

データ中心アプローチとユースケースに基づく オブジェクト指向フレームワーク構築手法

名 取 万 里[†] 加 賀 谷 聡[†] 本 位 田 真 一[†]

本稿では、オブジェクト指向フレームワークを構築するための手法を提案する。本手法は、現行のアプリケーションシステム群の分析とドメインの静的な特徴を抽出するためにデータ中心アプローチを取り入れる。また、ドメインのあるべき姿の分析と動的な特徴を抽出し、静的な特徴と合わせて一般化するために、ユースケースとそれに基づくオブジェクト間のインタラクションに着目する。さらに、これらの2つの観点を統合し、オブジェクト指向フレームワークを構築するプロセスを明らかにする。

Developing Object-oriented Frameworks Based on Data-oriented Approach and Use Cases

MARI NATORI,[†] AKIRA KAGAYA[†] and SHINICHI HONIDEN[†]

This paper presents a method for developing Object-oriented frameworks. Our framework development method includes three major ideas. The first is to extract static features of a domain by Data-oriented Approach. The second is to extract dynamic ones based on use cases and interactions among objects. The last is to integrate the above two ideas in order to cross-check features of the domain and to construct Object-oriented frameworks.

1. はじめに

フレームワーク中心のソフトウェア開発が注目されており^{1)~3)}、オブジェクト指向フレームワーク(以下、フレームワークと略す)を構築することの必要性が高まっている。フレームワークの構築には、ドメインのフローズスポットとホットスポット²⁾を特定し、これらをフレームワークの構成要素として表現するための形態を明らかにすることが必要である。フローズスポットとは、あるドメインのアプリケーションシステム群の共通部分である。ホットスポットとは、アプリケーションシステムごとに異なる可変部分である。フローズスポットとホットスポットの特定には、ドメイン分析が必要である²⁾。本稿では、ドメイン分析によりフローズスポットとホットスポットを抽出し、これらをフレームワークパターンとして形式化することにより、フレームワークを構築する手法について論じる。

ところで、ドメイン分析では、既存のアプリケーションシステム群を分析し、それに加えて、分析結果を新規のアプリケーションシステムへ適合しやすくするために、アプリケーションシステムのあるべき姿の分析を行う。すなわち、既存アプリケーションシステム群の特徴を明らかにし、その結果を一般化して、新規アプリケーションシステムの開発において再利用可能なモデルを構築する。また、ドメイン分析では、分析結果の信頼性、適用可能性、柔軟性などを向上させるために、ドメインを様々な側面から分析し、クロスチェックを行うことが必要である。

ソフトウェア再利用技術の1つとして、様々なドメイン分析手法が提案されている^{4)~7)}。我々は、アプリケーションシステムの持つ業務機能と、各機能における画面遷移に着目したドメイン分析手法を提案している^{8),9)}。しかし、これらのドメイン分析手法^{4)~9)}においては、既存のアプリケーションシステム群とシステムのあるべき姿を多角的な側面から分析するための観点および、それらの観点に基づく分析手順などは確立されていない。

既存のアプリケーションシステムについて明確な分析手順を提供し、最適で標準化されたデータモ

[†] 株式会社東芝研究開発センターシステム・ソフトウェア生産技術研究所
Systems and Software Engineering Laboratory, Research and Development Center, Toshiba Corporation

デルを構築する手法として、データ中心アプローチ (Domain-Oriented Approach, 以下, DOA と呼ぶ) がある^{10)~12)}。DOA とは、システムの分析や設計にあたり、先にデータの存在を認め、その分析を徹底した後に、システムの構造を決定するアプローチである。DOA は、既存のアプリケーションシステムで使用される画面・帳票類からデータ項目を抽出し、データ項目の標準化および正規化を行うという作業手順が明確である。DOA によると、アプリケーションシステムのデータモデルが標準化・正規化された実体関連図 (Entity Relationship Diagram, 以下, ERD と呼ぶ) の形式で表現される。各アプリケーションの ERD は、標準化・正規化により記述レベルが統一されるので、アプリケーション間で共通あるいは、アプリケーションごとに異なるエンティティとそれらの関連に着目することにより、フローズスポットとホットスポットを抽出しやすい。そこで、我々は、フレームワークの構築プロセスに DOA を組み込むことを提案する¹³⁾。

DOA は、既存のアプリケーションシステムを対象として、ドメインの静的な特徴であるデータの側面から分析を行う。上で述べたように、ドメイン分析では、アプリケーションシステムのあるべき姿の分析および、データの側面だけではなく、多角的な側面からの分析が必要である。アプリケーションシステムのあるべき姿を明らかにするためには、業務知識を持った顧客との相互作業が不可欠である。そこで、顧客にとっても分析結果が把握しやすく、検証しやすい手段として、ドメインのユースケース¹⁴⁾に着目する。ユースケースに基づいて、DOA で明らかにした ERD のエンティティ間のインタラクション、エンティティの状態遷移、インタラクションで出現するイベント/アクションのデータフローを記述し、DOA による分析結果と合わせて相互に検証することにより、多角的な側面からドメイン分析を行うことができると考えられる。

そこで、本稿では、DOA とユースケースの観点からドメイン分析を行い、これらの分析結果を統合し、ドメインのフローズスポットとホットスポットを特定し、フレームワークを構築する手法を提案する。また、フローズスポットとホットスポットを、フレームワークの断片であるフレームワークパターンとして表現することを提案する。

以下では、2章において、本研究におけるフレームワークおよびフレームワークパターンの定義を示す。3章では、フレームワークの構築手法を提案する。本手法は、既存のアプリケーションシステム群の分析をデータの観点から行うために DOA を取り入れる。ま

た、本手法では、システムのあるべき姿を明らかにし、多角的な側面から分析を行うために、ユースケースに着目し、それに基づいて、データ間のインタラクション、状態遷移、イベント・アクションのデータフローを記述する。さらに、本手法では、これらの分析結果から、ドメインのフローズスポットとホットスポットを特定し、これらをフレームワークパターンとして表現する。4章では、本フレームワーク構築手法の適用プロセスで作成された成果物およびフレームワークの事例を示す。5章では、本手法を実際のフレームワーク構築に適用した結果から、本手法の有効性を評価する。6章では、本研究と関連する研究を取り上げて、比較する。7章では、本研究のまとめを示し、今後の課題を整理する。

2. フレームワーク

我々の提案するフレームワークは、以下の2つの要素から構成される。

● フレームワークパターン

- フローズスポットとホットスポットの接続部分
- フレームワークパターンとは、フレームワークの断片である。フレームワークとは、抽象クラスと具象クラスの集合であり、それらの相互のインタフェースであり、それらの集合が特定のドメインに汎用的なソフトウェアシステムを包括しているものである²⁾。したがって、フレームワークパターンは、抽象クラスと具象クラスの集合であり、それらの相互のインタフェースである。

フレームワークは、フローズスポットとホットスポットから構成される²⁾。ホットスポットには、アプリケーションごとに組込み型の選択肢が存在するものと、アプリケーションの違いによって、カスタマイズが必要となるものがあると考えられる。そこで、我々は、ホットスポットの特徴を反映させて、フレームワークパターンを次のように、基本パターン、組込みパターン、付加パターン、事例パターンに分類する。図1にフレームワークパターンの特徴を表す。

基本パターン: あるドメインのフレームワークのフローズスポットに対応するフレームワークパターンである。1つのフレームワークに対して、1つの基本パターンが対応する。

組込み (build-in) パターン: あるドメインのフレームワークのホットスポットに対応する。また、クラス/クラスの属性/メソッドの追加・修正をすることなく、フローズスポットにそのまま組み込むことが可能なフレームワークパターンである。

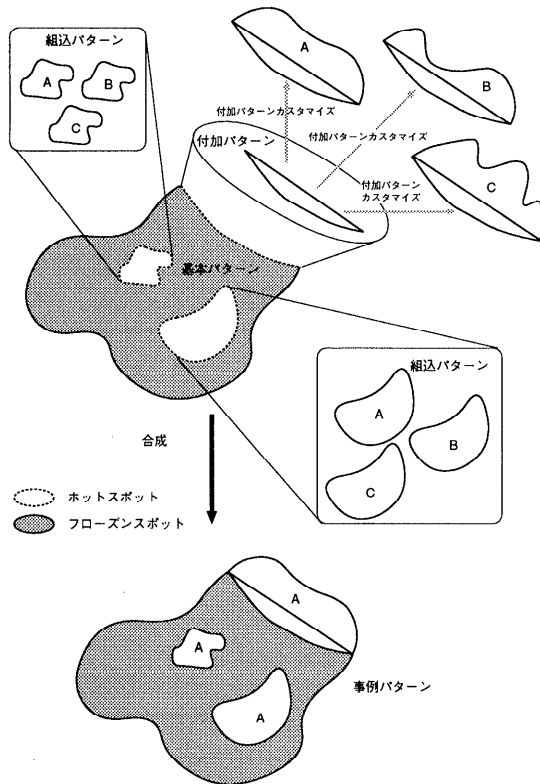


図1 フレームワークパターン
Fig.1 Framework patterns.

フレームワークの1つのホットスポットに対して、選択可能な複数の組込みパターンが存在する。

付加 (build-on) パターン： あるドメインのフレームワークのホットスポットに対応する。また、アプリケーションごとにクラス/クラスの属性/メソッドの追加・修正を行い、フローズンスポットに付加することにより実装するフレームワークパターンである。フレームワークの1つのホットスポットに対して、選択可能な複数の付加パターンが存在する。

事例パターン： 基本パターンが持つフローズンスポットとホットスポットの接続部分に、組込みパターンまたは、カスタマイズした付加パターンを合成することにより作成されるフレームワークパターンである。アプリケーションシステムは、複数の事例パターンの組合せにより作成される。事例パターンは、他のドメインの組込みパターンとしても活用される。

フローズンスポットとホットスポットの接続部分とは、フレームワークにおいて、アプリケーションごとに異なる部分が発生する分岐点となるクラスである。

これらのクラスとしては、たとえば、アプリケーションごとに異なるサブクラスが考えられる抽象クラス、部分クラスの数と種類がアプリケーションごとに異なる、それらの集約クラスなどが考えられる。

3. フレームワーク構築手法

本章では、フレームワークの構築手法を提案する。本手法は、既存のアプリケーションシステム群の分析をデータの観点から行うために、DOAを取り入れる。また、本手法では、システムのあるべき姿を明らかにし、多角的な側面から分析を行うために、ユースケースに着目し、それに基づいて、データ間のインタラクション、状態遷移、イベント・アクションのデータフローを記述する。さらに、本手法では、これらの分析結果から、ドメインのフローズンスポットとホットスポットを特定し、これらからフレームワークパターンを抽出する。

以下では、DOAとユースケースに着目する理由を整理する。また、我々の提案するフレームワークの構築手法の詳細を示す。さらに、フレームワーク構築プロセスとフレームワークとの関係を明らかにする。

3.1 DOAとユースケースに着目する理由

1章において述べたように、フレームワークの構築には、ドメインのフローズンスポットとホットスポットを特定することが必要であり、それにはドメイン分析が必要である。しかし、現状のドメイン分析手法には、次の課題を解決する必要があると考えられる。(1) 既存のアプリケーションシステム群とシステムのあるべき姿を、多角的な側面から分析するための観点を明らかにすること。(2) それらの観点に基づく分析手順を確立すること。

我々は、上記の課題を解決するために、DOAとユースケースに着目する。その理由を以下に示す。

● 課題 (1) に対して：

DOAとユースケースに着目すると、ドメインで使用されるデータと業務プロセスの分析が行える。DOAは、既存のアプリケーションシステムで使用される画面・帳票類からデータ項目を抽出し、データ項目の標準化および正規化を行い、ドメインの静的な特徴であるデータの側面から分析を行う。

