

# ドメイン分析・ モデリングとドメイン 指向システム開発

— 分野に適合したシステム開発のための構成要素 —

熊谷 敏 (株)山武  
伊藤 潔 上智大学

## システム開発とドメイン知識

システムを開発する組織（コミュニティ）は、企業であれ非営利の研究組織であれ、一般の汎用的な開発方法を調整したり最適化することが多い。また、その組織の独自の開発方法を考案し運用している場合もある。このときの開発方法は、仕様書や設計書のひな型や事例、ドキュメンテーションの標準、ライブラリ、ツール群、テストケース、組織編成等さまざまな形態をとる。またそのコミュニティの外部に対して、必ずしも明示的、形式的に示されず、いわゆる暗黙知として存在しているものもあるだろう。

そうした開発方法は、そのコミュニティがシステム化の対象としている特定領域（アプリケーションドメイン）において、システム開発を実践してきた集積として得られるものである。すなわち、そのコミュニティが対象としているアプリケーションドメインにおける問題の捉え方や解を得るための知識や経験を、いわゆるベストプラクティスとして獲得したことになる。

アプリケーションドメインは、その開発組織が関与している分野や業種や業態によってさまざまなものがある。例としては、事務処理、計測制御、リアルタイムシステム、診断システム、オンラインランザクションシステム、図書館業務、金融業務、スケジューリングシステム等があるが、いずれの場合もその組織が指向している市場ニーズ、提供する製品、採用している技術シーズに依存する傾向がある。そのどれかが異なっても、別のアプリケーションドメインとして認識すべきである。これは効果的な開発を行うために必要な知識や経験がそれぞれ異なるためである。

ドメイン指向システム開発の体系化の研究は、現在行われているアプリケーションドメインに密着した開発方法を、ドメイン分析・モデリングによって得られるドメインモデルを利用して整理しようとするものである。

## ドメイン分析・モデリング

ドメイン分析・モデリングは、同じドメインに属する対象システム自身の本来の各種の性質や開発上の多様な知識を認識し、システム開発に有効な固有なドメインモデルを得る技術である。

ここで代表的なドメイン分析・モデリングの定義を紹介する。Neighborsによると、ドメイン分析は「あるドメインで、そのドメインのエキスパートが重要と考えるオブジェクト、操作、関係付けを識別しようとする試み」と定義した<sup>3)</sup>。また、Prieto-Diazは、「新システムを構築する場合にこれまでの情報を再利用する目的で、ソフトウェアシステムの開発に使用した情報を識別し獲得し組織化するプロセス」と定義した<sup>4)</sup>。

これらの定義は主としてソフトウェアシステムにおける再利用を重点に置いたものだろう。この場合のドメイン分析・モデリングの効用は下記のように整理される。

- (1) ドメインごとに異なるモデルを採用するので、ドメインの特徴をより効率的に記述できる。
- (2) 汎用のソフトウェアライブラリではなく、ドメインごとにライブラリを蓄積するので、従来のソフトウェア再利用技術と比較して高い再利用率と品質が期待できる。  
すなわち既存のソフトウェアを部品化しても、そ

の用途が汎用的であったり、逆に1つのアプリケーション固有な問題に依存しすぎると、部品の再利用性は低くなる。このため、ドメイン分析・モデリングでは、同種のアプリケーション群からなる特定のドメインごとに、共通に必要な機能を抽出し、その機能に沿って部品化し、部品の再利用を促す。

この考え方を一歩進め、システム開発全体のスコープを捉え、要求定義から保守に至るさまざまなフェーズで、そのドメインの観点から情報を再利用できれば、従来に比べソフトウェアの生産性が一層向上するだろう。すなわちソフトウェアの部品の再利用だけでなく、要求分析や設計等の他の開発フェーズでも、ドメイン分析・モデリングは活用できよう。

