Vol.35 No.10 情報処理 Oct. 1994

解説

ドメイン分析・モデリング技術の現状と課題†

田村恭久†† 伊藤 潔†† 杵嶋修三†††

1. はじめに

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**(Domain Analysis and Modeling:DAM と略す)技術の重要性が近年認識されている．これは，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**: domain)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．

現在の開発手法，ツール，ソフトウェア部品の多くは，標準的な作業規約に基づいて，一般的な業務処理のソフトウェアの正確な開発を目指して作られている．中には，対象ドメインの認識が必ずしも十分ではなかったり，あまりにも汎用性を追求しようとして，適用の容易でないものもある．実際のシステムの開発現場では，これらの汎用的な手法やツールを，各々の対象ドメインのシステム開発にやむを得ず適用しようとして，非効率さと不具合を生ずる問題がある．また，汎用的であるがため，各々の対象ドメイン向きの再利用部品群の同定が容易でないという問題もある．さらには，これらの汎用的な手法やツールの適用を断念して，開発現場でこれまで長く採られてきた手作業を，そのままの手順でコンピュータ化するにとどまっている場合も少なくない．

システムの開発では，単品として新たに開発されるものは高々2割程度といわれる．多くの場合，過去の同種システムの開発で蓄積された技術，経験，ソフトウェア資源を有効に活用して，次のシステムの開発を図ろうとするのが現状である．この現状を踏まえた上で，DAM 技術は，同種のシステム全体の対象ドメインを明確に認識し，その対象ドメインに固有な性質や知識を包含したドメインモデルを構成し，このドメインモデルを基礎とした記述言語，開発方法論，開発手法，ツール，再利用部品群等を提示しようとするものである．

主にソフトウェア再利用のワークショップ(Intern. Workshop on Software Reuse, Dortmund 1991, Lucca 1993:IWSR と略す)を中心とする現在までの DAM 研究では，事務処理を主体とする定型業務のシステムを対象とすることが多い．これらのシステムは，対象ドメインを比較的認識・記述しやすく，ライブラリ等で対象ドメインからアプリケーションを導出する手順も比較的単純である．これに対し，大規模で複雑なシステムでは，そのドメインを記述するための言語に高い記述能力が要求され，また再利用するライブラリも複雑多岐にわたる．さらに，個々のアプリケーションを導出する手順も複雑になる可能性がある．

本稿では，2. でドメイン分析の基本概念とアプローチ，3. ではソフトウェア工学の他の技術との関連，4. で，主要な技術課題を述べる．

2. DAM 技術の基本概念

一般にドメインは，領域または分野と訳される．本稿でのドメインは，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．すなわち，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

† Current State and Problem of Domain Analysis and Modeling by Yasuhisa TAMURA and Kiyoshi ITOH (Sophia Univ.) and Shuzo KISHIMA (Yamatake Honeywell Corp.).
†† 上智大学
††† 山武ハネウェル(株)

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ことで得られる．

例えば，A銀行・B銀行のオンライン処理等，解くべき問題が複数ある時，これらからオンライン処理問題のドメインを抽出できる．銀行には，顧客，口座，支店等の構成要素，また預金の預け入れや払い戻し，借り入れ等の動作，さらに口座番号や利率等の属性がある．これらの共通概念がオンライン処理問題のドメインを構成する．

顧客数，支店数，サービス内容，利率等は銀行ごとに異なる．オンライン処理のドメインは，これらの要素を可変のパラメータとして扱う．あるドメインから個々の問題を導く場合には，個々の問題に特有な構成要素を付加したり，パラメータに問題に固有の値を与えることで得られる．

2.1 問題ドメインとアプリケーションドメイン

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とは異なる．Arango, Traczは，これらを**問題ドメイン**(problem domain)と**アプリケーションドメイン**(application domain)で区別した．解くべき問題は，いかにして解かれるべきかという情報を含まない．一方，アプリケーションは，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．

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，振舞い等を抽出したものから成る．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，振舞い，ソフトウェア部品等を抽出したものから成る．

2.2 ドメイン分析の定義

Neighborsは，**ドメイン分析**(domain analysis)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，再利用のプロセスまでを包含した概念として，ソフトウェアが開発される際に使用する情報を同定し，組織化することで，新システムを構築する際にそれらの情報を再利用するプロセスと定義した[Prieto-Diaz90,91]．

Arangoは，ドメイン分析を行える条件を次の通りに挙げる[Arango91]．

(1) ライブラリ等の再利用可能な部品はドメインごとに異なる，との認識が，分析を行うグループの中で共有されること．部品がドメインを超えて汎用的に再利用されることはなく，特定の問題ドメインやアプリケーションドメインの中で再利用される．

(2) 対象とする問題ドメインが流動的でなく，比較的安定していること．

2.3 ドメイン分析の効用

汎用モデルを採用するのではなく，アプリケーション分野ごとに固有のモデルを用いてドメイン分析を行うことが，従来の開発技術にはないドメイン分析の独特の主張である．

ドメインごとにそのモデルが異なることを前提とし，これによりドメインの特徴をより効率的に記述できる．また，汎用のソフトウェアライブラリを蓄積するのでなく，ドメインごとに再利用を図るため，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．さらに，DAM技術では要求定義から保守に至る様々なフェーズでドメインの観点から情報を再利用するため，従来に比べソフトウェアの生産性が向上する