一方、顧客とともにユースケースの抽出作業を行うことによって、企業の業務活動のプロセスを明らかにすることができる。ユースケースの抽出作業では、現行業務のプロセスをワークフローなどで記述して、現行業務のプロセスを明らかにし、

その過程で、業務の無駄や重複を排除し、また、新規に必要なプロセスを洗い出す。よって、ユースケースに着目すると、現行システムとシステムのあるべき姿の側面における業務プロセスの分析が行える。

ユースケースに基づいて、DOA で明らかにしたエンティティ間のインタラクション、エンティティの状態遷移、インタラクションで出現するイベント/アクションのデータフローを記述し、ERD も含めて相互に検証することにより、データ項目、データ間の関連、データの役割を明確にしやすい。また、データやプロセスの抜けや冗長部分を排除し、多角的な側面からのクロスチェックを含むドメイン分析を行うことができると考えられる。

● 課題 (2) に対して：

DOA に基づく現行アプリケーションシステムの分析は、ドメイン分析の出発点と作業手順が明確である。すなわち、画面・帳票類からのデータ項目の抽出という出発点と、データ項目の標準化、正規化を行うという作業手順が明確である。また、分析の範囲は、現行の画面帳票類で使用されるデータ類である。DOA によるデータモデルでは、各アプリケーションシステムのデータモデル構築において、データの名称、表現形式、データ項目の内容記述などが、アプリケーションシステム間において統一される。データの正規化により、データの重複の排除と項目の組合せの最適化が行われる。アプリケーション間において、正規化された同一名称のエンティティとそれらエンティティ間の関係に着目すると、フローズンスポットの候補を抽出しやすい。また、アプリケーションごとに異なるエンティティとその関連に着目すると、ホットスポットの候補を抽出しやすい。

ドメインのユースケースを抽出することは、システムのあるべき姿の分析の出発点になると考えられる。ユースケースに基づいて、DOA で明らかにしたエンティティ間のインタラクション、エンティティの状態遷移、インタラクションで出現するイベント/アクションのデータフローを記述することは、ホットスポットとフローズンスポットをドメインの動的な側面から特定するための主要なポイントになると考えられる。DOA の手順とこれらのポイントを主軸すると、ドメイン分析の作業手順を組み立てやすい。

以上の理由をふまえて、我々は、DOA とユースケースに基づき、ドメイン分析を行い、ドメインのフロー

ズンスポットとホットスポットを特定し、これらをフレームワークパターンとして抽出し、フレームワークを構築するプロセスを提案する。プロセスの詳細を、次節に示す。

3.2 フレームワーク構築プロセス

ここでは、フレームワークを構築するためのプロセスを示す。図 2 は、本フレームワーク構築手法とフレームワークの関係を示している。本フレームワーク構築手法は、次の 2 つのプロセスに分割する。

DA1： ドメイン分析を行い、フローズンスポットとホットスポットを特定するプロセス。

DA2： DA1 で明らかにしたフローズンスポットとホットスポットをフレームワークパターンとして形式化するプロセス。

DA1 は、次の 2 つのプロセスから構成される。

P1) DOA とユースケースの観点で、対象となるドメインの情報を収集し、これらを整理して、ドメインのスコープを決定するプロセス。

P2) DOA とユースケースの観点により、相互にクロスチェックを行い、オブジェクト図におけるフローズンスポットとホットスポットおよび、これらの接続部分を抽出するプロセス。

DA2 は、次の 2 つのプロセスから構成される。

P3) オブジェクト図におけるフローズンスポット、ホットスポットから、基本/組込み/付加パターンを抽出するプロセス。

P4) フローズンスポット、ホットスポットの接続部分に基づいて、基本/組込み/付加パターンを合成し、事例パターンを抽出するプロセス。

以下に、各プロセスの詳細を示す。

3.3 プロセス P1：スコープの決定

本プロセスでは、DOA によって現行アプリケーションシステム群のデータモデリングを行う。また、ドメインのユースケースを抽出する。ユースケースとアプリケーション群のデータモデルのクロスチェックを行い、ドメインのデータモデルを明らかにする。本プロセスの手順を以下に示す。また、本プロセスにおけるデータフローを付録の図 14 に示す。

P1-1) DOA によるデータモデリング：

DOA のプロセスに従って、現行の各アプリケーションシステムのデータモデルを構築し、ERD により記述する。また、DOA により、アプリケーションシステム群のデータモデルを統合し、これらを正規化し、ドメインのデータモデルを ERD で記述する。DOA によるドメインの ERD 作成プロセスでは、エンティティ、データ項目、リレー

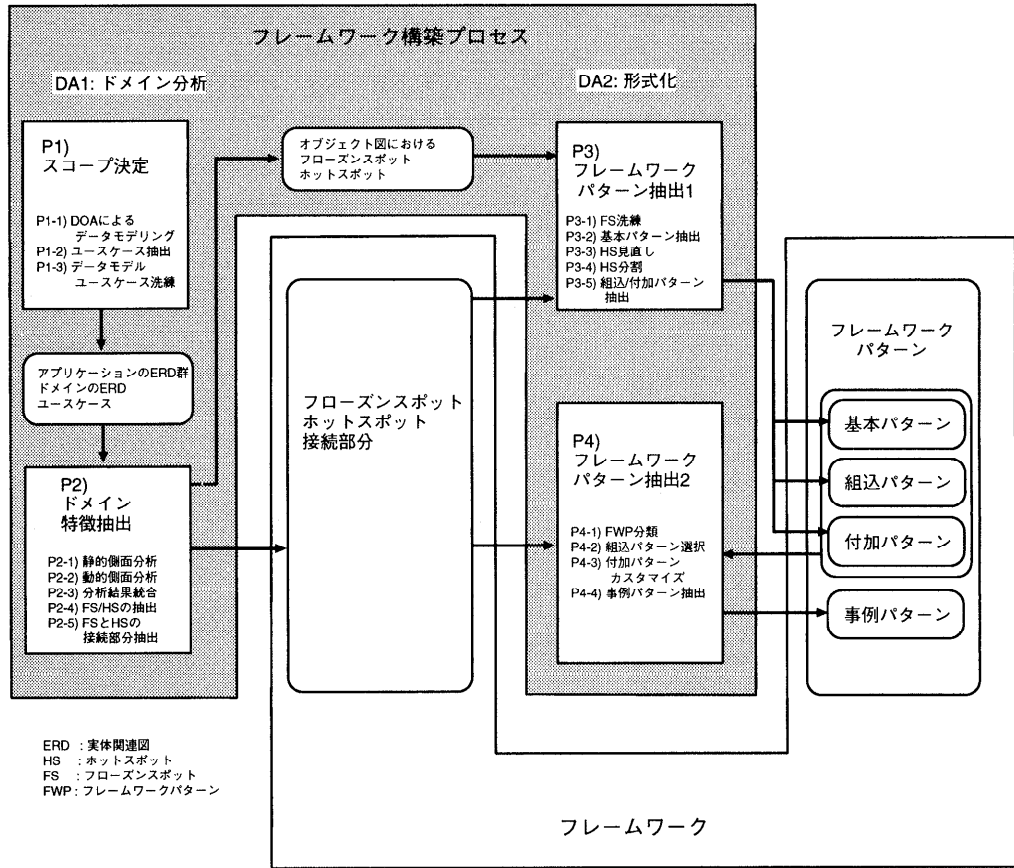


図2 フレームワーク構築プロセスとフレームワークの関係

Fig. 2 Relationships between our framework development processes and frameworks.

ションシップ名などが標準化される。アプリケーションのERD群に対して、名称の標準化の結果を反映させておく。本プロセスにおけるDOAのデータモデリングプロセスの詳細を、付録の図15を用いて、付録に示す。

P1-2) ユースケース抽出： 顧客へのインタビューを行い、ドメインのユースケースを抽出し、そのドメインのアプリケーションシステムの動的側面におけるあるべき姿を明らかにする。以下に手順を示す。

P1-2-1) アクター抽出： 顧客へのインタビューを行い、アプリケーションシステム群のアクターを抽出する。

P1-2-2) ワークフロー記述： アクターによる業務のワークフローを明確にする。

P1-2-3) ユースケース抽出： ワークフローから、シナリオを洗い出す。シナリオを分類・整理することにより、ドメインのユースケースを抽出する。

P1-3) データモデルおよびユースケース洗練： データモデルとユースケースを相互に検証し、それぞれを洗練する。また、本プロセスで明らかにしたデータモデルとユースケースを、ドメインの要求の範囲とする。以下に手順を示す。

P1-3-1) クロスチェック1： P1-1)で抽出したアプリケーション群のERDにおいて、P1-2)で抽出したユースケースを実現するためのエンティティが、抽出されているかどうかを検証する。検証は、P1-2-3)で洗い出したユースケースのシナリオ中に出現する名詞が、ERDのエンティティまたはデータ項目として表現されているかどうかを調査することにより行う。検証結果に基づいて、ERDおよびユースケースの過不足を補い、それぞれを洗練する。

P1-3-2) クロスチェック2： ERDのすべてのエンティティが、ユースケースで使用されているかどうかを検証し、過不足を補い、ユー

スペース、ERDを洗練する。検証は、クロスチェック1と同様にして、ユースケースごとに分類したシナリオ中に、ERDのエンティティが名詞句として使用されているかどうかを調査することにより行う。使用されていないエンティティがある場合には、これらを使用するようなシナリオを検討し、ユースケースを抽出する。

P1-3-3) スコープ定義： 洗練されたドメインのERDおよびユースケースを、ドメインの要求範囲として定義する。

3.4 プロセスP2：ドメインの特徴抽出

プロセスP1で明らかにした、ドメインのERD、アプリケーションシステムのERD群、ユースケースから、ドメインのフローズスポットおよびホットスポットを洗い出す。さらに、フローズスポット、ホットスポットの接続部分を明確にする。

本プロセスの手順を以下に示す。また、本プロセスにおけるデータフローを付録の図16に示す。

P2-1) 静的側面分析： DOAにより、ドメインの静的側面の特徴を抽出する。プロセスP1で作成したドメインのERDとアプリケーションごとのERDを比較する。アプリケーションシステム群で共通のエンティティと、特定のアプリケーションシステムに固有のエンティティを識別する。

P2-2) 動的側面分析： プロセスP1で抽出したドメインのユースケースに基づいて、ドメインの動的側面の特徴を抽出する。本プロセスの詳細手順を以下に示す。

P2-2-1) エンティティ間の相互作用抽出： プロセスP1で抽出したドメインのユースケースに基づいて、ドメインのERDのエンティティ間のインタラクションを抽出し、インタラクション図を作成する。

P2-2-2) エンティティの状態遷移図： 抽出したインタラクションに基づいて、エンティティの状態遷移を記述する。

P2-2-3) 状態遷移アクション系列抽出： P2-2-1)で記述したインタラクション図および、P2-2-2)で記述した状態遷移図のイベント/アクション系列から、各エンティティの持つメソッドを抽出する。

P2-2-4) メソッドのデータフロー記述： P2-2-3)で抽出したメソッドの、データフローを記述し、メソッドの性質を調べる。すなわち、該当メソッドが、実装において、ドメインに共通

して固定的に表現可能なものか、アプリケーションシステムごとに異なるものなのかを明らかにし、アプリケーションシステムごとにカスタマイズが必要となる部分を特定する。

P2-3) 分析結果統合： P2-1)で抽出したドメインの静的特徴とP2-2)で抽出したドメインの動的特徴を統合する。ドメインのオブジェクト図を作成する。本プロセスの詳細手順を以下に示す。