本稿では、システム開発全体のスコープを捉え、ドメイン分析・モデリングを次のように考える。ここでいうシステムは、計算機によるシステムだけではなく、人間系をも含めた、一般に「業務」と呼ばれるものであってもよい。「対象システム自身が本来持つ各種の性質や開発上のさまざま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、システムの作り方などの、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」<sup>6)</sup>

この場合のドメインモデルは、そのドメインのシステムを繰り返して開発する際のひな型として使われ、システム開発全体の生産性と再利用性を向上させることをねらいとする。

## ドメイン指向開発環境

ドメイン指向開発環境とは、システム要求分析、システム設計、システム実装などのシステム開発フェーズで、ドメインに固有な用語集、問題解決の方針、仕様記述言語、ツール、ソフトウェアコンポーネント、ライブラリなどを使って、効率的にシステム開発を行う、開発手法、開発ツール、環境の統合的な集まりをいう。

ドメイン指向開発におけるドメイン分析・モデリングの役割を図-1に示す。

ドメイン指向開発におけるドメイン分析・モデリングの対象は、ソフトウェアの実装レベルの再利用にとどまらない。ドメイン指向開発の目的を整理すると以下の3点になる。

(1) ドメインに属する過去の生産物（ソフトウェア自体、ソフトウェア部品、開発フェーズでの仕様

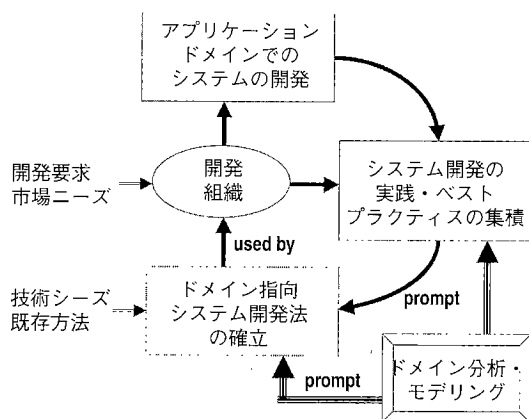


図-1 ドメイン指向システム開発のサイクル

書等)を再利用する。

- (2) ドメイン特有/固有の開発プロセス・方法・ツールを体系化/明確化/改良/創生/再利用する。
- (3) 組織内での開発事例/ノウハウ/経験/方法を収集/体系化/教育/継承/再利用する。

## ドメインモデルの構成要素

ドメインモデルには、大きく分けて、ドメイン記述 (domain description)、ドメイン辞書 (domain dictionary)、ドメイン表現 (domain representation)、ドメイン開発諸機能 (domain facilities)、ドメイン開発制約 (domain constraint)、がある。

ドメイン記述は、まだ完全には整理しきっていないが、ドメインの基本的な性質を明らかにしたもので、ドメインがどのようになっているか、どんなものを使っているかを列挙したものである。ドメイン記述の対象となるドメイン情報には以下のものが挙げられる。

- 目的業務
- 業務マニュアル
- そのドメインの教科書、パンフレット、用語集
- ドメインエキスパートへのインタビュー
- ビジネスの現状
- ドメインの境界、スコープ、近隣ドメイン
- 使用可能なテクノロジー
- D-AMEの適用範囲
- 開発事例
- 既存のシステム (パッケージ)
- 既存のシステムの開発担当者へのインタビュー
- 既存の仕様書、マニュアル
- 実現プラットフォーム
- ひな型アーキテクチャ
- 法規
- コスト制約

ドメイン記述は特に形式性にはこだわらずに自然言語等で記述してもよい。ドメイン記述の例を示す。

- 業務知識の列挙
- 業務の機能の（階層的）構造や関係の列挙
- 業務ノウハウ
- 業務担当者の列挙
- システムフロー
- エンティティ、イベント、データ、オペレーション、関係、属性の列挙
- システム間の類似点、共通点、相違点の列挙
- システム間のバリエーションの列挙
- ビジネスルール
- 関連ドメイン
- 性能／信頼性のレベル
- 変更要求の傾向

