

Domain Specific Software Process(DSSP) の提案

伊藤 潔, 田村恭久(上智大学), 杵嶋修三(山武ハネウエル)

1. ドメイン分析・モデリングの定義と効用

ソフトウェアやシステムの開発の基幹技術として、ドメイン分析・モデリング技術の重要性が認識されている。Neighborsは、ドメイン分析を「あるドメインで、そのドメインの専門家が重要と考えるオブジェクト、操作、関係づけを同定しようとする試み」と定義した[Neighbors81]。また、Prieto-Diazは、「新システムを構築する際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が開発される際に使用した情報を同定し獲得し組織化するプロセス」と定義した[Prieto-Diaz90]。

ドメイン分析でのドメインは、複数の類似した問題やソフトウェアが属する対象領域を指す。これらの問題群やソフトウェア群とドメインの関係は、オブジェクト指向のインスタンス群とクラスの関係とほぼ同一である。即ち、個々の問題やソフトウェアは、ドメインというクラスのインスタンスとみなせる。ドメインの性質や挙動は、問題群やソフトウェア群を分析し、それらに共通する要素を明らかにする事で得られる。

ドメイン分析・モデリングでは、ソフトウェア開発での「解くべき問題」と「その問題を解決するアプリケーション」を区別することが有効である。[Tracz94], [Schoen91], [Lubars91], [Neighbors89], [Garlan93a]等を総合すると、問題ドメインとアプリケーションドメインという用語で区別するとよいと考える。問題ドメインは、類似した複数の問題から、共通した構成要素、関係、属性、振る舞い等から成り、いかにして解かれるべきかという情報を含まない。一方、類似した問題群に対しては、類似したソフトウェア群が開発される。アプリケーションドメインは、このソフトウェア群が共通してもつ構成要素、関係、属性、振る舞い、ソフトウェア部品等を抽出したものから成り、解くべき手順、方法、仕掛け等の、問題自体には含まれない情報を含む。

筆者らは、「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95]。通常は、ドメインに依拠してシステム開発を効率的に行うために、解くべき問題群を表すためのドメインモデルと、解群を表すためのドメインモデルが必要であると言われているが、筆者らは、これに加えて、前者から後者を導出するプロセスを表すドメインモデルを含めた、Triadic Domain Model(三つ組のドメインモデル:TDMと略す)を提案した[Itou94ab][伊藤94]。このプロセスを表すドメインモデルに基づくソフトウェアプロセスを、筆者らは、ドメイン指向ソフトウェアプロセス(Domain Specific Software Process: DSSP)と名づける。

2. Triadic Domain Model: TDM

Triadic Domain Model(TDM)は、次の3種類の構成要素から成る。(表1, 図1参照)

- ・ Domain Problem Model
- ・ Domain Product Model
- ・ Domain Process Model

Domain Problem ModelとDomain Product Modelは、1節で述べた「問題ドメイン」と「アプリケーションドメイン」とほぼ同様である。Domain Process Modelを明確に認識する事が、TDMの特長である。

Domain Problem Modelは、類似する問題群から得られる、共通の構成要素、関係、属性、振る舞い等から成る、「解くべき問題」のドメイン固有のひな型モデルである。Domain Problem Modelは、何らかのダイアグラム、フォーム、テンプレート、シナリオ、穴埋め型文書等で記述される。これらのいずれを用いるかはドメイン固有である。また、これらの記述に使用する語彙、ある項目に書き込むデータタイプや数値の範囲、構成要素間の相互関係、個々の構成要素の状態遷移等のダイナミック

表1 Triadic Domain Modelに必要な諸項目

モデルを特徴づける諸要素	Domain Problem Model	Domain Process Model	Domain Product Model
モデル化対象	problem domain	development process domain	application domain
モデルの獲得法	同種のシステムの開発における共通な問題、目標、役割の集まりの分析	同種のシステムの開発過程の分析	同種のシステムの開発におけ共通なシステム製品（仕様/プログラム）の集まりの分析
識別すべきオブジェクト	典型的な問題の構成要素	システム開発の典型的な作業要素	典型的な仕様/プログラムの構成要素
識別すべき属性	用語、データの種類とタイプ、値の範囲、相互関係	作業要素間の合理的な順序	ライブラリ、振舞、パラメータ、アルゴリズム、データ構造、資源、実現上の制約
使うべき経験則	ドメインの範囲、ドメインとインスタンスの近さの識別	作業要素間の経験的な順序	制約とコストに基づくトレードオフ・推奨・選択の基準
表現のスタイル	自然/仕様/プログラミング言語、ダイアグラム、穴埋めフォーム、テンプレート、シナリオ	口頭/非形式的/形式的手続き	自然/仕様/プログラミング言語、ダイアグラム、穴埋めフォーム、テンプレート、シナリオ