P2-3-1) エンティティの役割の明確化： P2-2)で作成したインタラクション図とドメインのERDから、エンティティの役割を明確にする。また、アプリケーションごとに異なるエンティティでも、共通の役割を持つ、参照マスタなどのエンティティを抽出する。

P2-3-2) エンティティの抽象化： 共通の役割を持つエンティティについて、上位の概念により抽象化が可能な部分を洗い出す。抽象化可能な部分について、抽象クラスを導入し、ERDの関連を見直し、オブジェクト図作成の準備を行う。

P2-3-3) メソッドの抽象化： P2-2-4)で記述したメソッドのデータフローを比較し、抽象化の検討を行い、抽象メソッドを抽出する。比較と抽象化の対象は、P2-3-1)で共通の役割を持つと判断したエンティティの持つメソッドおよび、1つのエンティティが持つメソッドのうち、アプリケーションに依存しているメソッド群などである。前者のメソッドについては、共通の役割を表現する抽象クラスが持つメソッドを抽出する観点で抽象化する。後者のメソッドについては、アプリケーションごとの違いに関係なく、エンティティの役割を表現するメソッドを抽出する観点で、抽象化する。

抽象化は次のように行う。メソッド間で共通のプロセスの流れおよびメソッドごとに異なるプロセスの流れを分離する。メソッドごとに異なるプロセスの流れの役割を考慮して、一般化する。メソッド間で共通のプロセスの流れと一般化したプロセスの流れ全体をデータフローとして持つメソッドを、抽象メソッドとする。抽象メソッド名を定義する。

P2-3-4) メソッドのマッピング： P2-2)で洗い出したメソッドおよびP2-3-3)で抽象化したメソッドから、ERDのエンティティおよび抽象クラスのメソッドとして考えられるもの

を選択する。P2-3-3)で新規に作成した抽象メソッドおよびそれらをカスタマイズしたメソッドを所有する望ましいエンティティおよび抽象クラスが抽出されていない場合には、新規に具象クラス/抽象クラスを追加する。エンティティの抽象化と、メソッドの付与によって、ERDからオブジェクト図を作成する。新規に追加したクラスを考慮して、クラス間の関連を見直し、オブジェクト図を洗練する。

P2-4) フローズンスポット、ホットスポット抽出：

P2-3)で作成したオブジェクト図において、P2-1)で抽出したアプリケーション固有/共通のエンティティ、P2-2)で抽出したアプリケーション固有/共通のメソッドに基づいて、フローズンスポットとホットスポットを抽出する。フローズンスポットは、オブジェクト図において、アプリケーションシステム群に共通した抽象クラスと具象クラスの集合およびそれらの関連の集合の観点で抽出する。ホットスポットは、アプリケーションシステムの違いによって、バリエーションが考えられる抽象クラスと具象クラスの集合およびそれらの関連の集合の観点で抽出する。

P2-5) フローズンスポットとホットスポットの接続

部分の抽出： P2-4)で抽出したフローズンスポットおよびホットスポットにおいて、これらの接続部分となるクラスを抽出する。接続部分となるクラスは、フローズンスポットおよびホットスポットの両方に含まれ、アプリケーションごとに異なる部分が発生する分岐点となるクラスである。たとえば、アプリケーションごとに異なるサブクラスが考えられる抽象クラス、部分クラスの数と種類がアプリケーションごとに異なる、それらの集約クラス、これら以外のクラスであり、該当クラスに関連する別のクラスの数と種類がアプリケーションごとに異なるクラスなどである。

3.5 プロセス P3：フレームワークパターンの抽出 **その1**

プロセス P2で洗い出したフローズンスポット、ホットスポットに基づいて、オブジェクト図からフレームワークパターンを抽出する。本プロセスの手順を以下に示す。また、本プロセスにおけるデータフローを付録の図 17に示す。

P3-1) フローズンスポットの洗練： プロセス P2で抽出したフローズンスポットを、「汎化-特化」および「集約-分解」の観点から洗練する。各観点

の詳細を以下に示す。

P3-1-1) 汎化-特化の観点による FS の洗練：

ホットスポットから、共通する特性がさらに抽出できないかどうかを検証する。共通する特性を、上位の抽象化されたクラスとして表現し、フローズンスポットへの追加を検討し、フローズンスポットを洗練する (FS：フローズンスポット)。

P3-1-2) 集約-分解の観点による FS の洗練：

ホットスポットから、対象物の構成関係に着目したグループ化を行い、対象物に相当するクラスを洗い出す。抽出したクラスを、フローズンスポットへ追加することを検討し、フローズンスポットを洗練する。

フローズンスポットの「集約-分解」で表現されたクラス間の関係において、集約されたクラスの構成要素として、アプリケーション群に共通の要素を過不足なく抽出し、フローズンスポットの見直しを行う (FS：フローズンスポット)。

P3-2) 基本パターンの抽出： プロセス P3-1)で洗練したフローズンスポットにおいて、クラス間の関係を、拡張性、効率性、移植性などの観点から検証し、アプリケーションシステムごとの違いによらない構造を決定し、その結果を基本パターンと定義する。プロセス P2において抽出した、フローズンスポットとホットスポットの接続部分を、洗練されたフローズンスポットに基づいて見直す。基本パターンには、見直したフローズンスポットとホットスポットの接続部分となるクラスを含める。

P3-3) ホットスポットの見直し： プロセス P3-2)で抽出した基本パターンが生成される過程で、追加・変更されたクラスやクラス間の関係に基づいて、ホットスポットを見直す。たとえば、基本パターン抽出過程において、追加された抽象クラス、集約クラスに基づいて、これらのバリエーションをホットスポットに追加する。

P3-4) ホットスポットの分割： ホットスポットを、同じ意味や構造に基づくグループに分類し、アプリケーションシステムごとに取捨選択される最小単位に分割する。分類・分割した結果を、フレームワークパターンの組込みパターン、付加パターンの候補とする。本プロセスの分類・分割の観点と詳細手順を以下に示す。なお、以下の手順により抽出する各組込みパターン/付加パターンの候補には、プロセス P2で抽出し、プロセス P3-3で

見直した、フローズンスポットとホットスポットの接続部分を含める。

P3-4-1) スーパー/サブクラスに基づく分類・分割：

基本パターンの持つ接続部分における抽象クラスに対して、そのサブクラスとなるものを、ホットスポットの候補から洗い出し、抽象クラスごとにグループ化する。分類した1つのグループにおいて、抽象クラスと1つのサブクラスの関係に分割する。

P3-4-2) 集約とその部分に基づく分類・分割：

基本パターンの持つ接続部分における集約クラスに対して、その部分クラスの関係となるものを、ホットスポットの候補から洗い出し、集約クラスごとにグループ化する。分類した1つのグループにおいて、集約クラスと1つの部分クラスの関係に分割する。

P3-4-3) 共通な関連を持つクラスの集合に基づく分類・分割：

P3-4-1) および P3-4-2) で行った「汎化-特化」, 「集約-部分」の分類・分割以外で、アプリケーション間で共通の関係を洗い出し、それらの関係に基づいて、ホットスポットの候補を分類する。たとえば、履歴データの持ち方などは、アプリケーションごとに効率性、拡張性、移植性の違いによって、クラスや関連が異なる場合が考えられる。そのような履歴データの持ち方などを、ホットスポットの候補から分類する。分類したグループにおいて、アプリケーションの違いによって異なる部分を、最小単位に分割する。

P3-5) 組込み/付加パターンの抽出： P3-4) で分類・分割した組込み/付加パターンの候補を、「選択-組込み-実装」, 「選択-カスタマイズ-実装」の観点により分類し、アプリケーションの違いによるそれぞれのバリエーションを追加し、組込みパターン、付加パターンを生成する。本プロセスの詳細を以下に示す。なお、以下の手順により抽出する各組込みパターン/付加パターンには、プロセス P3-4) で各パターンの候補の持つ、フローズンスポットとホットスポットの接続部分を含める。

P3-5-1) 選択-組込み-実装の観点による分類：

P3-4) で分類したグループから、アプリケーションシステムの違いによって修正することなく、基本パターンに直接組み込むことにより実装可能な要素の集合を識別し、これらを組込みパターンとする。

組込みパターンは、基本パターンと共通して

持つ、フローズンスポットとホットスポットの接続部分となるクラスに基づいて、重ね合わせるにより実装する。

P3-5-2) 組込み型バリエーションの生成： P3-5-1) において抽出されたグループごとに、構成要素のバリエーションを追加する。

P3-5-3) 選択-カスタマイズ-実装の観点による分類：

P3-4) で分類したグループから、アプリケーションシステムの違いによって、カスタマイズして、基本パターンに付加することにより実装する要素の集合を識別し、これらを付加パターンとする。

付加パターンは、アプリケーションシステムの特性に応じてカスタマイズした後、基本パターンと共通して持つ接続部分となるクラスに基づいて、重ね合わせるにより実装する。

P3-5-4) 付加型バリエーションの生成： P3-5-3) において抽出されたグループごとに、他の構成要素のバリエーションを追加する。

**3.6 プロセス P4：フレームワークパターンの抽出
その2**

プロセス P2 で明らかにしたフローズンスポットとホットスポットの接続部分に基づいて、プロセス P3 により抽出した基本、組込み、付加パターンを合成する。合成した結果を事例パターンとする。本プロセスの手順を以下に示す。また、本プロセスにおけるデータフローを付録の図 18 に示す。

P4-1) フレームワークパターンの分類： 再構築するためのアプリケーションシステムの特徴に適合する組込みパターンおよび付加パターンの集合を洗い出す。

P4-2) 組込みパターンの選択： プロセス P3 で抽出した基本パターンのフローズンスポットとホットスポットの接続部分となるクラスに基づいて、上記手順で洗い出した組込みパターンを合成する。

P4-3) 付加パターンのカスタマイズ： アプリケーションシステムの特徴に基づいて付加パターンをカスタマイズする。

P4-4) 事例パターンの抽出： P4-2) で合成した結果において、接続部分となるクラスに基づいて、付加パターンを統合する。この結果を事例パターンとして蓄積する。

4. 事 例

本章では、販売管理システムを対象にし、3章で述

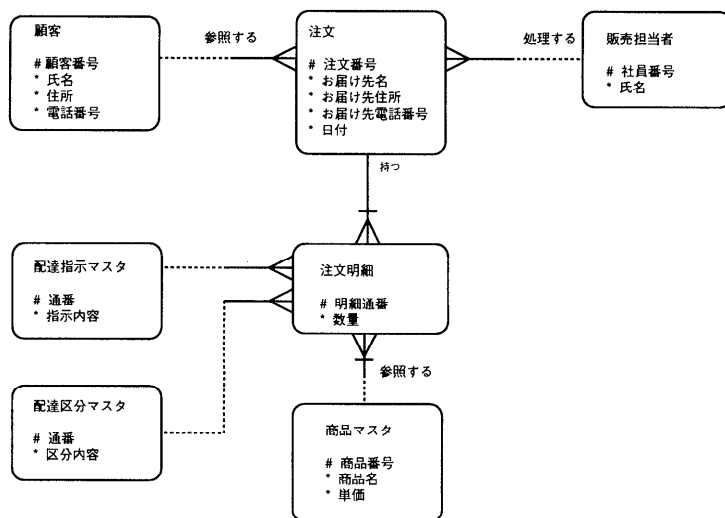


図3 販売管理アプリケーションシステム事例1のERD

Fig. 3 An example of ERD for sales systems.