ドメイン辞書は、そのドメインでよく使う用語、類語、用語対応などを把握したドメインの用語集である。ドメイン辞書には、この分野ではこの用語を使うことが望ましい、この分野では、用語の意味はこうである、ということが書かれる必要がある。用語には、ドメインモデルでの論理名（ドメイン用語）を使う。表層的な言語や用語を代えても、モデルの論理構成が変わらないことが重要であるので、ドメイン用語は、ドメインモデルの論理構成を代表する用語を用いることが望ましい。ドメイン辞書の例を示す。

- ドメイン共通の用語の列挙
- システム間の用語の対応表
- 近隣ドメイン間の用語の対応表

ドメイン表現は、ドメインの開発に使われたり必要である組織化、構造化、形式化した言語やチャートなどである。ひな型モデル、パラメータ化モデル、参照モデル、標準モデルなどのチャート、ダイアグラムが相当する。一般にドメインモデルというと、このドメイン表現のみを指すことがある。ドメイン表現を狭義のドメインモデルとすれば、上記のドメイン記述は広義のドメインモデルとなる。同じドメインにおいても対象とする局面によって、ドメイン表現が異なることもある。ドメイン表現の例を以下に示す。

- ドメイン向き記述言語（ドメイン言語）
- ドメイン向き記述チャート
- 穴埋め式テンプレート／フォーム
- オブジェクト図
- 項目表
- オペレーション一覧
- 業務モデル

- 挙動モデル
- 工程モデル
- 参照モデル
- 標準モデル
- ネットモデル
- 標準アーキテクチャ
- 計算モデル

ドメイン開発諸機能はドメインに属するシステムの開発を効率化するためのさまざまな機能、方法、開発環境等である。ドメイン開発諸機能の例を以下に示す。ドメインに属するシステムの実装フェーズばかりではなく、システムの要求分析や設計フェーズに活用される開発諸機能も入っていることに留意したい。

- 機能部品／エンティティ部品／オブジェクト部品のパッケージライブラリ
- パッケージライブラリの使用手順／カスタム化のガイドライン
- ドメイン指向の開発方法論／開発標準／開発プロセス
- システムの仕様の書き方のガイドライン
- システムの要求分析／システム設計／実装のマニュアル／ガイドライン
- システムの要求分析／システム設計／実装のハンドブック／チェックリスト
- ドメイン指向の専用CASE/CAD
- エンドユーザプログラミング言語と処理系
- アプリケーションジェネレータ
- ドメイン指向のシステム開発のナビゲータ
- 典型的システム（プロトタイプ）
- データディクショナリ
- 成功／失敗の開発事例集

ドメイン開発制約は、そのドメインに属するシステムを開発するときの前提や制約である。現実のシステムの仕様では、対象ドメインにとって本質的な性質と、実装上またはその他の流動的な条件に基づく制約は異なる。前者は上のドメイン記述やドメイン辞書に含まれ、後者がドメイン開発制約となる。ドメイン開発制約の例を以下に示す。

- 実装プラットフォーム制約
- 処理能力
- 性能・信頼性
- 開発組織の制約
- 外部とのインタフェース制約
- 法規、基準
- プロトコル

以上のようにドメインモデルの記述手段にはさまざまな方法がある。Arangoは、ドメインモデルが強調して記述する側面で、それらの記述手段を次のように分類している<sup>1), 5)</sup>。

- (1) 宣言的／定義的モデル（構成要素の抽出，列挙，分類を記述）：分類，クラス分け，シソーラス（同義語）等がある。
- (2) 構造的モデル（構成要素間の関係を明らかにする）：概念グラフ，ER図等がある。
- (3) 機能的モデル（システムや個々の構成要素のふるまいを記述）：SADT，状態遷移図，データフロー図等がある。

## ドメインモデルの使い方

ドメイン分析・モデリングにおける再利用という言葉の意味は、過去のシステム群や開発の実践をドメインモデルとして集積し、そのドメインモデルを使って、過去のシステムや開発経験を再利用することである。