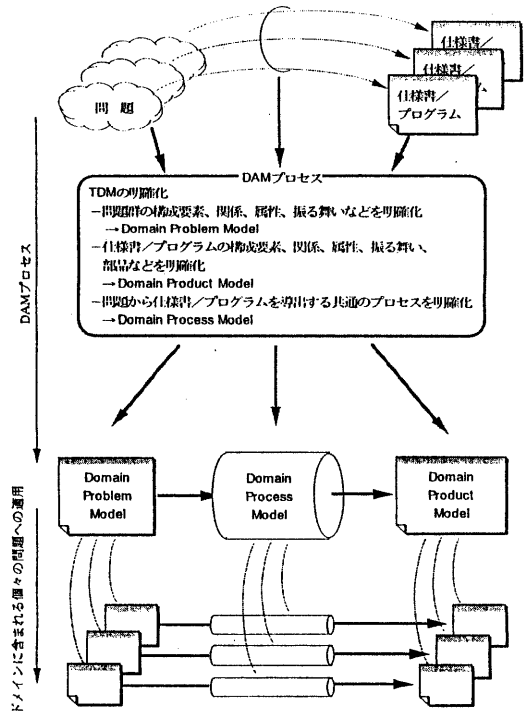


図1 Triadic Domain Modelの概念

クも、ドメイン毎に固有である。Domain Problem Modelは、問題の記述のみから成り、いかにして解かれるかという解法の記述は含まない。

Domain Product Modelは、類似する複数のシステムの開発により得られる、仕様書群やソフトウェア群が共通してもつ構成要素、関係、属性、振る舞い、ソフトウェア部品等を抽出したものから成る、「解くべき問題の解」のドメイン固有のひな型モデルである。より具体的には、実現方法、アルゴリズム、データ構造、資源、ライブラリ等の、Domain Problem Modelには含まれない、アプリケーションの仕掛けや構造を反映したものも含む。

Domain Process Modelは、複数のシステム開発事例から得られる、あるDomain Problem Modelに属する問題群から、そのProduct Modelに属する解群を導出する、開発手順のドメインに固有のひな型モデルである。Domain Problem Modelをフォームで記述したと考えると、Domain Process

Modelは「記述する項目の順序付け」になる。この順序付けは、全ての項目間で指定はされない半順序である。順序づけは、分析、仕様化、設計、又はインプリメンテーションでのドメイン固有の手順のひな型になる。この順序づけに加えて、採用すべき資源、ライブラリ、アルゴリズム、データ構造

の、候補、推奨、選択法、チューニング法、性能予測等の基準も、Domain Process Modelの構成要素である。

Domain Process Modelも、Problem ModelやProduct modelと同じく、ドメイン分析・モデリングの汎用手順を表すプロセスモデルではなく、個々のドメインに固有な開発手順のモデルである。また、このDomain Process Modelは、他のドメイン分析・モデリング研究では明確には認識されていない。このDomain Process Modelに基づくソフトウェアプロセスを、筆者らは、ドメイン指向ソ

ソフトウェアプロセス(Domain Specific Software Process: DSSP)と名づける。DSSPの位置づけを図2に示す。

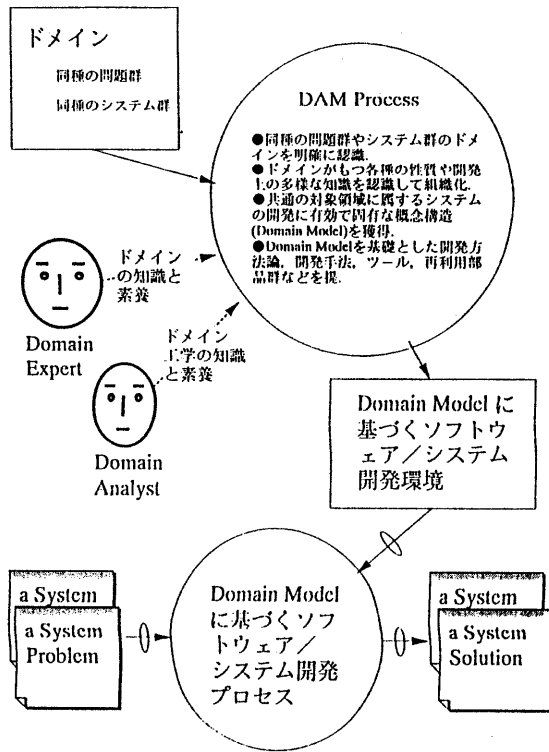


図2 ドメイン指向ソフトウェアプロセス (DSSP)

TDMは、要求分析、設計等の様々な開発フェーズに適用できる。例えば、要求分析のフェーズに適用するならば、Domain Problem Modelはユーザのもつインフォーマルな要求のモデル、Domain Product Modelは定式化された要求仕様のモデルになる。一方、設計フェーズにTDMを適用した場合は、Domain Problem Modelは要求仕様のモデル、Domain Product Modelは設計仕様のモデルになる。

図1にドメイン分析・モデリングによるTDMの生成とTDMによるソフトウェア開発を示す。まず、あるドメインに属する複数の問題の仕様と、それらに対応する解の仕様を入力として、問題群から抽出されたDomain Problem Model、解群から抽出されたDomain Product Model、解群を導出する共

通のプロセスであるDomain Process Modelの3つを生成する。次に、得られたTDMをひな型として同じドメインに属する他の問題の記述を効率的に行ったりその問題に対する解を導出する。