べたフレームワーク構築手法の適用事例について述べる。以下では、販売管理システムドメインの概要を述べる。次に、我々の提案するフレームワーク構築手法の主要プロセスにより作成される、主な成果物の事例を示す。なお、4.2節～4.5節において、文中の成果物名の後ろに記述した括弧内の文字列は、3章で提案したフレームワーク構築プロセスのプロセス番号である。

4.1 ドメイン概要

販売管理システムは、百貨店、通信販売会社などの販売部門の業務を支援する。この部門では、実績のある顧客に対するリストを作成し、顧客にダイレクトメール、カタログ、注文用紙を発送する。顧客から注文がなされると、注文伝票および配送伝票を作成し、配送部門に送付する。商品の発送前には、商品の在庫を確認し、在庫が不足の場合には商品の発注を行う。また、シーズンや年度の変り目などのタイミングで実績顧客リストを更新する。販売管理システムは、受注処理、配送伝票発行処理、顧客マスタ、商品マスタなどのマスタファイル保守処理などを行う。

4.2 プロセスP1) スコープの決定：成果物事例

プロセスP1) スコープの決定では、プロセスP1-1) 各アプリケーションシステムのERDをDOAにより作成し、これらを統合して、ドメインのERDを作成する。また、プロセスP1-2)により、ドメインのユースケースを抽出する。次に、プロセスP2-3)により、ユースケースとERDの間において、クロスチェックを行い、互いに洗練する。本プロセスの主な成果物は、アプリケーションシステムごとのERD (P1-3)、ドメインのユースケース図 (P1-3)、ドメインのERDであ

る (P1-3)。プロセスP1-3)によりデータ項目の標準化・正規化を行った結果であるアプリケーションシステムごとのERDを、図3および図4に示す。なお、本ERDの表記方法は、文献15)に基づく。図3および図4は、現行システムで使用される注文用紙、配送伝票、マスター一覧などの画面・帳票類を収集し、それらに基づいて作成した。事例1のシステムの注文用紙には、依頼主の情報と、複数の配達商品を記述する。配達先は依頼主自身である。事例2のシステムは、お中元やお歳暮時期の贈答品の注文処理を行うシステムである。後者の注文用紙には、依頼主の情報と複数の配達先と配達商品を記述する。また、のし指示、包装指示などの指示を記述する。さらに、事例2のシステムは、顧客の履歴情報の管理、配達伝票の発行などの処理も行う。

プロセスP1-3)により作成した、ドメインのユースケースを図5に示す。図5は、本ドメインのユースケースとアクターとの関係 (図では、矢印が相互作用があることを示す) を表す。企業の内部のアクターは、販売窓口担当、販売顧客管理担当、経理担当、在庫担当、配送担当、販売マスタ管理担当である。企業の外部のアクターは、顧客、メーカー、配送業者、届け先である。事例で抽出した、販売管理ドメインのユースケースは、受注処理、顧客管理、入金/支払処理、在庫管理、配送手配、商品配送、マスタメンテナンスである。

プロセスP1-3)により作成した、ドメインのERDを図6に示す。なお、本ERDの表記方法は、文献15)に基づく。図6は、図3と図4のERDに対し

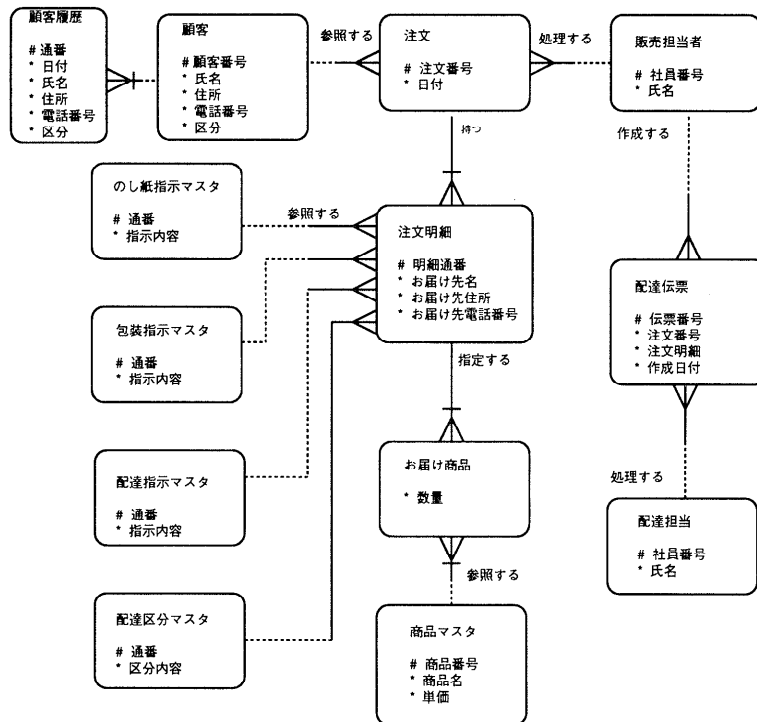


図4 販売管理アプリケーションシステムの事例2のERD

Fig. 4 Another example of ERD for sales systems.

て、用語の標準化、正規化、データの持ち方の検討、ドメインのユースケースの情報の追加を行って見直したERDである。図3と図4の統合では、主に次のことを考慮した。

- 事例1に事例2のマスタ関係のエンティティを追加する。
- 注文、注文明細に関する構造を変更する。すなわち、お届け先の情報を記述するための配送伝票エンティティを追加し、両者の場合の関係を表現可能とする。
- 顧客履歴について、履歴として保存する情報の種類の追加・変更に備えて、履歴の対象となるエンティティを追加する。

図6において、網掛けのエンティティは、プロセスP1-3-1のERDとユースケースのクロスチェックにより、現行のERDに追加したものである。追加手順は、まず、プロセスP1-2-3で洗い出したユースケースのシナリオ中に出現する名詞句が、ERDのエンティティまたはデータ項目として表現されているかどうかを調査する。次に、表現されていないと判断した名詞句を、ERD上にエンティティまたはデータ項目の形式で追加し、リレーションシップの検討を行い、ERDを洗練する。なお、本ERDでは、説明を簡潔にするため、

ユースケースのマスタメンテナンスに関する表現を省略した。

4.3 プロセスP2)ドメイン特徴抽出：成果物事例

プロセスP2)ドメインの特徴抽出では、プロセスP2-1においてドメインの静的側面の特徴を明確にし、プロセスP2-2において動的側面の特徴を明確にし、プロセスP2-3においてこれらの結果を統合し、ドメインのオブジェクト図を作成する。また、プロセスP2-4では、オブジェクト図におけるフローズスポットとホットスポットを洗い出し、プロセスP2-5でフローズスポットとホットスポットの接続部分を抽出する。本プロセスの主な成果物は、ドメインのERDに基づくホットスポットの候補(P2-1)、ユースケースに基づくエンティティ間のインタラクション図(P2-2-1)、エンティティの状態遷移図(P2-2-2)、メソッドのデータフロー図(P2-2-4)、ドメインのオブジェクト図(P2-3-4)および、オブジェクト図に基づくフローズスポット/ホットスポット(P2-4)である。プロセスP2-2-2において抽出した、図6のドメインのERDにおけるホットスポットの候補を図7に示す。図7の太枠で囲んだエンティティは、図3と図4の比較結果から、図6において、アプリケーションごとに異なるエンティティを抽出したものであり、これらがホット

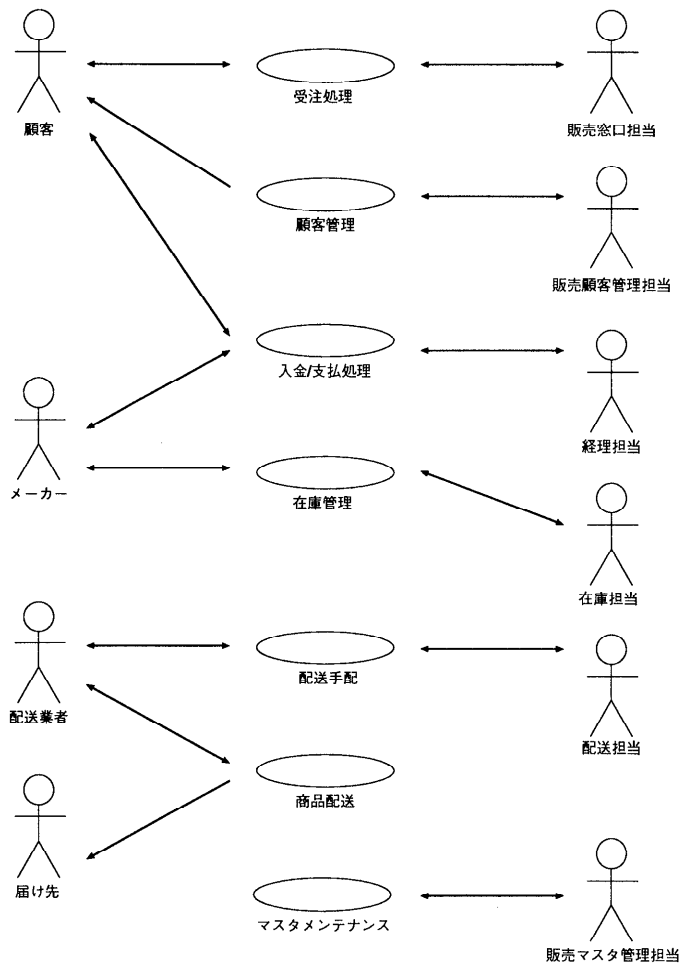


図5 販売管理システムドメインのユースケース

Fig. 5 An example of use cases for sales systems.

スポットの候補となる。

プロセス P2-2-1 において抽出した、ユースケース“受注処理”に基づくエンティティ間のインタラクション図を図8に示す。図8において、点線の欠印は、それらで囲まれたインタラクションの繰返しを表す。図8において、“のし紙指示マスタ”は、“包装指示マスタ”などの別のマスタに変更しても、他のエンティティ間とのインタラクションは変化しない。すなわち、アプリケーションシステムの違いにより、使用する指示マスタの種類は異なっても、これら指示マスタと他のエンティティとのインタラクションは、ドメインに共通していることが考察できる。このような共通したインタラクションの情報に基づいて、プロセス P2-3-1 では、共通の役割を持つエンティティを抽出する。また、共通の役割を持つエンティティは、プロセス P2-3-2 において、抽象化するエンティティの候補となる。

共通のインタラクションを持つエンティティ“のし紙指示マスタ”、“包装指示マスタ”などの“指示マスタ”エンティティの状態遷移図を図9に示す。共通のインタラクションを持つエンティティ群は、共通の役割を持つエンティティの候補および抽象化するエンティティの候補になる。

状態遷移などの観点で、このようなエンティティの性質を明らかにしておく、エンティティの抽象化の候補の選択がしやすくなることが考えられる。図8および9により、各エンティティの役割、メソッド候補、それらメソッドの実行順序が明確になる。

プロセス P2-2-4 により抽出した、指示マスタエンティティにおける情報取出しメソッドのデータフローを図10に示す。図10では、受け付ける指示要求の種類や検索方法がアプリケーションごとに異なることが明らかになる。メソッドのデータフローを記述する

