データ項目となるアイテムと、データ操作となるオペレーターに分類する。同時にアイテムの属性(データ型、データサイズ、値の範囲・制約など)を決定する。

(a) で列挙した語彙は、分類・階層化されていない、未整理の状態のものである。これらの語彙を分類するため、セールス業務ドメインに固有の経験的な知識、すなわち、ドメインを構成する語彙は、商品、顧客、問屋といった主要なエンティティか、あるいは仕入れ、販売といったオペレーションのいずれかに属する、という知識を用いる。これにより、語彙を2つのグループ、すなわち主要なエンティティに属するもの(表1)と、オペレーションに属するもの(表2)に分類する。さらに、アイテムの属性を決定する。属性とはシステム開発の際に即値を与えられた場合のデータ型やサイズ、代入可能な値の範囲、値の候補、その他の不定型の情報を指す。この属性はテーブル仕様を決定するときに必要なだけでなく、オペレーションの実行条件や代入に関する制約を考える際にも必要である。

ドメインの範疇では、この2種類の分類で充分であるが、システム開発の際にはどちらに分類すべきか判断し

かねる語彙がある可能性もある。こういった語彙の存在は、対象業務が本稿で示すプロダクトモデルに対して追

アイテム	データ型	代入値の制約
クレジットカード	integer	
卸し価格	money	> 0
割引率	float	0 <= n <= 1
在庫数量	integer	>= 0
支払方法	enumerate	
取引価格	money	
取引数量	integer	>= 1
取引先ID	integer	重複なし
取引先住所	text	
取引先電話番号	text	
取引先名	text	
商品区分	enumerate	
商品属性	enumerate	
商品番号	integer	重複なし
商品名	text	
顧客名	text	
顧客ID	integer	重複なし
顧客電話番号	text	
顧客住所	text	
製造日	date	
定価	money	> 0
日付	date	
分割回数	integer	
保証期間	date	
利率	float	
受注コード	integer	重複なし
発注コード	integer	重複なし
領収済	boolean	Yes/No
納品済	boolean	Yes/No

加すべき要素を持っていることを示唆している可能性がある。無理に分類せず次のプロセスで新たな分類

表2 オペレーション列挙表

顧客登録
顧客抹消
問屋登録
問屋抹消
商品登録
商品抹消
在庫確認
受注
発注
卸値決定
売値決定
支払い
請求
領収
納品(顧客)
入荷
返品(顧客)
返品(問屋)
返金

を行なう。

2.3 (c) アイテムのグループ化

テーブル仕様を導出するために、(b)で分類したアイテム同士の関係の有無を決定し、それをもとにアイテムをグループ化する。得られた各々のグループがドメインエンティティである。

具体的には、列挙したアイテムのうち、任意の2つのアイテムの関係の有無を明らか

にする。この作業は、表3のようにアイテムを縦横に同じ順番にならべて記入した表を用いて行なうとよい。この表を以後、アイテム関係表と呼ぶ。アイテムの関係表のi行j列の欄には、表のi行目に書かれているアイテムとj列目にかかっているアイテムが、関係あると判断するならば関係あることを示すマークを記入し、関係ないと判断するならば関係ないことを示すマークを記入する。

この表の対角要素は同じアイテム同士の関係を表すので、リーグ戦の対戦表と同様に意味を持たない。またi行j列の成分とj行i列の成分は等しいので、この表の上三角部分もしくは下三角部分の欄が埋まると残りの欄は機械的に埋まる。このアイテム関係表のすべての欄を埋めると、アイテムのすべての組み合わせに関してそれらの関係の有無をもらさず記述したことになる。表3では視覚で区別しやすいように関係あることを示すマークを◆、関係ないことを示すマークを一とした。

次に、それぞれ独立した情報である商品情報、問屋情報、顧客情報のテーブル仕様を明らかにするために、アイテムの関係表の行と列を、関係のあるアイテムができる限る隣接するように並べ換える。

この作業の結果、表3では関係ありを示すマークのみで構成され、かつ、マトリックスの対角成分が対分である正方行列が4つあらわれる(表2網掛部分)。この正方行列を以後、対角正方行列と呼ぶ。対角正方行列を構成する行(もしくは列)に書かれているアイテム群は相互に強い結合関係を持っており、すなわちある種のグループを形成すると考えられる。本稿の例では、商品情報、問屋情報、顧客情報、クレジットカード情報の4つのアイテムグループがあらわれた。このアイテムグループはドメインエンティティであり、そのまま1つのテーブルを構成すると考えられる。このようにアイテム関係表の行列を上述のように並べ換えることによって、商品情報、問屋情報、顧客情報のテーブルを導出することができる。また、クレジットカードのようにセールス業務に必ずしも付随しないエンティティの存在を明らかにする作用もある。