実システム開発にDAM技術を適用した例[Tracz94]では，従来の要求分析の前に，ドメイン分析の工程を付加することによる，生産性や品質の変化を分析した．ドメインモデルやドメインに関するノウハウが少ない段階では，新たな工程が加わったことで生産性の低下を招くが，それらを確立し蓄積すると，ドメイン分析の工程だけでなく，続く下流の工程も所要時間が短縮する現象が観察された．また，従来の開発と比較して品質も向上した．このように，DAMを実際のソフトウェア開発に適用した成果例も海外で報告され始めた．

2.4 DAMのプロセス

DAMの入力から出力へ変換するプロセスの方法として，Arangoの方法[Arango91], Prieto-Diazの方法[Prieto-Diaz85,87,90],Cleavelandの方法[Cleaveland88], FODA[Kang90],IDeA[Lubars91],KAPTUR[Moore91]等が提案された．Wartikは各々についてその構成要素間の入出力関係を明確にして分類した[Wartik92]．

現状では，広範囲のドメインに適用できるDAMプロセスは，未確立である．ここでは，ドメインに共通に利用できる汎用的であるが，かなり簡単化した，ArangoのDAMプロセス[Arango91]を

紹介する．

このプロセスは，(1)ドメインの同定，(2)知識獲得，(3)ドメインモデルの表現，(4)進化，(5)評価・検証，から成る．すなわち，まず対象ドメインを明確にし，当該ドメインに関する様々な情報を収集する．それらの情報をモデルとして表現する(モデルの表現方法は2.5)．さらにいったんドメインモデルを構成した後も，ドメイン自体の変化に追随するためDAMプロセスを繰り返し，ドメインモデルを進化させていく必要がある．また，実システムへのフィードバックやDAMプロセスの繰返しによりドメインモデルの正当性や有効性を検証する．

DAMプロセスの入力として，Arangoは次の様々な情報ソースからの知識を挙げた[Arango91]．(1)技術文献：教科書,技術雑誌,マニュアル等，(2)既存のアプリケーション:ソースコード,設計資料,ユーザマニュアル,実現結果をリバースエンジニアリングで解析した結果等，(3)顧客調査:マーケット分析の結果を含む，(4)問題ドメインの専門家の知識:例えば会計士，化学者等の知識，(5)対象ドメインのシステム設計の専門家の知識:システムアナリスト,デザイナ,プログラマ等，(6)ドメインの進化の経緯の記録．

DAプロセスの出力として，例えば次が生成される．(1)問題とソフトウェアの仕様化に使われた概念の定義，(2)ソフトウェアの標準的な代替案やトレードオフ，(3)ソフトウェアの実装の計画．

Arangoによると，この出力モデルは問題ドメインとアプリケーションドメインの両ドメインモデルを含む．形式的な文法をもち，パーサで解析可能なドメイン固有な言語を生成する場合には，ソフトウェアの仕様記述からソースコードへ直接変換するプロセスをサポートする可能性がある．この汎用的な方法論は，実際のシステム開発での有効性や実現可能性の点からさらに検証する必要がある，と著者らは考える．

Fraserは，ドメイン分析はチームによる作業である場合が多いことから，グループウェアによるドメイン分析の手順や仕事分担を議論した[Frascr93]．

2.5 ドメインモデル

ドメインモデル(domain model)は，個々の問題ドメインやアプリケーションドメインの表現である．これは，個々のドメインの分析を行いモデリングを行うDAMプロセスの出力結果である．例えば，銀行オンラインシステムのドメインモデルや，計測制御システムのドメインモデルが存在する．

2.5.1 ドメインモデルの構成要素

ドメインモデルを記述するためには，ドメインの構成要素，構成要素間の関係，属性，振舞い等を記述する語彙や文法が必要になる．IWSRでは，ドメインモデルの構成要素は大きく次の3種類に分類可能であるという認識が一般的である．

(1) Language:ドメインの仕様を記述する言語．個々のドメインで，記述に用いる用語(Vocabulary)と文法(Syntax)が異なる．

(2) Library:要求仕様や設計仕様をプログラムコードに変換する時に用いる，ドメインに固有の標準的なライブラリ．

(3) Generator:Languageで記述された問題ドメインをLibraryに適合させてアプリケーションドメインに自動あるいは半自動で変換する，ドメイン固有のジェネレータやナビゲータ．

例えば，航空機座席予約ドメインでは，用語として，座席，フライト，日付，空港，乗務員，フライトスケジュール，座席予約，乗務員割当等がある．ドメインモデルは，このドメインで許される全ての妥当なアクションを生成できる文法と意味をもった言語で記述される[Prieto-Diaz87]．

2.5.2 ドメインモデルの記述手段の分類

ドメインモデルの記述する手段として，一般的にデータフロー図やER図が採用される傾向がある．ドメインモデルを記述する際に採用する記述手段は，対象とするドメインで異なる．Traczはドメインモデルが強調して記述する側面で，手段を次の通りに分類する[Tracz92]．