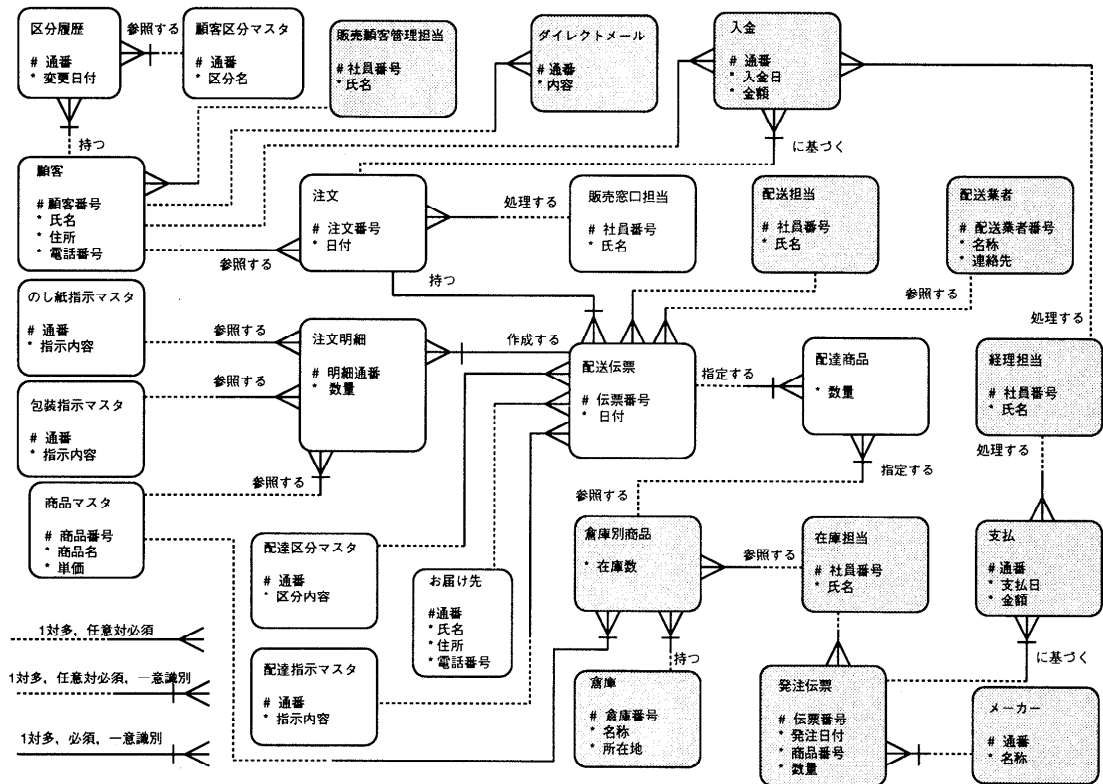


図 6 販売管理システムドメインの ERD 事例
Fig. 6 An example of ERD for a sales system domain.

ことによって、アプリケーションごとに異なるかどうかおよび異なる部分を特定することが可能になる。これらの情報は、プロセス P2-3-3 において、抽象メソッドを抽出する際に活用する。

プロセス P2-3-4 により抽出したドメインのオブジェクト図を図 11 に示す。図 11 において、網掛け部分 (a)~(f) は、プロセス P2-4 で抽出したホットスポットであり、それ以外がフローズスポットである。ただし、フローズスポットは、ホットスポットとの接続部分を含む。図 11 の太枠で囲まれたクラスが、フローズスポットとホットスポットの接続部分である。これらの接続部分は、プロセス P2-5 において抽出する。

図 11 は、プロセス P2-1 で抽出した図 7 のホットスポットの候補に着目し、プロセス P2-3-2 においてエンティティの抽象化を行い、プロセス P2-3-3 において抽象化したメソッドを、プロセス P2-3-4 においてマッピングすることにより作成する。エンティティの抽象化では、指示マスタなどのように、アプリケーションごとに種類と数が異なるが、共通化できる部分を、抽象クラスとして表現する。メソッドは、図 8, 図 9 によって抽出したメソッドを、各エンティティに割り付

ける。ドメインに共通のメソッドでも、図 10 によって実装がアプリケーションごとに異なるメソッドは、抽象メソッドでインタフェースを表現し、実装をサブクラスで実現する構造とする。図 11 において、“明細構成要素”, “指示マスタ” “配送伝票構成要素”, “配達マスタ”, “顧客履歴種別”, “履歴種別マスタ” は抽象クラスである。またこれらのオブジェクトに記述されたメソッドは、抽象メソッドである。

各ホットスポットの内容は、以下のとおりである。ホットスポット (a) は、“顧客の履歴を持つかどうか”である。ホットスポット (b) は、“履歴の持ち方”である。ホットスポット (c) は、“注文明細の構成要素の種類と数”である。ホットスポット (d) は、“注文明細の構成要素が参照する指示マスタの種類と数”である。ホットスポット (e) は、“配送伝票の構成要素の種類と数”である。ホットスポット (f) は、“配送伝票の構成要素が参照する配達マスタの種類と数”である。

4.4 プロセス P3, P4) フレームワークパターン抽出その 1, その 2 : 成果物事例

プロセス P3) フレームワークパターン抽出その 1 では、プロセス P2) において抽出した、フローズス

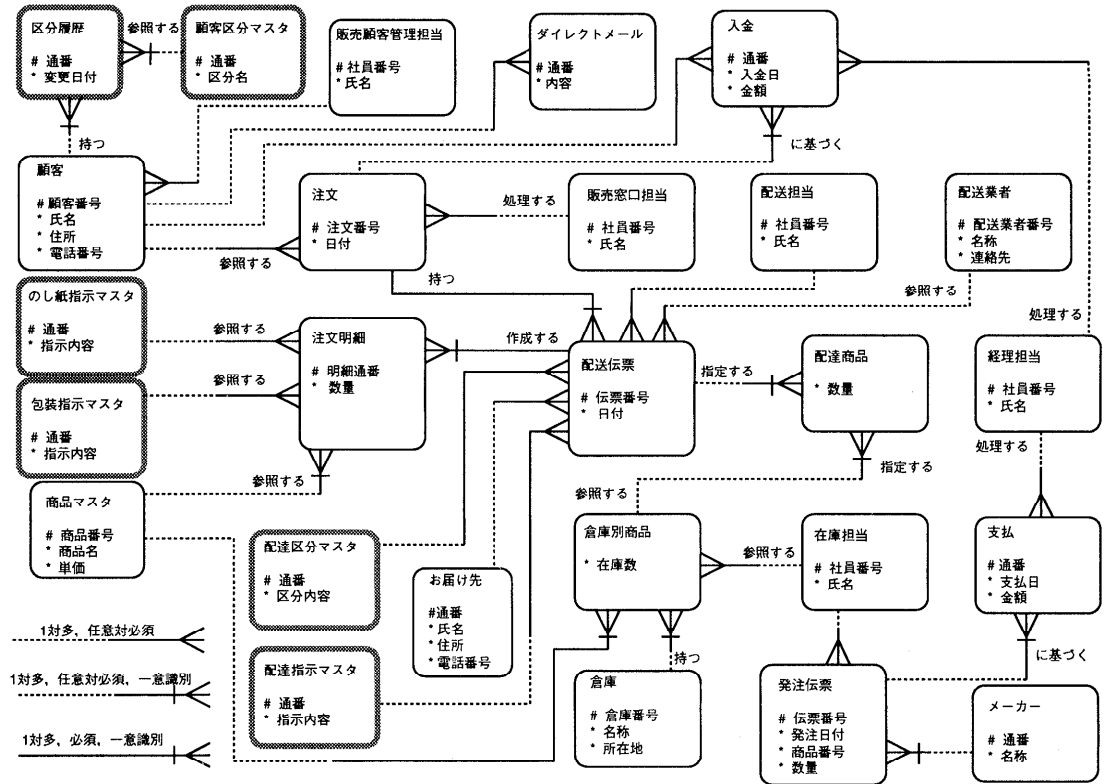


図7 ドメインのERD図におけるホットスポット候補
Fig. 7 Examples of hot-spots for ERD.

ポット, ホットスポットに基づいて, 基本/組込み/付加パターンを抽出する. 本プロセスの主な成果物は, 基本パターン (P3-2), 組込みパターン (P3-5), 付加パターン (P3-5) のフレームワークパターンである. プロセス P4) フレームワークパターン抽出その 2 では, プロセス P2-5 で明らかにしたフローズスポットとホットスポットの接続部分を支点として, プロセス P3 で抽出した, 基本/組込み/付加パターンを合成し, 事例パターンを抽出する. 本プロセスの主な成果物は事例パターン (P4-4) である.

プロセス P3-4 から P3-5 における, 組込み/付加パターンの抽出プロセスを図 12 に示す. 図 12 の事例 A は, 図 11 に示したホットスポット (c) についての, 組込みパターンの抽出プロセスである. 図 12 事例 A では, プロセス P3-4-1 において, プロセス P2-4 で抽出したホットスポットに着目し, ホットスポットをスーパークラスとサブクラスの観点で分割する. 分割したホットスポットを持つサブクラスは, アプリケーションごとにカスタマイズが必要ないため, プロセス P3-5-1 において, 組込みパターンとして分類する. 次に, プロセス P3-5-2 において, 組込みパターンのバ

リエーションを生成する.

図 12 の事例 B は, 図 11 に示したホットスポット (d) についての, 付加パターンの抽出プロセスである. 図 12 事例 B では, プロセス P3-4-1 において, プロセス P2-5 で抽出したホットスポットに着目し, ホットスポットをスーパークラスとサブクラスの観点で分割する. 分割したホットスポットを持つサブクラスは, 実装の際にアプリケーションごとに, カスタマイズが必要となるので, プロセス P3-5-3 において, 付加パターンとして分類する. 次に, プロセス P3-5-4 において, 付加パターンのバリエーションを生成する.

各フレームワークパターンの事例を図 13 に示す. 図 13 (イ) は基本パターンを, 図 13 (ロ) は組込みパターンを, 図 13 (ハ) は付加パターンを, 図 13 (ニ) は事例パターンを表す. 基本/組込み/付加パターンには, フローズスポットとホットスポットの接続部分, すなわち, 基本パターンと組込み/付加パターンとの接続部分となるクラス (図 13 では太枠の部分) を含める. 図 13 の点線矢印は, パターンの重ね合わせ先を表現している. 図 13 (イ) は基本パターンに, 図 13 (ロ) および (ハ) を, 点線矢印に従って接続部分と