ソフトウェアの再利用の分野では、いくつかの限定されたアプリケーション分野で、部品ライブラリを用意しておき、ユーザが仕様記述すればジェネレータにより、ターゲットのプログラムが生成される方法が採られている。この関係を図-2 (a) に示す。

ドメインモデルを、システム開発における設計や要求分析に活用する場合も、図-2 (b), (c) のように同じ構造で捉えることができる。すなわち、ドメインモデルの対象は、実装レベルにとどまらず、要求／設計仕様書のひな型などの分析／設計レベルもあり得るため、ライブラリもソフトウェアパッケージに限らず、仕様書ライブラリ、パターンやテンプレートの場合もある。

図-2に示す入力と出力の関係は、ドメインモデルの問題構造と解構造との関係と捉えることもできる。

図-2 (b), (c) では、それぞれ左から入る記述用のドメイン言語があり、右へ出るドメインプロダクトがある。ドメイン言語は、そのドメインで明らかなオブジェクト群や機能群とそれらの属性・ふるまいを関係付けて、ドメインの仕様を記述する言語である。ドメイン言語も、テキスト形式もあれば、分析／設計向けのチャートもある。

下から入る矢印は、ドメインライブラリ（さまざまな呼び方あり）である。これは、解である要求仕様や設計仕様を得るための変換に用いるドメイン固有の標準ライブラリである。

中央のボックスは、ドメイン向き変換プロセスである。この変換プロセスは、自動のジェネレータも

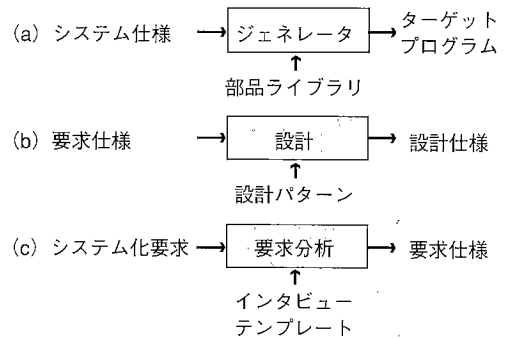


図-2 ドメインモデルの再利用

あれば、半自動やすべて人手による開発プロセスということもあり得る。その場合は、ジェネレータというより、要求分析／設計の仕方となる。

## ドメインモデル作成の勘所

ドメインモデルを得るための汎用的な方法を定義するのは容易でない。システム開発の実践の中で培われるベストプラクティスは、一朝一夕で得られるものではない。また、いわゆるウォーターフォールモデル的に、各工程の成果として、ドメインモデル、ドメイン記述、ドメイン辞書、ドメイン表現、ドメイン開発制約等を定義したとしても、その順序で、ドメインモデルが構築されるとは限らない。あるドメインでは、ドメイン辞書をまず作成することが重要である場合もあれば、別のドメインでは、ドメイン記述があらかじめ与えられることもある。また工程間では多くの行きつ戻りつのプロセスが発生し、その発生の仕方もドメインによって異なるであろう。

ここでは、ドメインモデル作成の勘所として下記を強調しておきたい。最も重要なのは、ドメイン分析・モデリングを始める際にドメインを設定し目標を設定することである。目標設定をしないで漫然と分析を始めてはならない。

### (1) ドメインの選択と情報源の設定

既存システム、マニュアル、ドメインエキスパートとのインタビュー、業務現況、業務テキスト等に着目する。これらがそのドメインを事実上規定しており、ドメインモデル作成の情報源となる。