(1) Definitional Model(構成要素の抽出，列挙，分類を記述)…分類(taxonomy)，クラス分け(class scheme)，シソーラス(thesaurus)

(2) Structural Model(構成要素間の関係を明確にする)…概念グラフ(conceptual graph)，ER図(E/R diagrams)

(3) Functional Model(システムや個々の構成要

素の振舞いを記述)… SADT，状態遷移図 (state transition diagrams)，データフロー図 (data flow diagrams)

問題ドメインの記述手段の一例として，上記の分類には適合しないが，Bjorner は，リアルタイムシステムのドメインモデル記述手段として，VDM に基づく形式的仕様記述法を用いた[Bjorner92]．

筆者らは，ドメインモデルの記述手段，および問題ドメインとアプリケーションドメインのそれの使い分けについて，次の通りに考える．

アプリケーションドメインの記述手段は，問題ドメインとは異なり，実現方式を意識したものになるので，問題ドメインの記述手段より選択の範囲が狭くなる．すなわち，ソフトウェアの構造をある程度反映させた記述手段が必要になる．

2.5.3 Triadic Domain Model ： TDM

筆者らは，図-1 に示す通り，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(Domain Problem Model)，解群を表すためのドメインモデル(Domain Product Model)，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(Domain Process Model)から成る Triadic Domain Model(三つ組のドメインモデル:TDM と略す)を提唱している(文献[Itoh94])．

Domain Problem Model と Domain Product Model は，2.1 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 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，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．いずれを用いるかはドメイン固有である．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スも，ドメインごとに固有である．Domain Problem Model は，問題の記述のみから成り，解法の記述は含まない．例えば，銀行のオンライン処理のドメインは，顧客口座，支店，各種預金，金利等の構成要素から成る．

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

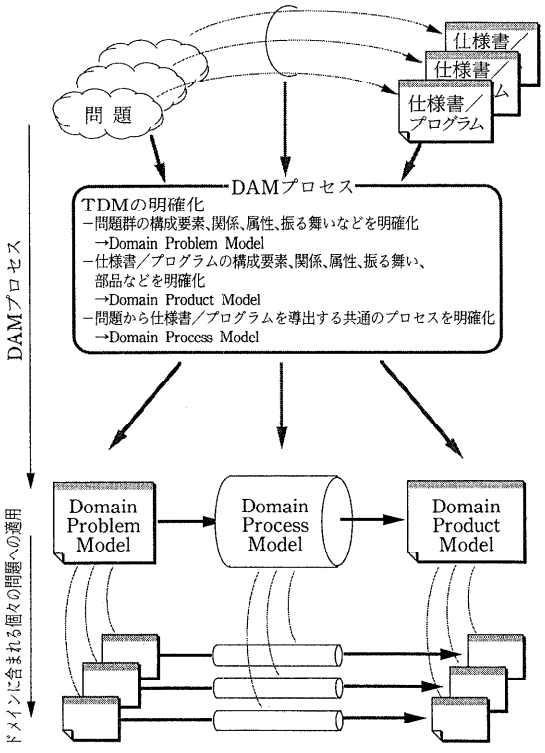


図-1 Triadic Domain Model の概念

ウェア群に共通な構成要素，関係，属性，振舞い，ソフトウェア部品等から成る，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．これは，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 には含まれないものも含む．例えば，オンライン処理のドメインでは，端末装置，ネットワーク装置，集中処理センタの CPU 等から成る．

Domain Process Model は，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．順序付けは，分析，仕様化，設計，実装での，手順のひな型になる．これに加えて，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．この Domain

Process Model は，他の DAM 研究では明確には認識されていない．

3. DAM 技術とソフトウェア工学

3.1 ソフトウェアの再利用やリバースエンジニアリングとの関係

DAM の手法は，従来のソフトウェアの再利用(reuse)を補完する技術として注目された．例えば，IWSR ではドメイン分析に関するグループの活動が活発である[Tracz92]．

ソフトウェアの再利用では，既存のソフトウェアから再利用可能なソフトウェア部品を抽出し，それを開発するソフトウェアに適用し，コーディングの工数を圧縮する手法が提案されてきたが，既存ソフトウェアの中で，再利用可能な情報の特定が困難である問題点があった．すなわち，既存ソフトウェアの全ての部分を部品化しても，その用途が汎用的であったり，逆に過度にドメインや固有の問題に依存すると，部品の再利用率は低下する．

このため，DAM では，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を行い部品の再利用率の向上を図る[Tracz92]．

Hutchinson は，開発済みのソフトウェアモジュールをドメインと照合し，再利用可能 / 変更して再利用可能 / 再利用不可能，に分類する試みを行った[Hutchinson88]．Simos は，部品の再利用を促進するためには単にライブラリを充実させるだけでなく，知識ベースを基にしたインテリジェントライブラリを構築し，ドメインに固有なライブラリ蓄積と再利用をすべきだと主張する[Simos91]．