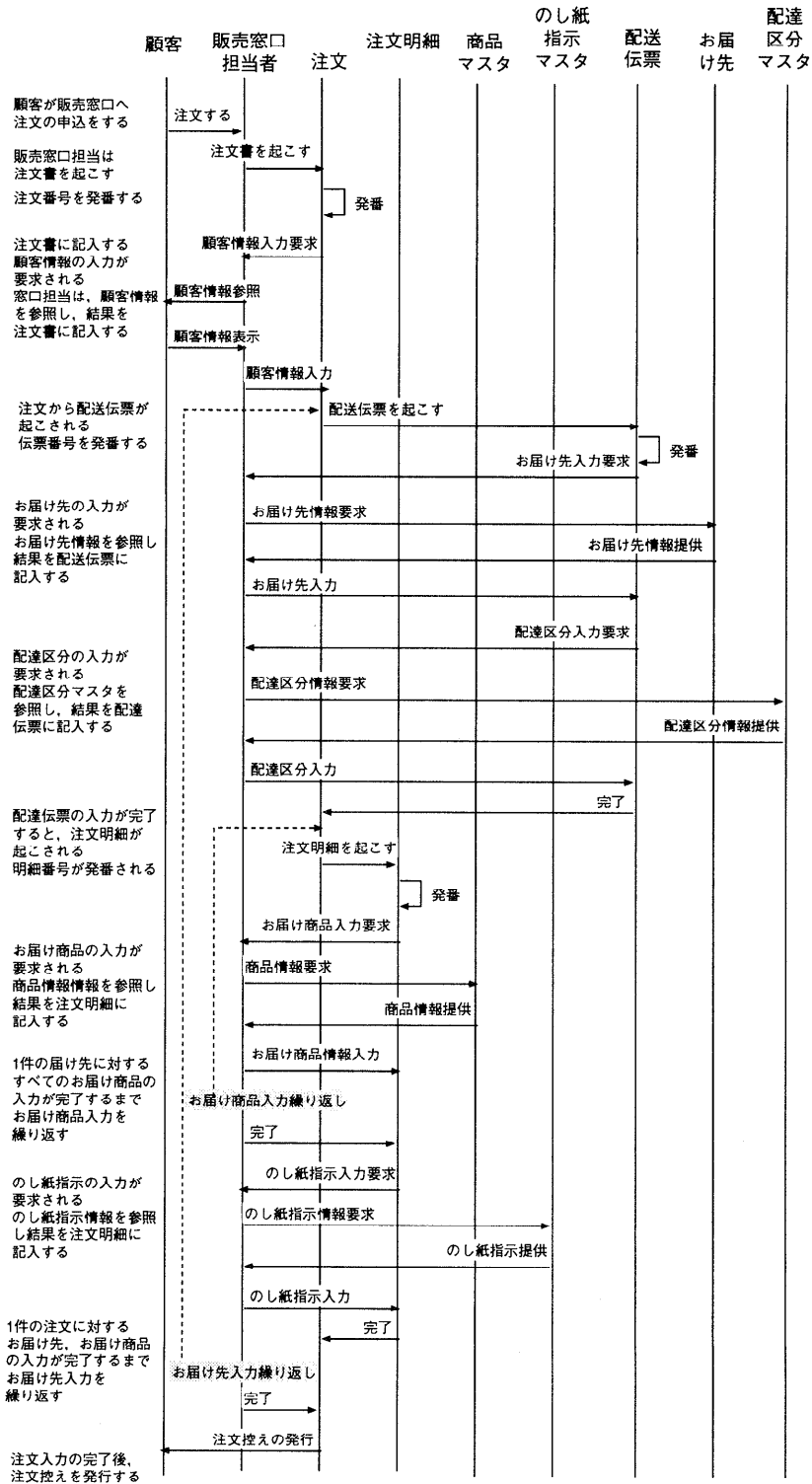


図 8 ユースケース「受注処理」のインタラクション図
 Fig. 8 An example of event trace for input_order.

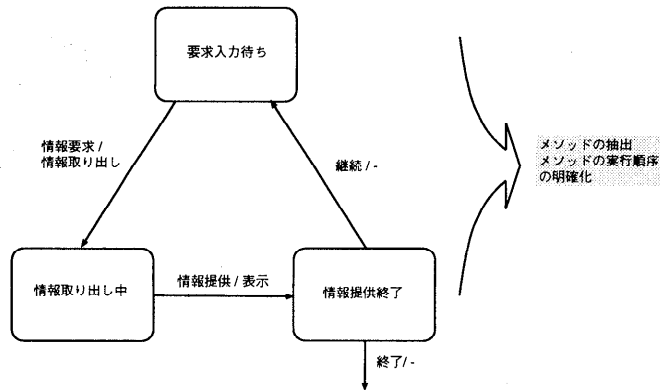


図 9 指示マスタ状態遷移図事例

Fig. 9 An example of state transition diagrams for master file.

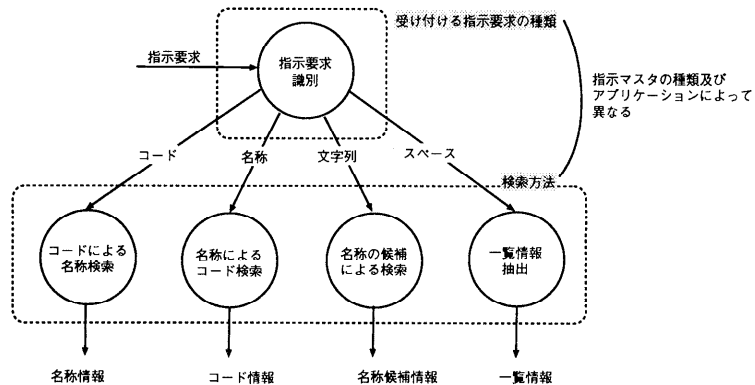


図 10 指示マスタ情報取り出しメソッド DFD 事例

Fig. 10 An example of data flow diagrams for a data extraction method of master file.

なるクラスを重ね合わせることにより、図 13 (二) 事例パターンを作成する。図 13 (二)において、背景が網掛けの部分は、組込みパターンを示し、背景が斜線の部分は、付加パターンを示し、それ以外の部分と太枠のクラスとが基本パターンを示す。なお、図 13 は、表現の簡潔化のため、オブジェクト名のみを記述した。各オブジェクトの属性とメソッドは、図 11 の同一名称のオブジェクトと同様である。

5. 評 価

本章では、4 章で示した適用事例の結果から、我々の提案するフレームワーク構築プロセスの有効性を考察する。以下では、本手法の主要プロセスの有効性と限界について論じる。また、本手法の適用可能性について検討する。

プロセス P1 において、DOA により抽出したドメインの ERD および、ユースケースにより、既存システムとシステムのあるべき姿の分析結果をクロスチェ

クする。クロスチェックの目的は、ドメイン分析の結果から抜けを排除し、かつ、ドメイン分析結果に柔軟性を持たせることである。このような目的を達成するためには、DOA による ERD の抽出および、ユースケースの抽出において、次の 2 つの条件を満たすことが必要であると考えられる。第一に、DOA によってドメインの ERD を作成する際、分析対象となる既存のアプリケーションシステム群から、使用されている画面帳票類をできるだけ多く収集することである。4 章で示した我々の適用事例では、事例 1 の画面帳票として約 500、事例 2 では約 700 枚の画面帳票を収集し、それらについて DOA によるデータ整理を行っている。

第二に、ユースケースの抽出および記述には、顧客の要求を反映させ、新規システム開発への拡張性および柔軟性を考慮することが必要である。しかし、我々の方法論においては、ユースケースの記述の詳細度を提示していない。したがって、技術者による顧客へのインタビュー能力、インタビュー結果の記述能力、顧

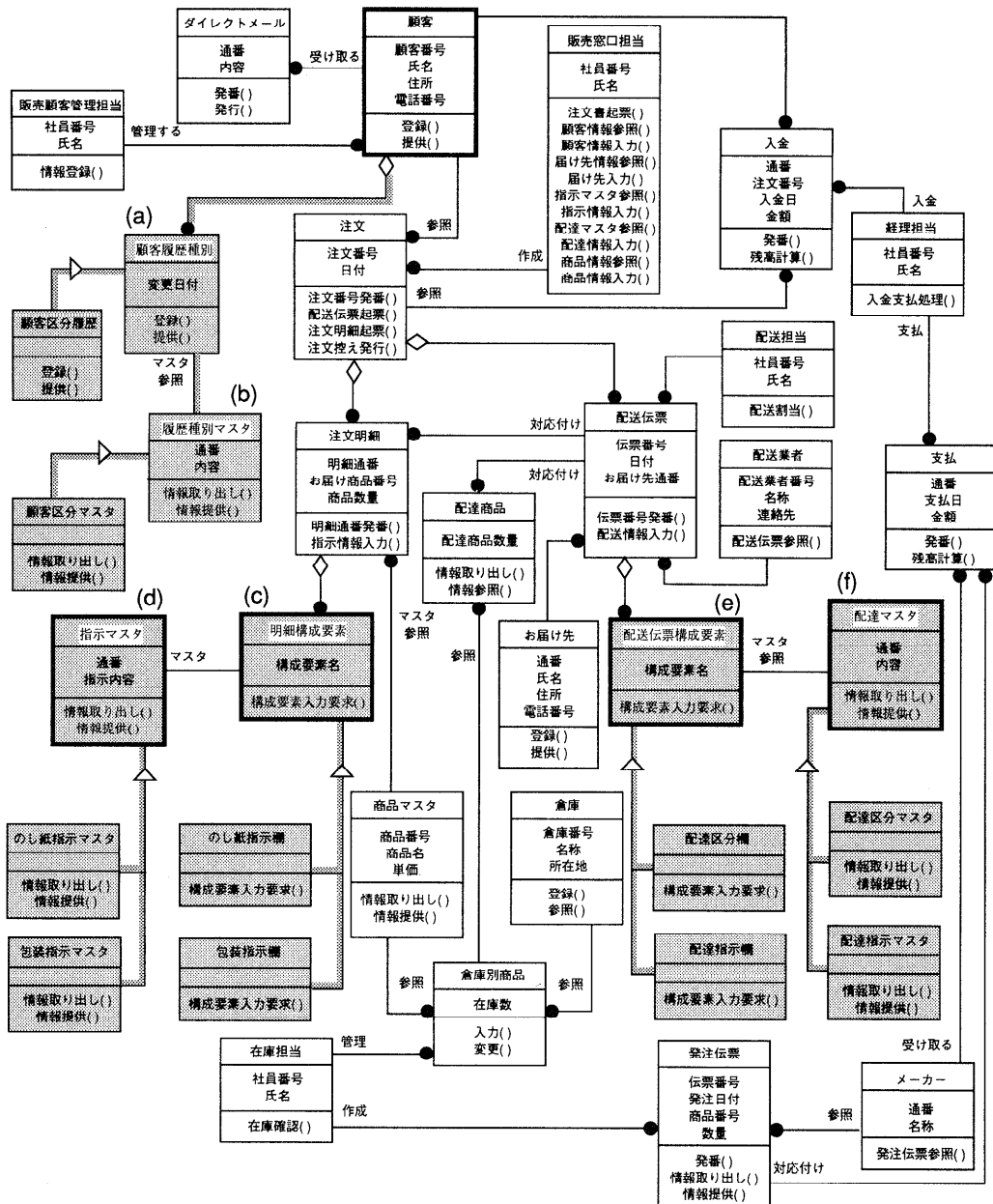


図 11 販売管理システムオブジェクト図事例
Fig. 11 An example of object diagrams for sales systems.