### (2) システム開発で何を共有／再利用するか

ドメイン指向開発では、再利用の対象は実装レベルだけでなく、要求／設計仕様も含まれる。仕様書のライブラリ、パターン、テンプレート等の再利用する対象を設定する。

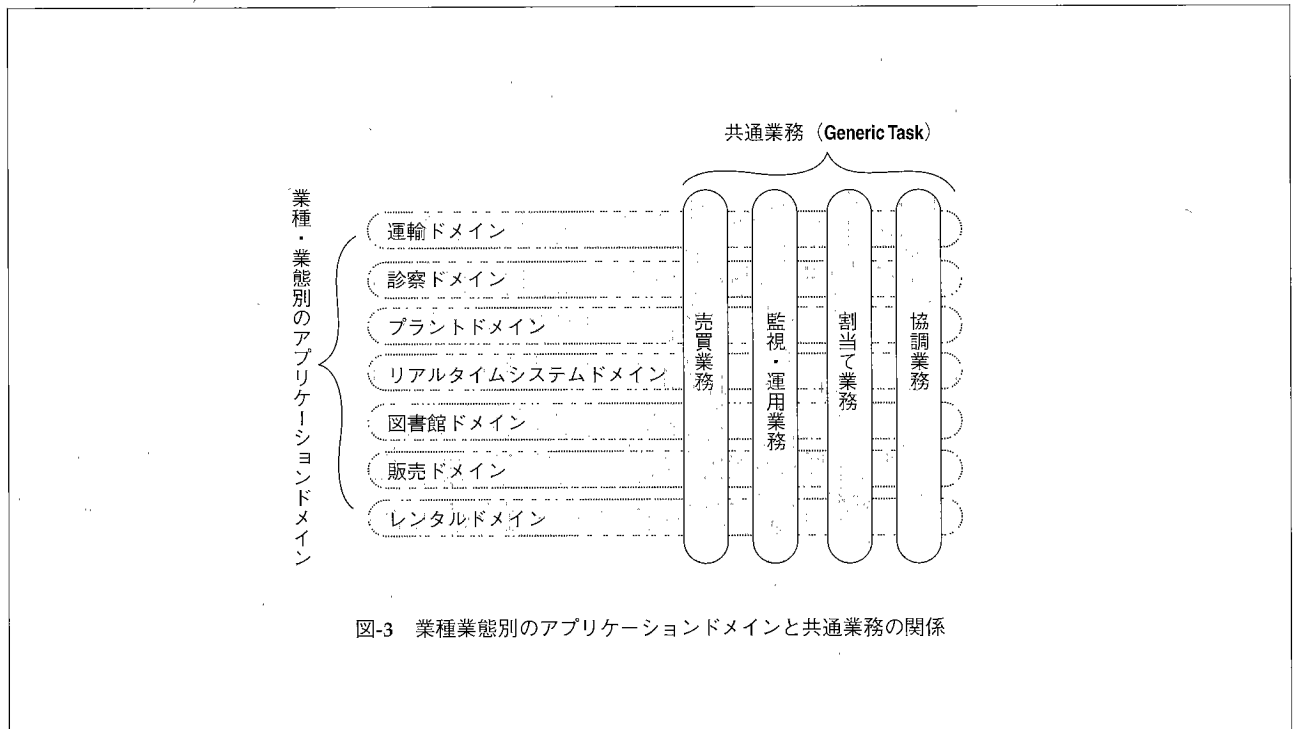


図-3 業種業態別のアプリケーションドメインと共通業務の関係

(3) 誰が共有／再利用するか

情報を共有／再利用するものは、情報処理技術者のみの場合もあれば、業務エキスパートの場合もある。共有／再利用の対象によって、ドメイン辞書の内容も変わってくる。

(4) 作成するドメインモデルは問題／プロセス／プロダクトのどのモデルか

ドメイン分析・モデリングでは、解くべき問題とその問題の解であるプロダクトを明確に区別することが重要である。前者をプロブレムモデル、後者をプロダクトモデルと呼ぶ。すなわちプロブレムモデルは、システム化の目標となる問題群 (a class of problems) であり、プロダクトモデルはシステム化した解群 (a class of solutions) である。さらに両者を結びつけるプロセス表すものをプロセスモデルと呼ぶ<sup>2)</sup>。これらの関係は、図-2の入出力と捉えてよい。これらの関係を明確にすることがドメインモデルを利用するときの前提となる。

