

## ソフトウェア再利用のための ドメイン分析手法

4J-5

安藤秀樹 紫関昭光 岡田全裕 浅井昌樹

日本アイビーエム株式会社

### 1. はじめに

『ドメイン分析』は、システムより上位に位置づけられる「領域」に対する分析を行い、その領域の情報を整理・分類(機能やオブジェクトの選別・抽象化、関係の定義・抽象化)し、共通の性質を識別し、再利用可能な成果物を作成することを目的とする。ドメイン分析の過程で生成されるモデルを『ドメイン・モデル』と呼び、再利用可能部品を組み立てるアーキテクチャをDSSA(Domain Specific Software Architecture)という。このDSSAを生成する手法が、Domain-Engineering Processである。本論文では、この手法を紹介し、プロセスを実践するための指針を提供することを目的とする。

### 2. DSSA Domain-Engineering Processの概要

この手法は、5つのステージより構成される。

#### (1) ドメインの定義 [User Needs]

まず、ドメインの内側と外側にあるオブジェクトを切り出し、その境界を見極めることで分析の対象ドメインを定義し、その範囲を明確にする。

あまりモデル化を意識せずに、functions/objects/thingsの洗い出しのみを行い、オブジェクト指向分析の手法に近いやり方でハイレベルのブロック・ダイアグラムを作成する。

また、共通性を見つけたすために、対象となるドメインと関連を持つサブ・ドメインや類似のドメインも明確にしなくてはならない。

#### (2) コンセプト/実体/要件の定義 [Problem Space]

ステージ1のアウトプットをもとに、データ・フ

ロー/制御フロー・ダイアグラムを作成し、対象ドメイン内のコンセプト(機能面から見た処理の単位)の働き・関係を明確にする。そしてそれらを、異なるアプリケーション間で「必須」とされるもの、「オプション」的なもの、また、トレードオフを考え「代替可能」なものへの振り分けを行い、共通性を見つけたす。

#### (3) デザインと制約の定義 [Solution Space]

ステージ2のアウトプットをもとに、ドメインを実現するためのデザインと制約を定義する。その制約と、コンセプトとの関係を明確にする。

異なったアプリケーション間で変化しえないもの(Stable)と変化しうるもの(Variable)を区別する。このStableなものが再利用可能部品の候補となる。

このステージで、対象ドメインに対する分析が終わり、次のステージからは、再利用性を重視した開発支援環境の構築となる。

#### (4) アーキテクチャー/詳細モデルの開発

ここではアーキテクチャー(DSSA)を定義し、それぞれのモジュール間のインターフェースを明確にし再利用可能なモジュールの仕様を決定する。

#### (5) 再利用部品の生成

DSSAに基づいて、再利用可能なモジュール部品を開発生成し、ドメイン・モデルなどとの対応付けを明らかにする。

### 3. Domain-Engineering Process実践のための指針

ドメイン分析は、Domain Engineer(分析の過程をリードできる人)とDomain Expert(対象ドメインを熟知している人)の2者によって、実施されるべきである。Domain Engineerは、ステージ1でドメインを定義し、ドメイン分析のゴール・イメージを確立させ、Domain Expert達との間でコンセンサスを得る必要がある。特にステージ1は、非常に抽象的な対象物を扱うため、その記述は不明確になりやすく、頭の中で各自がわかったつもりになってもその実体は異質