2.4 (d) テーブル仕様の決定

(c)で得られた各々ドメインエンティティの中で、主

表3 アイテム関係表

	割引率	支払方法	取引価格	分割回数	利率	クレジットカード	顧客電話番号	顧客住所	顧客名	顧客ID	問屋電話番号	問屋住所	問屋名	問屋ID	卸し価格	商品番号	商品名	商品属性	商品区分	保証期間	製造日	在庫数量	取引数量	定価	日付
割引率	-	-	◆	-	-	-	-	-	-	-	-	-	-	-	-	◆	-	-	-	-	-	-	-	-	◆
支払方法	-	-	◆	-	-	◆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	◆
取引価格	◆	◆	-	◆	◆	-	-	-	◆	-	-	-	-	-	-	◆	-	-	-	-	-	-	-	◆	-
分割回数	-	-	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	◆	-
利率	-	-	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	◆	-
クレジットカード	-	◆	◆	◆	◆	-	-	◆	◆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	◆
顧客電話番号	-	-	-	-	-	-	◆	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
顧客住所	-	-	-	-	-	-	◆	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
顧客名	-	-	-	-	-	-	◆	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
顧客ID	-	-	◆	-	-	-	◆	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-	-	◆	-	-
問屋電話番号	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-
問屋住所	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-
問屋名	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-
問屋ID	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	-
卸し価格	-	-	-	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	◆	◆	-	-	-	-	◆	-
商品番号	◆	-	◆	-	-	-	-	-	◆	-	-	-	-	-	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	-
商品名	-	-	-	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	-
商品属性	-	-	-	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	-
商品区分	-	-	-	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	-
保証期間	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
製造日	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
在庫数量	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	◆	◆	◆	◆	-	-
取引数量	-	-	◆	-	-	-	-	-	◆	-	-	-	-	◆	-	-	-	-	-	-	-	-	◆	◆	-
定価	◆	-	◆	◆	◆	-	-	-	-	-	-	-	-	-	-	◆	◆	◆	◆	◆	-	-	-	◆	-
日付	-	-	-	-	-	◆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

キーとなるアイテムを選択、もしくは追加してテーブル仕様とする。

まず、商品情報、問屋情報、顧客情報、クレジットカード情報の4つのグループを考える。前述の作業で得られたエンティティに含まれるアイテムのうち、セールスグループを代表し、かつユニークな値を持つアイテムをセ

ールスグループの主キーとすることで、テーブル仕様が決まる。ここで主キーの候補となるアイテムがない場合は、適切なアイテムを追加し主キーとする。

またシステムイメージを明確にするため、アイテムを仕入伝票テーブルと販売伝票テーブルという2種類のテーブルに分類する。更に、仕入伝票テーブルには問屋情

表4 オペレーションアイテム相関表

	販売に関するオペレーション													仕入れに関するオペレーション					
	顧客登録	顧客抹消	問屋登録	問屋抹消	商品登録	商品抹消	在庫確認	受注	返品(顧客)	請求	領収	納品(顧客)	卸値決定	販売価格決定	発注	支払	入荷	返品(問屋)	
問屋住所	-	-	C	D	-	-	-	-	-	-	-	-	-	-	R	R	R	R	-
問屋電話番号	-	-	C	D	-	-	-	-	-	-	-	-	-	-	R	R	R	R	-
問屋名	-	-	C	D	-	-	-	-	-	-	-	-	-	-	R	Q	Q	R	-
問屋ID	-	-	C	D	-	-	-	-	-	-	-	-	-	-	R	Q	Q	Q	-
顧客住所	C	D	-	-	-	-	-	R	R	R	R	R	-	-	-	-	-	-	-
顧客電話番号	C	D	-	-	-	-	-	R	R	R	R	R	-	-	-	-	-	-	-
顧客名	C	D	-	-	-	-	-	Q	Q	Q	Q	Q	-	-	-	-	-	-	-
顧客ID	C	D	-	-	-	-	-	R	Q	Q	Q	Q	-	-	-	-	-	-	-
取引数量	-	-	-	-	-	-	-	C	R	-	-	R	R	R	C	-	-	-	-
定価	-	-	-	-	C	D	-	R	-	-	-	R	R	R	-	-	-	-	-
商品番号	-	-	-	-	C	D	Q	Q	Q	-	-	Q	Q	Q	-	-	Q	Q	-
商品名	-	-	-	-	C	D	-	Q	Q	-	-	R	-	-	R	-	R	Q	-
商品属性	-	-	-	-	C	D	-	Q	Q	-	-	R	-	-	R	-	R	Q	-
商品区分	-	-	-	-	C	D	-	Q	Q	-	-	R	-	-	R	-	R	Q	-
在庫数量	-	-	-	-	-	-	-	R	R	U	-	U	-	-	C	-	U	U	-
卸し価格	-	-	-	-	-	-	-	-	-	-	-	-	R	-	C	-	-	-	-
製造日	-	-	-	-	C	D	-	-	-	-	-	-	-	-	-	-	-	-	-
保証期間	-	-	-	-	C	D	-	-	R	-	-	-	-	-	-	-	-	-	-
割引率	-	-	-	-	-	-	-	C	-	-	-	-	R	-	-	-	-	-	-
取引価格	-	-	-	-	-	-	-	-	R	R	R	-	C	C	-	R	-	-	-
日付	-	-	-	-	-	-	-	C	Q	Q	Q	Q	-	-	C	Q	Q	Q	-
支払方法	-	-	-	-	-	-	-	C	R	R	R	-	-	R	-	-	-	-	-
利率	-	-	-	-	-	-	-	C	R	R	R	-	-	R	-	-	-	-	-
分割回数	-	-	-	-	-	-	-	C	R	R	R	-	-	R	-	-	-	-	-
クレジットカード	C	D	-	-	-	-	-	C	R	R	R	-	-	R	-	-	-	-	-