(5) ドメインモデルはシステム開発の分析／設計／実装のどこで使用するものか。

上に述べたように、ドメインモデルは、分析／設計／実装のそれぞれのフェーズによって、問題モデル、プロダクトモデル、プロセスモデルが異なるため、それらを明確にする必要がある。

(6) 何を開発諸機能とするか

ドメイン分析・モデリングを開始する場合には、ドメインの開発諸機能をまず明確にすることが現実的なアプローチとして望ましい。これは開発諸機能

が、そのドメインを事実上、規定しているためである。どのような開発諸機能となるかが、ドメイン・分析モデリングの当初は不明な場合でも、自分たちがどのような開発諸機能が必要としているかを明確にすることは、目標として具体的であり把握しやすい。

開発諸機能が、パッケージライブラリ、CASEツール、ノウハウ集等さまざまな形態になるにせよ、ドメイン分析・モデリングのゴールとして設定すれば、そこに必要なドメイン辞書、その開発諸機能を必要とするドメイン開発制約、開発諸機能の前提となるドメイン表現などを把握しやすくなることが多い。

## これからのドメイン分析・モデリング

ドメインモデルは、特定コミュニティ内部での再利用を最優先に作成される。したがって、そのドメインのコミュニティが指向する特定の業種、業態に依存するモデルとなる傾向がある。そうしたモデルは、そのままの形で別の異なるドメインに対して適用しようとしても困難である。これはコミュニティにおける慣習上の違いや用語の違いが、実は表層的なものにもかかわらず、業種、業態ごとに構成されたドメインモデル間では、本質的な違いに写ることがあるためである。また、業種、業態に強く依存したドメインモデルは、近隣のドメインモデルには適用可能な場合があるかもしれないが、どのくらいの“近さ”ならば適用可能かを判断する指標は存在しないので、具体的にどの近隣ドメインに適用するか

を判断することは容易でないだろう。

ドメインモデルを、業種や業態、組織が指向している市場ニーズ、提供する製品、採用している技術シーズごとに異なるとして列挙しようとするれば枚挙にいとまがない。また個々のアプリケーションドメインは細部が異なるので、そこにおける方法もそれぞれ最適なものにすべきである。しかし複数の異なるアプリケーションドメインに対して横断的に共通する要素は存在することが分かってきている。

筆者らがこれまで進めてきたさまざまな業種、業態別のアプリケーションドメインの分析の中で、協調業務 (Collaboration Task)、売買業務 (Trading Task)、割当て業務 (Assignment Task)、監視・運用業務 (Monitoring and Operations Task) 等の同種の業務が行われていることが認識されている。

これらの共通業務は、図-3に示すように、業種・業態別のアプリケーションドメインとは直交関係にある。

共通業務を1つのアプリケーションドメインと考えることもできる。共通業務ごとのドメインモデルを業種や業態に依存したアプリケーションドメインで活用しようとする場合にも、図-2に対応するような問題構造、解構造ならびにドメイン固有の変換プロセスの関係を認識できる。

共通業務ごとのドメインモデルは、この変換プロセスに着目する必要がある。これは、問題構造や解構造を表現するドメインモデルが、業種や業態ごとのドメイン用語 (辞書) により強く依存するものと考えられるためである。したがって、共通業務のドメインモデルは、よりプロセスを指向したものと捉え、Generic Taskと呼ぶのがふさわしいだろう。“Task”という名称は、プロセスを連想させることを意図して採用したものである。