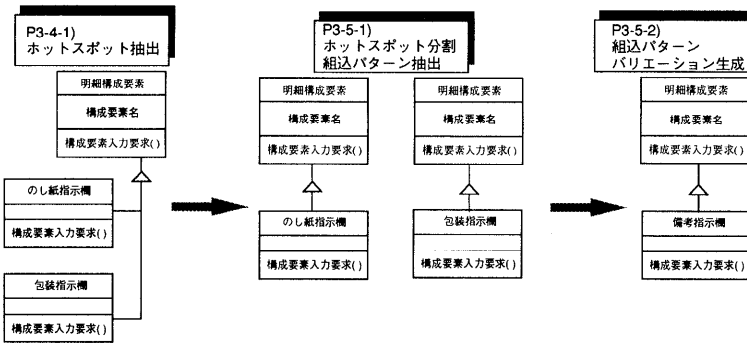
客のドメイン知識の量，将来システムへの問題意識の程度などにより，抽出されるユースケースのシナリオの詳細度にはばらつきがあることが考えられる．第二の条件を満たすためには，ユースケース抽出における作業手順の詳細化および表現形態の定義，記述の詳細度へ指針の提示が必要であると考えられる．

プロセス P2 では，ドメインの ERD とユースケースから，抽象クラスの抽出，メソッドの抽出とマッピ

ングを行いドメインのオブジェクト図を作成し，ドメインのフローズスポットとホットスポットを特定する．これらのプロセスにおいて，共通の役割を持ったエンティティの抽出，エンティティの抽象化の度合いの検討，メソッドのエンティティへの最適な配置，フローズスポットとホットスポットの接続部分の抽出などの作業に関しては，我々の方法論に加えて，技術者の試行錯誤や経験が必要である．このような観点か

●フレームワークパターン抽出：事例A

(c) ホットスポット：注文明細の構成要素の種類と数に着目した場合



●フレームワークパターン抽出：事例B

(d) ホットスポット：注文明細の構成要素が参照する指示マスタの種類と数に着目した場合

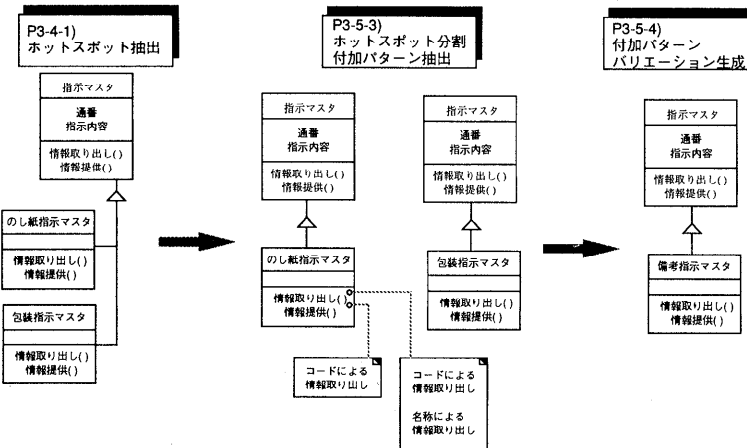


図 12 組込み/付加パターン抽出プロセス事例

Fig. 12 Examples of processes for extracting build-in/build-on patterns.

らの支援については、今後の課題である。

プロセス P3 において、プロセス P2 で明らかにしたオブジェクト図におけるフローズスポットとホットスポットを、フレームワークパターンに分割する。フローズスポットとホットスポットの位置が特定されたドメインのオブジェクト図は、1つのフレームワークである。フレームワークの利用方法として、フレームワークパターンとして分割されたパターンのカスタマイズと合成による方法以外に、フレームワークのホットスポットに直接、修正や追加を加える方法も考えられる。しかし、システムの規模、複雑さが増すと、フレームワークを理解すること、ホットスポットの場所を特定することが困難になると考えられる。フレームワークパターンへ分割することは、カスタマイズの

範囲を、組込み/付加パターンに特定するので、ユーザの修正や追加の注力範囲を絞り込むことができると考えられる。しかし、組込みパターンの選択および付加パターンのカスタマイズには、パターンを使用するユーザは、パターンを理解すること、ドメインの知識を持っていること、実装の経験的知識があることなどが必要である。そのようなパターンの利用に関する支援については、今後の課題である。

我々の提案するフレームワーク構築手法は、事務処理系のデータベースへの処理が中心となるドメインに対して有効である。我々の手法において重要な技術となる DOA は、本来、事務処理分野のデータベースを再構築するための手法である。DOA により作成した ERD は、そのようなドメインの本質部分であるデー

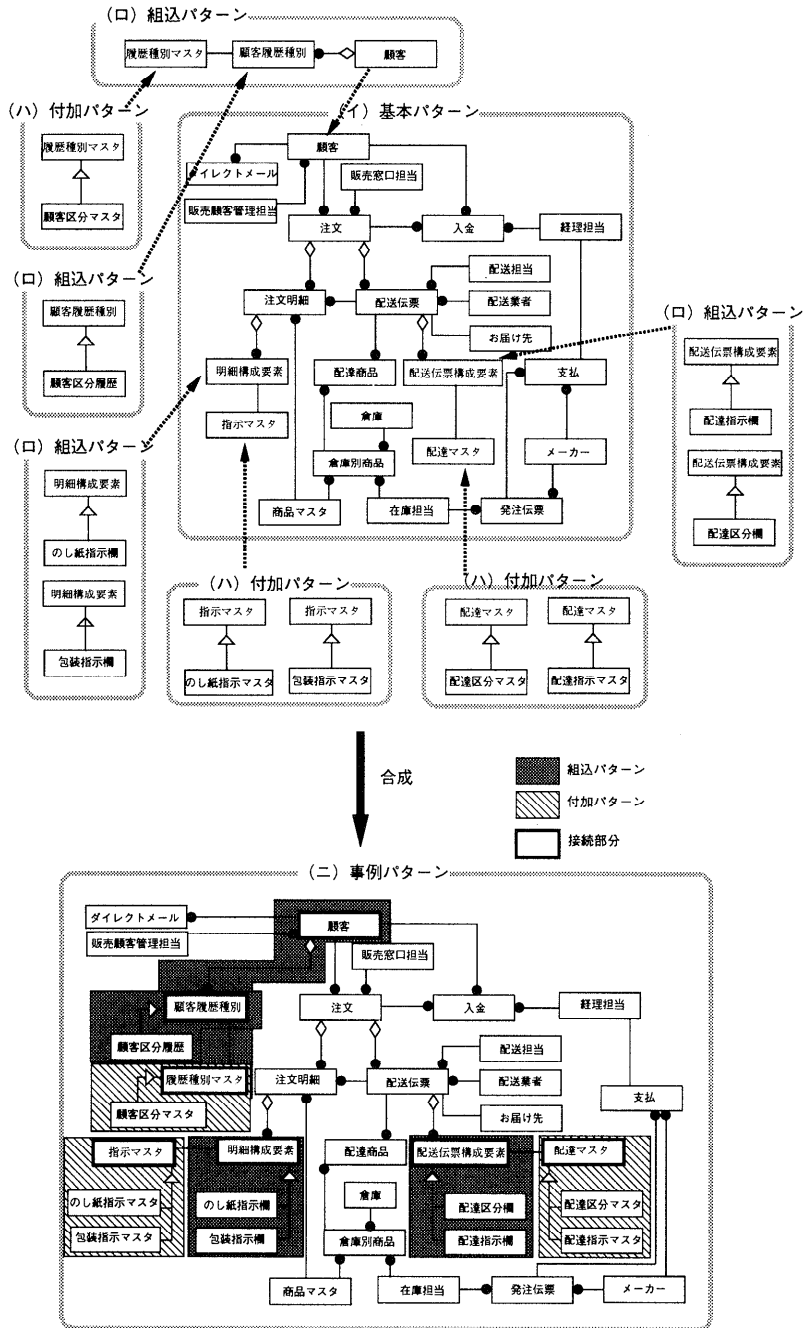


図 13 フレームワークパターン事例
 Fig.13 Examples of framework patterns.

データベースの構造を示す。DOA 手法は、作業手順が明確なので、ドメインの本質的な部分を示す ERD を、手続き的に後戻りが少なく作成できる。

一方、データベースが中心ではないドメインの場合には、DOA による ERD では、ドメインの特徴を十分に洗い出すことができない可能性がある。その場

合は、ユースケースの側面からの特徴抽出のウェイトが高くなる。上で述べたように、我々の方法論では、ユースケースの抽出/記述の詳細を規定していないので、ユースケースの品質が安定しないことなどが予想される。したがって、我々の提案するフレームワーク構築手法は、そのようなドメインへの適用効果は低い

と考えられる。

6. 関連研究

本章では、フレームワークの構築手法、ドメイン分析手法の観点から、我々の提案するフレームワーク構築手法と関連する研究を取り上げて比較する。

Pree は、フレームワークを効率的に開発するために、既存のオブジェクト指向分析/設計方法論を拡張したホットスポット駆動アプローチを提案している²⁾。このアプローチの主な手順は、(1) オブジェクト/クラスの識別、(2) ホットスポットの識別、(3) フレームワーク(再)設計、(4) フレームワーク適合、である。(1) は、オブジェクト指向ソフトウェア開発の出発点であり、そのドメインにおけるオブジェクトやクラスを特定する。(2) では、ドメイン分析を行い、アプリケーションごとに異なるホットスポットを特定し、ホットスポットの柔軟性を調査する。(3) では、ホットスポットの特徴と柔軟性の度合いを考慮して、メタパターンを選択し、フレームワークを構築する。メタパターンは、フレームワークのホットスポットおよび、ホットスポットの柔軟性に基づいて分類・表現される、フレームワークの設計情報を提供するパターンである。(4) では、ホットスポットが十分にドメインの特徴を表現しているかどうかに着目し、フレームワークの適合を改良する。

Pree のアプローチでは、ホットスポットの柔軟性に基づいてメタパターンを選択し、メタパターンはフレームワークの設計情報の輪郭を提供する。ホットスポットを見つけるためには、ドメインに固有の知識が必要であり、そのためにはドメイン分析が不可欠である。しかし、Pree の方法論には、そのようなドメイン分析のための手順が提供されていない。我々の提案するフレームワーク構築手法では、DOA とユースケースに基づいてドメインの特徴を抽出し、ドメインのホットスポットを特定し、フレームワークパターンにより表現するための手順を提供している。

既存のドメイン分析手法としては、Feature Oriented Domain Analysis (以下、FODA と呼ぶ)⁴⁾、Organizational Domain Modeling (以下、ODM と呼ぶ)⁵⁾、Domain Analysis & Reuse Environment (以下、DARE と呼ぶ)⁶⁾ などがある。FODA は、ドメインの中で卓越した側面、または、特色のある側面、かつユーザに明らかな側面である特徴を認識することを強調している。FODA のドメインモデリングプロセスは、特徴モデル、実体関連モデル、機能モデルの構築プロセスを持ち、ドメインモデリングの出発点は、

ドメインにおける特徴の認識である。FODA は、実体関連モデルを構築する点および、ドメインを複数の観点からモデル化する点で、我々の手法と類似している。しかし、我々のドメイン分析手法では、ドメインの特徴を、DOA による実体関連モデル、ユースケースに基づく、エンティティのインタラクション、状態遷移、インタラクションにおけるイベント/アクションのデータフローの観点で分析する。さらに、これらの観点での分析に加えて、分析結果を統合し、各観点で抽出したフローズスポットとホットスポットの候補からドメインのホットスポットとフローズスポットをフレームワークパターンとして抽出する観点および手順を含む。

ODM は、ドメイン分析において、ドメインを組織とプロジェクトの関わりの中でとらえることを特徴としている。企業の組織やプロジェクトの変更は、企業の持つ様々なアプリケーションシステムの構造にも影響を及ぼすことが考えられる。今後は、企業の組織や戦略の変化に柔軟に対処可能なドメインモデルを構築するためにも、ODM の示す視点を我々のドメイン分析・フレームワーク構築手法にも、取り入れることを検討する必要があると考えられる。

DARE は、ドメイン分析と再利用のための統合環境である。DARE のドメイン分析方法論では、トップダウンとボトムアップの2つの観点からドメイン分析を行うことを提案している。トップダウンの観点では、ドメインの専門家により、システムのアーキテクチャのあるべき姿を抽出する。ボトムアップの観点では、ドキュメント類で使用されている語の整理分類を、図書分類学のファセット分類を用いて行い、ドメインの特徴を洗い出すことを行う。DARE は、現行システムの分析とあるべき姿の分析の観点を備えている点で、我々の手法と類似している。我々のドメイン分析手法では、現行システムの分析のためには、DOA を採用している。また、システムのあるべき姿を明らかにするために、ユースケースに着目している。さらに、これらの分析結果を統合し、ドメインのホットスポットとフローズスポットを抽出するための手順を提供している。