C:Create, D:Delete, Q:Query-key, R:Refer, U:Update

報テーブルの主キーと商品情報テーブルの主キーを、販売伝票には顧客情報テーブルの主キーと商品情報テーブルの主キーを外部キーとして加える。すべての語彙は(b)の作業でその属性としてデータ型とデータサイズを与えられているので、以上の作業ですべてのテーブル仕様が決まることができる。

2. 5 (e) アイテムとオペレーションの関係付け

(b)で分類されたアイテムとオペレーションを相互に関係付ける。同時にアイテムのデータへのアクセス方法(refer, query, update, create, delete)を決め、アイテムとオペレーションの相関表を作成する(表4)。

関係付けを行う前に、与えられたオペレーション群を仕入と販売に分類しておく、後の作業の効率が良い。これは仕入のオペレーション群は顧客エンティティとは関係付けられることはなく、同様に販売のオペレーション群は問屋エンティティと関係付けられることはないからである。

エンティティのテーブル仕様と、(b)で得られたオペレーションに属する語彙の表をもとに、アイテムとオペレーションの相関表を作成する。この作業は次の(f)で、データベースの操作の処理フローを描くための準備である。まず、各々のオペレーションについてその機能を実行するために使われるアイテムを調べる。次にそのオペレーションがアイテムに対してどのようなアクセスするのか(refer, query, update, create, delete)を調査する。このとき、1つのオペレーションの中で1つのアイテムに対して複数種類のアクセスが行われる、つまり複数回アクセスされることがあることに注意する。ここで、複数個のアイテムに対して複数回のアクセスを行うオペレーションは、より細かい処理単位に分割する。

2. 6 (f) 問合せ仕様の決定

(e)で得られた相関表(表4)をもとに、それぞれのオペレーションについてデータ操作の手順、条件、制約などを処理フローに描き、正当性を検証する。このとき処理フローの各段階でアクセスされるアイテムとそのアクセス方法も併記する。得られた処理フローが問合せ仕様である。