一方，実装済みのソフトウェアから設計の仕様を抽出して，類似のソフトウェア開発時にそれらの情報を利用するリバースエンジニアリング(reverse engineering)も，ソフトウェア再利用の一種である．このリバースエンジニアリングも，DAM の手法によりさらに効果的なものになる．

リバースエンジニアリングで得られる情報は，設計時と異なり，変更，バグフィックス，バージョンアップ等で変更／追加された情報も含むため，以前の仕様を推定する問題がある．これに DAM の技術を応用し，対象のソフトウェアが属するドメインを与えて，それに属するソフトウェアの機能や構造を特定する．この情報と，リバースエンジニアリングで得られた情報を整合させ，実装時に変更・追加した情報を推定する．

この時，問題ドメインやアプリケーションドメインのドメインモデルを用いると，リバースエンジニアリングで得た情報とかけ離れる危険性がある．このため，Tracz は，ドメイン固有の実装情報から成る実装ドメインのドメインモデルを用いることがよいと述べた[Tracz92]．例えば，銀行オンラインシステムでは，ファイルの物理フォーマット等の実装時のみに現われる構成要素が実装ドメインを構成する．実装ドメインを導入した DAM の手法によりリバースエンジニアリングで得た情報を再利用効果の高いものにする．

IWSR でのドメイン分析グループの討論では，ドメイン分析が再利用しようとする情報をリバースエンジニアリングで得ようとする場合の注意点を次の通りにまとめた[Tracz92]．

(1) 最初のソフトウェアの構造が崩される可能性がある．パッチ，バグフィックス，バージョンアップは，この要因である．

(2) 最初のソフトウェアの構造や構成要素が明らかになっても，その構造を採用した設計の指標(design rationale)が明らかにならない．

(3) ソースコードや実行形式コードは実装に関わる条件が加味されるので，リバースエンジニアリングで得られた情報には，他のシステムへの移植の際に不必要な情報も含まれる．

DAM 技術は，ソフトウェアの再利用を効率化する手段として注目された経緯がある．このためドメインモデルは既存のソフトウェアをモデル化し，それを新システムの開発時に再利用する方法が一般的である．Neighbors は，これを Draco と呼ぶ再利用を主な目的とする開発環境上で実現した[Neighbors89,92]．Arango は，これを分析ーモデリングー再利用ループ(analysis-modeling-reuse loop)と呼ぶ[Arango92]．Arango はさらに，文献[Arango93]で，あるドメインの技術情報から成る設計ハンドブックをドメインモデルとみなし，この設計ハンドブックのインスタンスとして，特定のソフトウェア開発を行い設計の再利用を行う方式を示した．

一方，IWSR の討論では，ドメインモデルをソフトウェアの保守，教育，リスク分析等にも利用可能であるとする意見を紹介した[Tracz92]．

3.2 要求分析技術との関係

DAM技術はソフトウェアの再利用を補完するものとして生まれたが，ソフトウェア群を共通のドメインとして捉える観点で要求分析(requirements analysis)プロセスも効率化すると期待された．

McCainは，個々のソフトウェアシステムの要求定義に先立ち，ユーザが望むソフトウェアのドメインを市場分析する必要性を論じ，そのための枠組みを示した[McCain85]．またFickasは，確定した要求仕様をドメインモデルと照合することにより，仕様をより明確化できると主張する[Fickas88]．

GreenSpan[Greenspan82]は，各ドメイン固有のRMF (Requirement Models Framework)を提唱する．3種類の用語(オブジェクト Objects，活動 Activities，断定 Assertions)を定義し，それらに3種類の操作(分類 Classification，集合 Aggregation，一般化 Generalization)を施す．

ソフトウェアの仕様化と設計の国際ワークショップ(International Workshop on Software Specification and Design: IWSSDと略す)では，DAM技術を要求分析にも適用すべきとの立場から次の議論があった[IWSSD91]．

・アプリケーションの要求仕様をドメインごとに分類して再利用すべきである．

・ドメイン知識は，要求仕様の抽出と要求から設計への移行の手助けになる．

・研究者はドメイン知識を活用すべきとの認識をもつが，実際の開発プロセスでドメインモデルを抽出したり設計に適用する例はまだ少ない．

要求工学国際会議(International Workshop on Requirement Engineering:REと略す)でも，DAMを要求工学に適用すべきとの観点から議論があった．

Easterbrook[Easterbrook93]は，要求分析を行う担当者が使うべき概念モデルを表すために，要求工学でもドメインモデルを用いるべきであると述べた．しかし，既存の要求仕様の抽出技術は，単純な認知モデルに基づいている．例えば，要求分析に参加する人達の，社会的・組織的なビューを組み込めていない．このため，ビューの階層的な記述を導入し，相反したり矛盾するビューを一つのドメインモデルの中に組み込む．