あるドメインでのドメイン分析・モデリングの事例や経験が少ない場合や、ドメインが変化した場合には、ドメイン分析・モデリングプロセスは、そのドメインに有効な、何らかの定型フォームやテンプレートを作成する作業と対比できる。即ち、ドメイン分析・モデリングプロセスの中でドメインの構成要素、構成要素間の関係、属性、振る舞い等を明らかにする。また、解群の中で再利用可能な構成要素があれば、それを抽出する。また、記述に使用する語彙も、ドメイン毎に固有に存在する。これは、フォームのある項目に書き込むデータタイプや数字の範囲が、書き込む以前に規定されている事と対比できる。

ドメインが変化したととらえる場合には、以前行ったドメイン分析・モデリングで得られたTDMとの差を明らかにする。これは、例えば、以前得られたフォームのチェックを行う作業である。この差異は、時間の経過で対象ドメインが変化した場合に生じやすい。例えば、銀行オンラインシステムでは、要求される機能や性能、信頼性等は数年単位で変化する。このような要求の変化に追従するため、過去にドメイン分析・モデリングが行われたドメインであっても、再度ドメイン分析・モデリングプロセスを通過する必要がある。

3 TDMの例示

TDMの概念に基づいていくつかのプロセスモデルを構成した。これらのドメインの現実のシステムは、膨大な量のソースコードから成る。そのコーディングの段階までの全てをTDMでカバーするのは無理であるが、要求仕様の明確化や概要設計の工程までは、有効に適用可能であると考えられる。

TDMの例として、筆者らはreactive systemの分析と設計についてのドメインモデルAsdreas STD Triad [杵嶋93][Itoh94ab][伊藤95]を考案し、リアルタイムシステムの性能評価・改善向きドメインモデルを[田村95]で考察した。

4 おわりに

ソフトウェアやシステムの開発の基幹技術として近年認識されているドメイン分析・モデリング

技術について、ドメインに依拠してシステム開発を効率的に行うために、解くべき問題群を表すためのドメインモデル、解群を表すためのドメインモデル、および、前者から後者を導出するプロセスを表すドメインモデルから成るTriadic Domain Model(三つ組のドメインモデル:TDM)を提案した。

TDMとDSSPに基づくシステム開発は、実際の開発作業からは遊離せず、システム開発での様々な問題をドメインという枠組みで見直す。これにより、実際の開発作業が効率的になり、システム製品の品質が向上すると期待できる。

TDMとDSSPでとらえることが、ソフトウェア開発でのドメイン分析・モデリングの有効な利用法と考える。また、一般化したソフトウェアプロセスばかりではなく、実務的には価値があると考えられるDSSPの認識を研究として確立していくことが重要である、と考える。

参考文献

- [Garlan93a]D. Garlan: Domain Specifications Require First Class Connectors, IEEE RE' 93, pp. 78, (1993).
- [Itoh94a]Itoh, K., Kishima, S., Tamura, Y.: Systems Integration on Specification, Design and Generation of Reactive System - Triadic Domain Model-Based Approach -, 3rd IEEE ICSI, (1994).
- [Itoh94b]K. Itoh., Y. Tamura., S. Kishima.: Triadic Domain Model-Based Development of Software Systems, IEEE ICSR' 94, (Nov., '94).
- [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95]伊藤潔, 田村恭久, 杵嶋修三: ドメイン分析・モデリング技術概説, 情報処理学会ソフトウェア工学研究会, SIGSE-103, (1995).
- [杵嶋93]杵嶋修三, 伊藤潔, リアクティブシステムの分析・設計向きドメインモデル: Asdrean STD Triad, 情報処理学会論文誌, Vol. 34, No. 9, pp. 2025-2036 (Sept. '93).
- [Lubars91]Lubars, M. D.: Domain Analysis and Domain Engineering in IDEa, Domain Analysis and Software Systems Modeling (R. Prieto-Diaz and G. Arango ed.), IEEE, (1991).
- [Neighbors81]Neighbors, J. M.: Software Constructing Using Components, Ph. D. Thesis, Univ. Calif. Irvine (1981).
- [Neighbors89]Neighbors, J. M.: DRACO: A Method for Engineering Reusable Software Systems, Softw. Reusability, Vol. 1, ACM, pp. 295-320, (1989).
- [Prieto-Diaz90]Prieto-Diaz, R.: Domain Analysis: An Introduction, ACM Sigsoft Softw. Engi. Notes, Vol. 15, No. 2, pp. 47-54, (1990).
- [Schoen91]E. Schoen: Active assistance for domain modeling, 6th IEEE Annual Knowledge-Based Softw. Engi. Conf., pp. 26-35, IEEE, ('91).
- [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, (1994).
- [田村95]田村恭久, 伊藤潔: リアルタイムシステムの性能評価・改善向けドメインモデルの提案, 情報処理学会ソフトウェア工学研究会, IPSJ SIGSE-102, (January, '95).
- [Tracz94]Tracz, W., Poukin, J.: A Domain Specific Software Architecture Tutorial, 3rd ICSR, (1994).