Generic Taskごとのプロセスモデルは、ドメインをまたがっても再利用できるプロセスモデルとなる。業種、業態ごとの表層的部分が、ドメイン辞書を入れ替えることによってGeneric Taskのプロセスモデルで吸収されると期待できる。Generic Taskは特定の組織や業種、業態に依存するものではなく、共通の業務として存在する。これは汎用モデルではないが、Generic Taskごとに適用できる有効なモデルとなる。また、ドメインを固定すると、そのドメインの1つのGeneric Taskのプロセスモデルは、そのドメインのその業務のシステムを開発するプロセスのモデルとなり、支援システムのひな型として利用できよう。

従来の業種や業態ごとのドメインモデルはGeneric Taskドメインモデルの集合として表されるだろう。これは、Generic Taskドメインモデルの

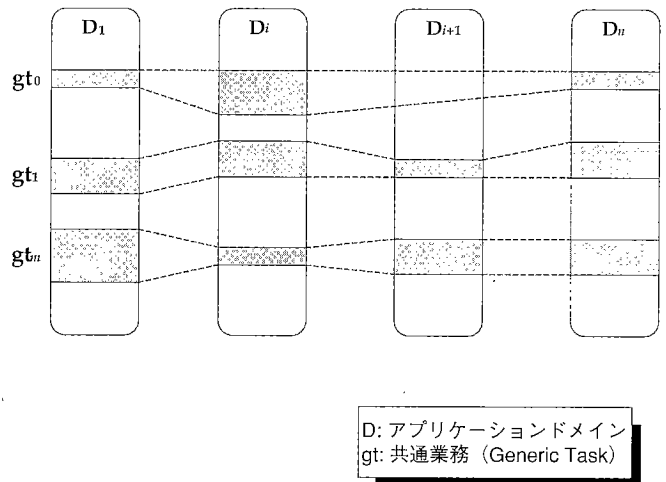


図-4 アプリケーションドメインに点在する共通業務 (Generic Task)

単純な和集合として業種や業態ごとのドメインモデルが表されるということではない。むしろGeneric Taskは、各ドメインに部分的に点在するものであろう。すなわち共通業務は個々のアプリケーションドメインに部分的に混在し、それぞれ存在している割合は、各アプリケーションドメインごとに異なる。たとえば図-4のようにアプリケーションドメイン $D_i$ と $D_n$ では、同じ種類の共通業務 ( $gt_0, gt_1, gt_m$ ) を含むが、 $D_i$ のほうがより $gt_0$ という共通業務の側面が強い。 $D_{i+1}$ のように、 $gt_0$ という共通業務の側面がないドメインもあるだろう。

Generic Taskごとにドメインを分析し、いくつかのGeneric Taskを統合して、個々のドメインモデルを獲得しようとするアプローチは実務的にも有用であると考えられる。またGeneric Taskを認識し、それを基礎として構築した方法は、さまざまなドメインで有用なものとなり、システム開発に関する知識の流通を容易なものとするだろう。

参考文献

- 1) Arango, G. and Prieto-Diaz, R.: Introduction and Overview: Domain Analysis Concepts and Research Directions, pp.9-26, Domain Analysis and Software Systems Modeling, Prieto-Diaz, R. and Arango, G. (ed.), IEEE (1991).
- 2) Itoh, K., Hirota, T., Kumagai, S. and Yoshida, H.: Domain Oriented Systems Development: Principles and Approaches, Gordon and Breach Science Publishers (1998).
- 3) Nighbors, J. M.: Software Construction Using Components, Ph.D. Thesis, Univ. of California Irvine (1981).
- 4) Prieto-Diaz, R.: Domain Analysis: An Introduction, ACM SIGSOFT, Software Engineering Notes, Vol.15, No.2, pp.47-54 (1990).
- 5) Tractz, W.: Domain Analysis Working Group Report - First Int'l Workshop on Software Reusability, ACM SIGSOFT Software, Engineering Notes, Vol.17, No.3, pp.27-32 (1992).
- 6) 伊藤 潔, 杵島修三, 田村恭久, 廣田豊彦, 吉田裕之 (編著): ドメイン分析・モデリング, 共立出版 (1996).

(平成 11 年 10 月 29 日 受付)