Kramer[Kramer93]は，要求工学でのDAMの重要性を指摘し，ドメインを超えた過度の一般化を戒めた上で，解くべき課題として，(1)ドメインモデルを単なる抽象化として捉えるのではなく，システムを仕様化・開発した経験からドメインモデルを抽出しなければいけないこと，(2)システムの仕様化の経験をドメインエンジニアに正しく伝達する仕組みを作らなければならないこと，を挙げた． Barstow[Barstow93]は，ドメインモデルに基づいて個々のシステムを開発する利点について，(1)システムを構成する物理的あるいは計算的資源や現象の要素の概念，属性，関係をドメインモデルとして明らかにするので，個々のシステムの分析が容易になる，(2)システムのカスタム化や進化をドメインモデル上で把握しやすい，を挙げた．

Jackson[Jackson93]は，要求仕様はドメインを明確に記述すべきであり，個々のシステムに共通なドメインの性質と，個々のインスタンスの性質を明確に区別すべきであると，述べた．

Feather[Feather93]は，ドメインにまつわる要求を，ドメインモデルとしてERモデルで表現し，将来あるいは既存のシステムの要求との適合を手早く調べる手法を提案した．

3.3 ドメイン工学

Arangoは，ドメイン工学(domain engineering)を問題の明確化やソフトウェアの開発プロセスをドメインを意識して行う活動全般と定義した[Arango91]．これは本稿でDAMと呼んでいる活動に他ならない．ドメイン工学をもう少し限定して捉えると，問題ドメインとアプリケーションドメインの間にある概念のギャップを埋める技術である．

ドメイン工学で扱う領域は，(1)ドメイン分析，(2)インフラストラクチャの仕様化，(3)インフラストラクチャの実装，の3領域に分類される[Arango89]．このインフラストラクチャは，アプリケーションドメインとほぼ等しい概念である．すなわち，同一ドメインに属するソフトウェア群がもつ共通の構成要素，それらの関係，属性を指す．インフラストラクチャの仕様化はアプリケーションドメインの明確化，インフラストラクチャの実装は，アプリケーションドメインと実装ドメインの対応付けに相当する．ドメイン工学でのドメインモデルとは，ドメイン分析とインフラストラク

チャの仕様化の両プロセスの間で生成される中間生産物である．この見方も，2．で述べたドメインモデルの成立過程，すなわちドメイン分析によりドメインモデルが形成・明確化されるという主張と一致する．

3.4 ドメイン分析プロセスの進化と知識獲得

Prieto-Diazは，ドメイン分析プロセスを，知識を獲得しながら進化するプロセスであると捉える[Prieto-Diaz87]．解決すべき問題ドメインと，それを解決するアプリケーションドメインの両者が，少しずつであるが常に変化する．この変化があるため，以前のDAMプロセスで得られたドメインモデルは常に再利用可能な安定したものではなく，徐々に更新/進化させていく必要がある．このため，ソフトウェア開発が行われるごとにDAMプロセスを繰り返してドメインモデルを更新するか，あるいは以前得たドメインモデルがその後変化していないことを検証する．Prieto-Diazは，このドメインモデルの進化のプロセスを知識獲得と捉える．すなわち，DAMプロセス自体を知識獲得のプロセスであるとみなす．

Schoenは，知識獲得プロセスの最終製品である知識ベースに注目し，これを構築するための枠組みやツールを考察した[Schoen91]．ドメインの専門家等の知識を計算機の知識ベースに変換し，格納するためのエディタを開発した．これをドメインモデル構築の環境の一つと捉える．

Devanbuは，大規模なソフトウェアのドメインモデルを徐々に明らかにしていくプロセスがDAMであると捉えた[Devanbu90]．ドメインモデルはインクリメンタルな情報追加を吸収するため知識表現を採用し，また複雑さや隠れた仕様を明らかにするため推論プロセスを利用する．

3.5 ドメイン固有のソフトウェアアーキテクチャ

ソフトウェアアーキテクチャ(software architecture)[Garlan93]とは，ソフトウェアのもつ機能，構造，マンマシンインタフェース等の，比較的抽象的な構成要素，相互の関係，属性等を総称したものである．通常，ソフトウェアアーキテクチャには，ウォータフォールモデルでの上流のフェーズ(要求分析や概要設計)で決定される項目が含まれる．下流のフェーズ，すなわち詳細設計や実装の段階で初めて明らかになる項目は含まない．

Traczが提唱するDSSA (Domain-Specific Software Architecture)は，各ドメインに固有のソフトウェアアーキテクチャを指す[Tracz93]．これは，アプリケーションドメインがもつ構成要素にほぼ等しい．Traczによると，DSSAを得るためのドメイン分析のプロセスは次の手順をとる．

- ドメインのスコープの決定
- ドメイン固有の概念や要求を定義し明確化
- ドメイン固有の設計や実装に伴う制約条件を定義し明確化
- ドメインアーキテクチャのドメインモデルを構築
- 再利用可能なWork Productsを生成・再利用

3.6 ソフトウェア開発におけるセマンティクスギャップとの関わり

従来のソフトウェア開発で，要求仕様と設計仕様，あるいは設計仕様と実装仕様の間には，意味的な隔たり，すなわちセマンティクスギャップ(semantics gap)が存在し，このギャップを埋めることがソフトウェア開発の大きな負担である，という意見がある[Fischer92,93]．このセマンティクスギャップはまだ解決されていないが，次の通りに，ドメインモデルが用いる記述手段と併せて考えると整理できる，と著者らは考える．