処理フローを図示することによって、データ操作の順序、満たすべき条件、必要なデータなどを視覚で捉え検証することができる。

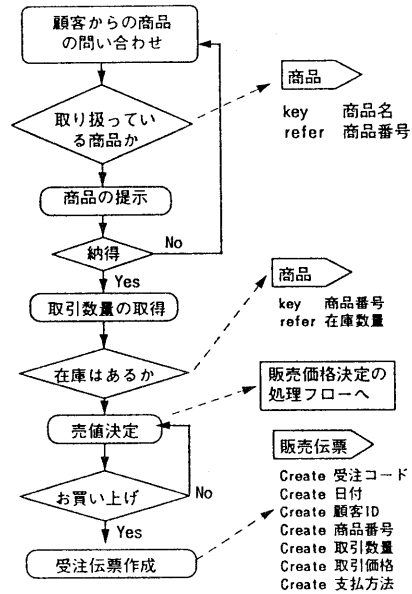


図2 a 受注の処理フロー

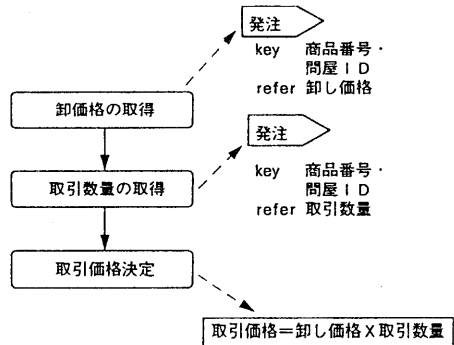


図2 b 卸値決定の処理フロー

もし、処理フローを描いてアイテムとオペレーションの相関表に誤りがあることが発見されれば、相関表の該当欄を訂正する。また、あるオペレーションを行うために必要なアイテムが新たに明らかになったときは、そのアイテムをアイテム表に追加し、その属性を決定する。さらに、そのアイテムがどのテーブル(エンティティ)の属すればよいのかを判断し追加する。もし判断できないならば、2. 3節で述べたアイテムのグループ化を行

い、再度エンティティの導出を行う。このようにして得られた処理フローの例を図2 a 図2 bに示す。

以上がセールスドメインのプロセスモデルである。

3 個々のシステム開発におけるドメインモデルの再利用と支援環境の考察

本節では本稿で構築したドメインモデルを個々のシステム開発に利用する指針を示す。まず、ドメインモデルからのシステム生成の全体図を図3に示す。

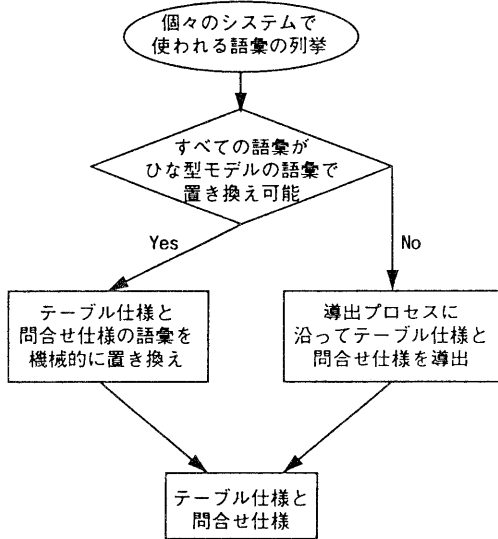


図3 セールスドメインのシステム生成プロセス

個々のシステムを開発するときは、プロダクトモデルの語彙をシステム特有の語彙に置き換えることによって、プロダクトモデルの持っているアイテムの属性やアイテム同士の関係、オペレーションの詳細などを再利用できる。アイテムの属性やその関係はセールス業務ドメインに共通の特徴だと考えられるので、大部分はプロダクトモデルのアイテムの属性などを利用することが可能である。開発者は与えられたアイテム関係表をみて、記述されている事柄が所望のシステムに適切か、適切でないかを判断し、不適切だとおもわれる箇所を修正すればよい。設計者はドメインモデルで示した開発のチェックポイントごとに検証を行い、必要であれば修正を行うという対話的な手順で開発を進めればよい。チェックポイント間の作業はドメインモデルで記述されており、適当な支援環境を整えると、計算機によって部分的に作業

の支援や作業の自動化が可能である。この開発支援方法を表5に示す。

表5 分析・設計手順の支援方法

分析・設計ステップ	支援方法
(a) 語彙の列挙	語彙辞書
(b) 語彙の分類	語彙辞書
(c) アイテムのグループ化	自動化
(d) テーブル仕様の決定	自動化
(e) オペレーションとアイテムの関係付け	ドメイン知識ライブラリ
(f) 問合せ仕様の決定	テンプレート

計算機をシステムの分析・設計の支援に用いるためには、分析・設計手順を可能な限り事細かに、手続的に記述することが不可欠である。また、はっきりした基準がない物事の判断は人間でなければ不可能であるので、分析・設計手順のどの部分に対してどのように計算機が支援できるのかを明らかにした表5の意義は大きい。