Domain Analysis Method for Software Reuse

Hideki Andoh, Akimitsu Shiseki, Masahiro Okada  
and Masaki Asai

Yamato Lab., IBM-Japan, 1623-14 Shimotsuruma,  
Yamato, Kanagawa 242, Japan

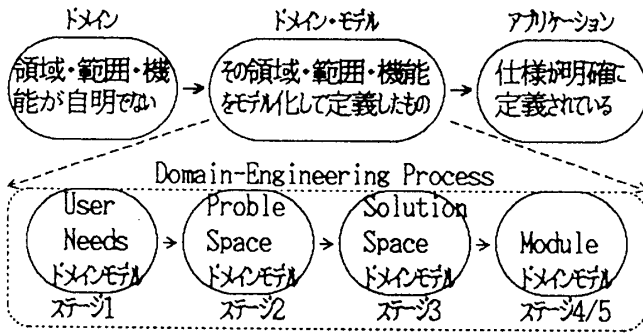


図1. ドメイン分析のフロー

なものになりやすい。図1のようなプロセスのフローを共通の認識として位置づけなければならない。

分析の対象となるドメインとは、「実在するが、その領域・範囲・機能等が、自明でない」ものであり、ステージ1～5のモデル化(実在しないが定義は明確)を経て、「現実に存在し仕様が明確」なアプリケーションの開発に用いられるアーキテクチャ(DSSA)と再利用可能なモジュール群を生成する。

Domain-Engineering Processの手法が他のドメイン分析手法と異なる点は、functional requirements と implementation constraintsを区別し、Problem SpaceとSolution Spaceの分析の位置づけを明確にしているところにある。この区別は困難を伴うが、解決しなければならない問題は正しく認識されうる。この関係を図2に示す。

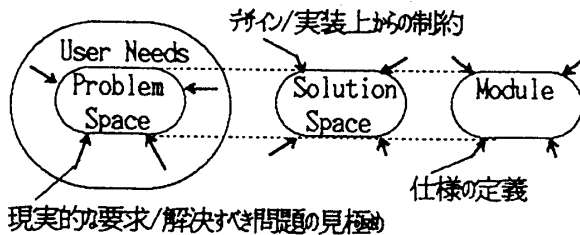


図2. 分析の対象と範囲

4. ステージ1におけるドメイン定義の考慮点

ドメインとは、「objectを要素とする集合体」を意味し、ドメインを定義するとは、「そのドメインに含まれるobjectを決定し、そのobjectのどの属性に着目するかを決定する」ことである。

ドメインから functions/objects/thingsを洗いだす時、明確に実体のあるobjectとして定義しにくい抽

象的概念をthingsとする。ステージ1で作成するハイレベルのブロック・ダイアグラムは、主要なオブジェクトの記述に限定し、ステージ2以降で詳細(モデル)化していく。[例:図3]

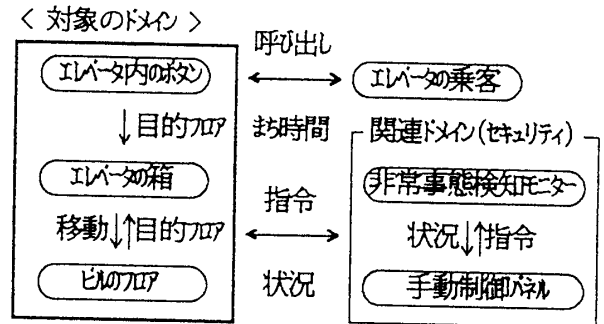


図3. ブロック・ダイアグラムの例

またsimilar/related domainsを見出たす時には、含まれているobjects間でmessageのやりとりがあるか、共通のobjectを共有しているかに着眼する。

ここでドメインの範囲が明確になっていないと、次の段階で後戻りの作業が発生するため、この段階で正しく漏れなくユーザーニーズを分析し、オブジェクトを洗いだしておくなければならない。

5. 終わりに

ひとたびステージ1でドメインが定義されると、ステージ2以降のモデルの詳細化は主に外部要因によって機械的に進められる。それ故に、Domain Expertsの主観・センスに負うところの大きいUser Needsに基づくドメイン定義は、最重要項目である。従って本論文では、ステージ1に焦点を当てた。

いかにして適度に狭いドメインの境界を設定し、類似のアプリケーションを数多く開発できるようにするかが、基本的課題として研究されている。

ドメイン定義のための理論的研究、表記法の開発、Domain Engineerの育成、試行錯誤による経験の蓄積等が今後さらに必要である。

参考文献

A TUTORIAL ON DOMAIN-SPECIFIC SOFTWARE ARCHITECTURES AND MEGAPROGRAMMING, Will Tracz IBM-FSD, Tutorial of COMPSAC'92