同一ドメインに属する問題ドメインとアプリケーションドメインのドメインモデルは，同一の記述手段を用いるとは限らない．異なる記述手段を用いる時，問題ドメインからアプリケーションドメインへの移行，すなわち与えられた要求仕様からアプリケーションを開発する際に，ドメインを記述する視点が移動することを意味する．アプリケーションドメインと実装ドメインの間でも同様なことが起こり得るが，通常では両者の用いる記述手段は，共にアルゴリズムの実現手段を考慮したものになるので，その実装はスムーズである場合が多い．

二つのドメインモデルが異なる記述手段を用いた時には，ドメインの移行は単なるマッピングや詳細化にとどまらず，視点の移行を伴う．例えば，リアルタイムソフトウェアの性能要求を記述するためには，待ち行列モデル等の応答時間モデルをを用いるであろう．一方，アプリケーションドメインでは，処理のアルゴリズム等を記述するため，フローチャート等の記述手段を用いる．異なる視

点をもったドメイン間を移行する場合，これらの一見関係のない二つのドメインモデルの構成要素間に対応関係をつける必要がある．

両者の視点のギャップが大きいほど，この対応付けの負荷が大きくなる．あるドメインで，問題ドメインとアプリケーションドメインの両モデルのギャップが，ソフトウェアの設計フェーズの負荷である．ただし，視点のギャップの対応付けには未知の問題も多く，今後の研究を待つ必要がある．

3.7 ドメインモデルの複数記述手段による表現

著者らは，ドメインモデルは複数の側面を記述する手段が必要であると考える[伊藤93]．例えば，2.5で述べた記述手段には，構造等の静的な側面の記述に優れたもの，状態遷移等の動的な側面の記述に優れたものがある．上に述べたリアルタイムシステムに対しては待ち行列の挙動等のさらに動的な要素の記述が必要な場合がある．

Goguen[Goguen86]は，抽象言語 LIL(Library Interconnection Language)で様々なパラダイムの記述・変換をサポートする環境を構築した．LILでは，トップダウン，ボトムアップ，仕様からプログラムへの変換，データドリブン指向やデータフロー指向の記述をサポートする．Srinivasは，1つのドメインに複数の表現やビューが必要である場合があると主張し，この表現の1つとして代数記述が有効であると主張する[Srinivas90]．

オブジェクト指向のシステム開発技法では，要求分析から設計・実装に至るまで，一貫して対象システムを複数オブジェクトとそれらの相互作用としてモデル化する．DAMの観点からは，ソフトウェア開発の全工程に統一したモデルを適用することで，工程間のセマンティクスギャップを最小化した点で評価できる．例えばJohnsonは，他の対象システムに適用可能な，一般的なオブジェクトを作ることで再利用が促進される，と主張する[Johnson88]．また，Batoryは，オブジェクト指向で階層的にシステムをとらえる，GenVocaと呼ぶドメインモデリングの方式を提案した[Batory92]．

しかし，オブジェクト指向に限らず一つの手段のみで対象システムをモデル化しようとすると見方が一面的になり，ドメインモデルがもつ様々な側面を記述し損なう可能性もあることに留意したい．

4. 今後の課題

ソフトウェア工学で注目されているドメイン分析・モデリング(DAM)技術を概観した．DAM技術は，従来の開発作業をそのままCASEとして実現するために分析するものではない．複数のソフトウェアをドメインという枠組みでとらえ，得られたドメインモデルを基礎として，開発の方法論や手法を構成する．ドメインモデル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．

筆者らは対象ドメインごとに固有の用語，概念，要求の獲得方法，分析・仕様化の方法や手順等があると考える．このため，DAMの研究では今後次の研究を行う必要があると考える．

- 対象ドメインごとのDAM手順
- 対象ドメインごとのドメインモデル(記述項目，構造等)
- ドメインモデルに基づいた対象ドメイン向きの再利用部品群
- ドメインモデルに基づいた，一般的な開発手法の特定のドメイン向け特化方法
- ドメインモデルに基づいた個々のドメインに固有な開発手法，ツール
- 対象ドメインの進展に応じたドメインモデルおよび対応する開発手法の更新方法

DAM技術の研究室レベルでの研究は盛んである．この成果は，文献[Hess90]等にある．しかし，その実用面での妥当性や有効性は未だ未知数である．米国等ではドメイン分析向けのツールが徐々に実用化され，その有効性が数社のソフトウェアメーカで検証され始めた．実際のソフトウェア開発の現場でのドメインモデルの開発を通してその有効性，特長，欠点を検証する必要がある．

筆者らは，メーカ各社の方の自由な発表と討議を中心に，DAM技術に関する本会時限研究グループとして1995年3月末まで活動中である．

参 考 文 献

[Arango89]G.Arango:Domain Analysis - From Art Form to Engineering Discipline, 5th IWSSD, pp.152-159(1989).

[Arango91]G.Arango, R.Prieto-Diaz:Introduction and Overview:Domain Analysis Concepts and Research Directions, pp.9-26, Domain Analysis and Software Systems Modeling (Ruben Prieto-Diaz and Guillermo Arangoed.), IEEE(1991).