また、個々のシステムを開発する際、プロダクトモデルの持つアイテム群やオペレーション群では要求された仕様を満たせない場合、新しくアイテムやオペレーションを追加する。アイテムを追加する場合、まずそのアイテムの属性を決定する。次に、プロダクトモデルの持つエンティティ（商品、問屋、顧客など）に分類できるか否かを分析する。分類できるならばそのエンティティに新しいアイテムを追加するだけでテーブル仕様を導出することができる。既存のテーブルへの分類が難しい場合、また明らかに分類できない場合は、2節で述べているアイテムの関係付け、グループ化を行いテーブル仕様を導出する。この作業を行うことで、セールスドメインに欠くことのできない既存の5つのテーブル（商品、問屋、顧客、仕入れ、販売）以外のテーブル仕様の構築が可能である。本稿の例では、クレジットカードのテーブルがこれにあたり、このエンティティはプロダクトモデルのオプションとして位置付ける。さらに、アイテムが追加される場合はそれに伴い、問い合わせ仕様も修正する必要がある。特にアイテムを追加したことで、新しくテーブルが定義されたときは、新たなオペレーションを追加する必要がある。このときは、2節で述べた、オペレーションの分類、アイテムとの関連付けを経て、問い合わせ仕様を構築する。

新しく構築したオペレーション、アイテムとオペレー

ションの相関表, 問い合わせ仕様なども, すべてプロダクトモデルのオプションであると考えられる。

新しく得られたテーブル仕様や問合せ仕様およびそれらの導出手順で得られた中間生成物をセールスドメインのプロダクトモデルにオプションとして位置付けることで, プロダクトモデルが徐々に成長する。一度オプションとしてプロダクトモデルに取り込んだアイテムやオペレーションは, セールスドメインの次のシステム開発の時に再利用可能である。プロダクトモデルが十分に多くのアイテムやオペレーションを提供できるならば, 開発者は目的のシステムに要求される仕様をもとに, 必要なアイテムやオペレーションをプロダクトモデルが提供するアイテムやオペレーションから選択し, 語彙の対応付けを行うだけで, 所望のシステムを構築することができるであろう。

上述のように, セールスドメインのシステム開発にプロダクトモデルを利用し, 同時にプロダクトモデルを成長させることで, 後々のセールスドメインのシステムを構築が非常に容易に行えるようになると考えられる。

4 考察

本稿ではセールス業務ドメインのドメインモデルを構築し, それを実際のシステム開発に利用するための指針を示した。このセールスドメインモデルには, テーブル仕様や問い合わせ仕様の他に, セールス業務ドメインを分析する手順やプロセスも含まれており, これらはセールスドメインに含まれる個々のシステムを開発する際に再利用できる。個々のシステムのテーブル仕様や問い合わせ仕様を導出するために, このドメインモデルに含まれる分析の手順やプロセスを組み込んだ計算機による分析支援環境の実現可能性を示した。

本稿では, アイテムの分析に表を用いたが, これは[林95]に解説されているエンティティの分析方法を参考としたものである。アイテム同士の関係をもろさず考察するためにはこのような表を用いることが有効だと判断したからである。ただ, エンティティ同士の関係を記述することで終わりとせず, さらに分析を進めテーブル構成を導出する点, その手順を詳細に記述したことが筆者らの独自の点である。

今後, 本稿のドメインモデルの, 実際の規模に近いシ

ステム開発への適用を試みたい。

参考文献:

[Arango91] Arango,G., R.Prieto-Diaz: Introduction and Overview:Domain Analysis Concepts and Research Directions, pp.9-26, Domain Analysis and Software Systems Modeling (R.Prieto-Diaz and G.Arango ed.), IEEE,(’91).

[伊藤94]伊藤潔, 田村恭久, 杵嶋修三:Triadic Domain Modelに基づくシステムの分析・設計, FOSE’94, 日本ソフトウェア科学会, (Dec.,’93).

[伊藤95a]伊藤潔, 田村恭久, 杵嶋修三:ドメイン分析・モデリング概説, 情報処理学会ソフトウェア工学研究会,(Mar.,’95).

[伊藤95b]伊藤潔, 田村恭久, 杵嶋修三:ドメイン分析・モデリングの利用法・研究法, 情報処理学会研究報告95-SE-103, 情報処理学会, (Mar.,’95).

[Prieto-Diaz87] R.Prieto-Diaz: Domain Analysis for Reusability,COMPSAC’87, pp.23-29, (’87).

[田村94]田村恭久, 伊藤潔, 杵嶋修三:ドメイン分析・モデリング技術の現状と課題, 情報処理, (’94).

[田村95]田村恭久, 平井俊光, 伊藤 潔, 比較分析によるドメインモデル獲得手法の提案, 情報処理学会研究報告95-SE-103, 情報処理学会, (Mar.,’95).

[Tracz92]W.Tracz Domain Analysis Working Group Report-First Int.Workshop on Software Reusability,ACM SIGSOFT SE.Notes,Vol.17, No.3,pp.27-34,(’92).

[林95]林衛, DOAとRADのためのシステム分析・設計技法, ソフトウェアリサーチセンター, (Mar.,’95)