[Arango92]G.Arango,B.I.Blum:Applications of Domain Modeling to Software Construction EDITORIAL, Int.J.SE&KE,Vol.2,No.3,pp.321-323 (Sept. 1992).

[Arango93]G.Arango,E.Schoen,R.Pettengill:Design as Evolution and Reuse,IEEE 2nd IWSR,pp.9-18(1993).

[Barstow93]D.R.Barstow:Should We Specify Systems or Domain, IEEE RE'93,pp.79(Jan. 1993).

[Batory92]D.Batory,V.Singhal,M.Sirkin:Implementing a Domain Model for Data Structures,Int.J.SE &KE,Vol.2,No.3,pp.375-402(Sept. 1992).

[Bjorner92]Dines Bjorner:Trusted Computing Systems:The ProCos Experience, ICSE'92, pp.15-34 (May 1992).

[Cleaveland88]J.C.Cleaveland:Building Application Generators, IEEE Softw., Vol.5, No.4, pp.25-33 (July 1988).

[Devanbu90]P.Devanbu, R.J.Brachman, P.G. Selfridge, B.W.Ballard:LaSSIE:A Knowledge-Based Software Information System, ICSE'90(1990).

[Easterbrook93]S.Easterbrook:Domain Modeling with Hierarchies of Alternative Viewpoints, IEEE RE'93, pp.65-72 (Jan. 1993).

[Feather93]M.S.Feather:Requirements Reconnoitering at the Juncture of Domain and Instance, IEEE RE'93, pp.73-76 (Jan. 1993).

[Fischer92]G.Fischer,A.Girgensohn,K.Nakakoji, D.Redmiles:Supporting software designers with integrated domain-oriented design environments, IEEE Trans.SE,Vol.18,No.6,pp.511-522(June 1992).

[Fischer93]G.Fischer:Integrating Construction and Argumentation in Domain-oriented Design Environments,IEEE RE'93,pp.284(Jan. 1993).

[Fickas88]S.Fickas, P.Nagarajan:Critiquing Software Specifications, IEEE Software, Vol.5, No.6, pp.37-47(Nov. 1988).

[Fraser93]S.D.Fraser, C.S.Saunders:Enhanced Reuse with Group Decision Support Systems,IEEE 2nd IWSR,pp.168-175 (1993).

[Garlan93]D.Garlan, Mary Shaw:Architectures for Software Systems, Tutorial on ACM SigSoft'93 (Dec. 1993).

[Goguen86]J.A.Goguen:Reusing and Interconnecting Software Components, IEEE Computer, Vol.19, No.2, pp.16-28(Feb. 1986).

[Greenspan82]S.J.Greenspan, J.Mylopoulos, A. Borgida:Capturing More World Knowledge in the Requirements Specification, 6th ICSE, pp.225-234 (1982).

[Hess90]J.A.Hess, W.E.Novak, P.C.Carroll, S.G.Cohen, R.R.Holibaugh, K.C.Kang, A. S.Peterson:A Domain Analysis Bibliography, Technical Report CMU-SCI-90-SR-3, Softw. Engine. Institute, Carnegie-Mellon University (1990).

[Hutchinson88]J.W.Hutchinson,P.G.Hindley:A Preliminary Study of Large-Scale Software Reuse,Softw.Engi.J.,Vol.3,No.5,pp.208-212(1988).

[伊藤 93]伊藤潔, 杵嶋修三：リアルタイムシステムにおけるネット指向開発技術の適用, 情報処理, Vol.34, No.6, pp.747-760 (June 1993).

[Itoh94]K.Itoh, S.Kishima, Y.Tamura:Systems Integration on Specification, Design and Generation of Reactive System - Triadic Domain Model-Based Approach-,3rd IEEE ICSI(Aug.1994).

[IWSSD91]IWSSD'91(Int.Workshop on Softw.Specification and Design) ,A.Volta, Como(Oct.1991).

[Jackson93]M.Jackson, P.Zave:Domain Description, IEEE RE'93, pp.56-64 (Jan.1993).

[Johnson88]R.E.Johnson, B.Foote:Designing Reusable Classes, J. of Object-Oriented Programming, Vol.1,No.2, pp.22-35,(June /July 1988).

[Kang90]K.Kang,S.Cohen,J.Hess,W.Novak,A.S.Peterson: Feature-Oriented Domain Analysis (FODA) Feasibility Study,Carnegie Mellon Univ.-Softw. Engi. Institute, CMU/SEI-90-TR-21(Nov. 1990).

[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:Asdreas STD Triad, 情報処理学会論文誌, Vol.34, No.9, pp.2025-2036 (Sept.1993).

[Kramer93]J.Kramer:Generalizations are False, IEEE RE'93, pp.80 (Jan. 1993).

[Lubars91]M.D.Lubars:Domain Analysis and Domain Engineering in IDeA, Domain Analysis and Software Systems Modeling (Ruben Prieto-Diaz and Guillermo Arango ed.), IEEE (1991).

[McCain85]R.McCain:Reusable Software Component Construction:A Product-Oriented Paradigm, 5th AIAA/ACM/NASA/IEEE Computer in Aerospace Conference, pp.125-135 (1985).

[Moore91]J.M.Moore,S.C.Bailin:Domain Analysis:Framework for Reuse,Domain Analysis and Software Systems Modeling(Ruben Prieto-Diaz and Guillermo Arango ed.),pp.163-178,IEEE(1991).

[Neighbors81]Neighbors, Ph.D. Thesis, Univ. Calif. Irvine (1981).

[Neighbors89]J.M.Neighbors:DRACO:A Method for Engineering Reusable Software Systems,Softw. Reusability,Vol.1,ACM,pp.295-320(1989).

[Neighbors92]J.M.Neighbours:The Evolution from Software Components to Domain Analysis, Int. J.SE &KE,Vol.2,No.3,pp.325-354(Sept. 1992).

[Prieto-Diaz85]R.Prieto-Diaz:A Software Classification Scheme,PhD thesis,Univ.California (1985).

[Prieto-Diaz87]R.Prieto-Diaz:Domain Analysis for Reusability, COMPSAC'87, pp.23-29 (1987).

[Prieto-Diaz90]R.Prieto-Diaz:Domain Analysis:An Introduction, ACM Sigsoft Softw. Engi. Notes, Vol.15, No.2, pp.47-54(1990).

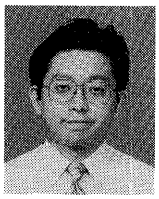
[Prieto-Diaz91]R.Prieto-Diaz:A Domain Analysis Methodology, Workshop on Domain Modeling, 13th ICSE, pp.138-140 (May 1991).

[Schoen91]E.Schoen:Active assistance for domain modeling, 6th IEEE Annual Knowledge-Based Softw. Engi. Conf., pp.26-35,IEEE(1991).

[Simos91]M.A.Simos:The Growing of an Organon:A Hybrid Knowledge-Based Technology and Methodolo-

gy for Software Reuse,Domain Analysis and Software Systems Modeling(Ruben Prieto-Diaz and Guillermo Arango ed.),pp.204-223,IEEE(1991).
[Srinivas90]Y.V.Srinivas:Algebraic Specification for Domains, Technical Report ASE-RTP-102, Dep. Information and Computer Science, University of California, Irvine (1990).
[Tracz92]W.Tracz:Domain Analysis Working Group Report-First Int.Workshop on Software Reusability, ACM SIGSOFT Softw. Engi. Notes, Vol.17, No.3, pp.27-34 (1992).
[Tracz93]W.Tracz, L.Coglianese and P.Young:A Domain-Specific Software Architecture Engineering Process Outline, ACM SIGSOFT Softw. Engi. Notes, Vol.18, No.2, pp.40-49(1993).
[Tracz94]W.Tracz:Domain Analysis Technology for Software Reuse, Software Tool Symposium, Tokyo (Jan.1994).
[Wartik92]S.Wartik,R.Prieto-Diaz:Criteria for Comparing Reuse-Oriented Domain Analysis Approaches, Int.J.SE&KE,Vol.2,No.3(Sept.1992).

（平成6年3月7日受付）

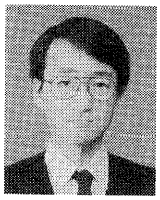


田村　恭久（正会員）

1961年生．1985年上智大学理工学部機械工学科卒業．1987年同大学院理工学研究科博士前期課程修了．同年，(株)日立製作所入社，同システム開発研究所勤務．大型計算機マルチプロセスアーキテクチャの研究開発に従事．1993年上智大学理工学部助手．アーキテクチャ設計支援，システム性能評価・改善手法，プロトタイピング手法，ドメイン分析・モデリング手法等に関する研究に従事．人工知能学会，IEEE，ACM，Society for Computer Simulation等各会員．

伊藤　潔（正会員）

1951年生．1974年京都大学工学部情報工学科卒業．1979年同大学院情報工学専攻博士課程修了．京都大学工学博士．1979年より上智大学に勤務．1985年より助教授．現在，同理工学部機械工学科情報システム講座所属．ソフトウェア工学，プロトタイピング手法，シミュレーション手法，ドメイン分析・モデリング手法，定性推論の待ち行列ネットワークへの応用，三面図からのソリッドモデルの自動合成法の研究に従事．本会ドメイン分析・モデリング研究グループ主査．本会論文誌編集委員会ソフトウェアグループ主査，元本会学会誌編集委員．著書「ソフトウェア開発のためのプロトタイピングツール」（共著），「システムプログラム」，訳書「並行処理とUnix」（共訳）．人工知能学会，IEEE，ACM等各会員．



杵嶋　修三（正会員）

1948年生．1968年佐世保高等専門学校機械工学科卒業．同年山武ハネウエル(株)入社．現在アドバンステクノロジーセンター主任研究員．研究テーマ：システムインテグレーション，アーキテクチャ，オペレーティングシステム，言語処理，記号処理，システム分析／モデリング，CASE，リアクティブシステム，ソフトウェア工学．ACM会員．本会ドメイン分析／モデリング研究グループ幹事．