7. おわりに

本稿では、フレームワーク構築に、DOA 手法およびユースケースの観点に基づくドメイン分析を行い、ドメインのフローズスポットおよびホットスポットを特定し、これらをフレームワークパターンとして表現する手法を提案した。また、我々の提案するフレー

ムワーク構築手法により作成される成果物の事例を示した。適用事例により、我々の提案するフレームワーク構築手法の有する以下の特徴を確認した。

- きめの細かいフレームワーク構築プロセスの提供すること、
- DOAにより作成したERDから、フローズンスポット/ホットスポットの候補を抽出すること、
- ユースケースの観点からエンティティ間のインタラクションなどを考えることによる、エンティティの役割の明確化すること、
- DOAとユースケースに基づいた、オブジェクト図のフローズンスポット/ホットスポットを特定すること、
- カスタマイズのための注力範囲を絞り込みやすくするために、フレームワークをフレームワークパターンへ分割すること。

しかし、実際の適用には、ドメインのフローズンスポット/ホットスポットの特定とフレームワークパターンによる表現において、本稿で提案したフレームワーク構築方法論に加えて、抽象化の度合いの検討、メソッドの最適配置、フローズンスポットとホットスポットの接続部分の抽出などに関する試行錯誤が必要である。また、フレームワークパターンを用いて新規のアプリケーションの開発を行う場合には、パターンを使用するユーザが、パターンの選択およびパターンのカスタマイズに関して、ドメインや実装の経験的知識を必要とすることが考えられる。

今後は、次の4項目について検討することが課題である。第一に、我々の手法により構築したフレームワークの再利用性、適用可能性を定量的に評価すること、特に、フレームワークをフレームワークパターンに分割することの有効性の評価を行い、それらの結果を反映させ、我々の手法を洗練することである。第二に、我々の手法において、プロセスの記述が不足していると考えられる、ユースケースの抽出/記述の手順および、試行錯誤や経験的知識を必要とする抽象化およびカスタマイズなどの手順について、それらのプロセスの記述の詳細化を行うことである。これらのプロセスの詳細化については、ユーザや技術者の経験や試行錯誤によって積み上げられた知識を、メタパターンとして抽出し、メタパターンを我々の手法に取り入れることを検討中である。第三に、我々のドメイン分析手法は、DOAとユースケースの観点に着目しているが、これらに加えて、企業の組織や戦略面の分析も、システムのあるべき姿の分析として取り入れることである。検討が必要な理由は、企業の組織やプロジェク

トの変更は、企業の持つ様々なアプリケーションシステムの構造にも影響を及ぼすことが考えられるためである。第四に、本稿では、フレームワークを構築する手法を提案したが、フレームワークを利用し、アプリケーションシステムを構築するためのユーザへの支援環境を構築することである。

謝辞 研究の機会を与えて下さった、(株)東芝研究開発センターシステム・ソフトウェア生産技術研究所有信睦弘所長、開発第四部松村一夫部長に感謝します。

参考文献

- 1) Gamma, E., Helm, R., Johnson, R. and Vlissides, J.: *Design Patterns: Elements of Reusable Object-oriented Software*, Addison-Wesley (1994). 本位田, 吉田 (監訳): オブジェクト指向における再利用のためのデザインパターン, ソフトバンク (1995).
- 2) Pree, W.: *Design Patterns for Object-Oriented Software Development*, Addison-Wesley (1994).
- 3) Sitaraman, M. (Ed.): *Proc. 4th International Conference on Software Reuse*, IEEE Computer Society Press, Orlando, FL (1996).
- 4) Kang, C., et al.: *Feature-oriented Domain Analysis Feasibility Study*: (CMU/SEI-90-TR-21). Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. (1990).
- 5) Simos, M.: Organization Domain Modeling (ODM): Formalizing the Core Domain Modeling Life Cycle, *Proc. ACM SIGSOFT, Symposium on Software Reusability*, Samadzadeh, M. and Zand, M. (Eds.), pp.196-205, ACM Press (1995).
- 6) Frakes, B., Prieto-Diaz, R.: Introduction to Domain Analysis and Domain Engineering, *Tutorial Text of 4th International Conference on Software Reuse*, IEEE Computer Society Press, Orlando, FL (1996).
- 7) Stephenson, M., Stropky, M.: Overview of the Domain Analysis and Design Process, *Tutorial Text of 4th International Conference on Software Reuse*, IEEE Computer Society Press, Orlando, FL (1996).
- 8) 名取, 加賀谷, 本位田: ドメイン分析に基づく仕様再利用手法, 情報処理学会論文誌, Vol.37, No.3, pp.393-408 (1996).
- 9) Natori, M., Kagaya, A. and Honiden, S.: Reuse of Design Processes Based on Domain Analysis, *Proc. 4th International Conference on Software Reuse*, IEEE Computer Society Press, Orlando, FL (1996).
- 10) 堀内: データ中心システム設計, オーム社

- (1988).
- 11) 椿：データ中心システム入門，オーム社 (1994).
 - 12) 林：～データ中心設計のための～ER モデル・システム分析/設計技法，SRC ソフトリサーチセンター (1993).
 - 13) 名取，加賀谷，本位田：DOA に基づくフレームワーク構築手法，情報処理学会，ソフトウェア開発のためのドメイン分析・モデリング技術シンポジウム論文集，Vol.96, No.5, pp.133-141 (1996).
 - 14) Jacobson, I., Ericsson, M., Jacobson, A.: *The Object Advantage, Business Process Reengineering with Object Technology*, ACM Press (1995). 本位田 (監訳)：ビジネスオブジェクト，ユースケースによる企業変革，トッパン (1996).
 - 15) Barker, R.: CASE*METHODSM: *Entity Relationship Modeling*, Addison-Wesley (1990).
 - 16) Codd, E.F.: A Relational Model of Data for Large Shared Data Bank, *CACM*, Vol.13, No.6 (1970).

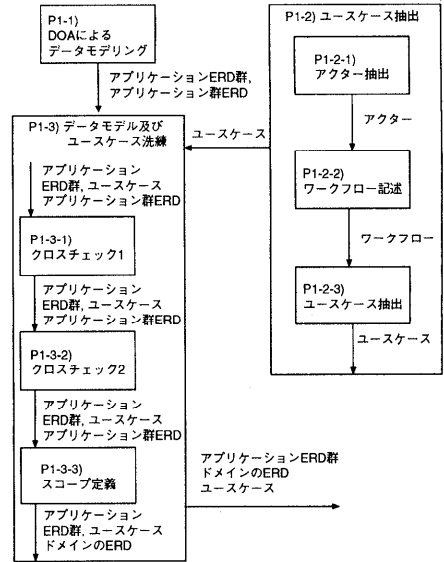


図 14 P1) スコープ決定のデータフロー
Fig. 14 P1) A dataflow diagram for scoping a domain.

付録 フレームワーク構築プロセスにおけるデータフロー

3章で示したドメイン分析のプロセスP1)~P4)のデータフロー図を示す。またプロセスP1)のうち、DOAに基づくデータモデリングプロセスの詳細を示す。データフロー図は、長方形がプロセスを表し、矢印がデータの流れを示す。矢印の近傍には、プロセス間を流れるデータを記述する。矢印の行き先に相当するプロセスがないものは、次のフェーズにデータが渡されることを表す。

以下、図14にプロセス“P1)スコープ決定”のデータフロー図を示す。図15にプロセス“P1-1)DOAによるデータモデリング”のデータフロー図を示す。図16にプロセス“P2)ドメイン特徴抽出”のデータフロー図を示す。図17にプロセス“P3)フレームワークパターン抽出その1”のデータフロー図を示す。図18にプロセス“フレームワークパターン抽出その2”のデータフロー図を示す。

図15に示すプロセスP1-1では、現行アプリケーションシステム群の画面・帳票類から各アプリケーションシステムのデータ構造を明確にする。また、これらのデータ構造を統合して、アプリケーションシステム群のデータモデルを作成する。本プロセスの詳細を以下に示す。

<手順概要>

- P1-1-1) 画面・帳票類の収集：現行システムの画面仕様、帳票類などのドキュメント類を収集する。
- P1-1-2) データ項目抽出：画面・帳票ごとに、出

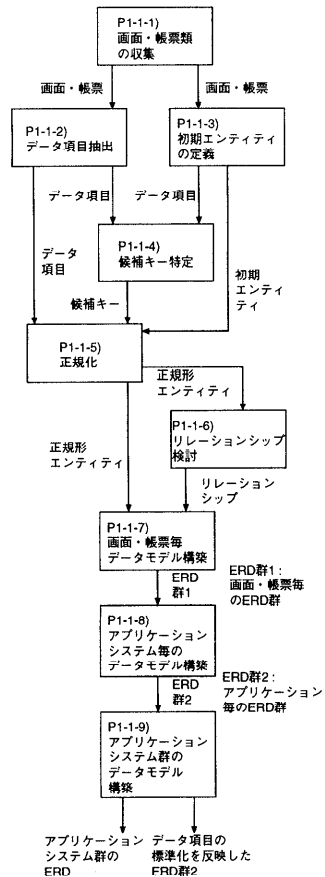


図 15 P1-1) DOA によるデータモデリングデータフロー
Fig. 15 P1-1) A dataflow diagram of data modeling processes based on DOA.

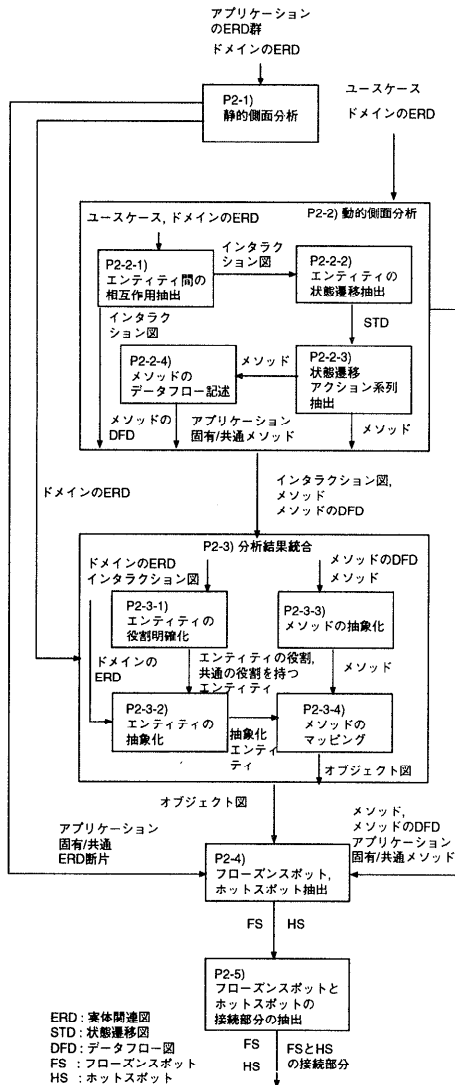


図 16 P2) ドメイン特徴抽出のデータフロー

Fig. 16 P2) A dataflow diagram for extracting domain features.

現するデータ項目を抽出する。

- P1-1-3) 初期エンティティの定義：一枚の画面・帳票をエンティティと見なし、エンティティの名称を定義する。
- P1-1-4) 候補キー特定：P1-1-2) で抽出された各画面帳票エンティティのデータ項目から、候補キーを洗い出す。各データ項目の特性（任意/必須）を検討する。
- P1-1-5) 正規化：P1-1-3), P1-1-4) で抽出したデータ項目の集合に対して、第1正規化～第3正規化¹⁶⁾までを行い、1つの画面・帳票を構成するエンティティと、各エンティティが持つデータ項目

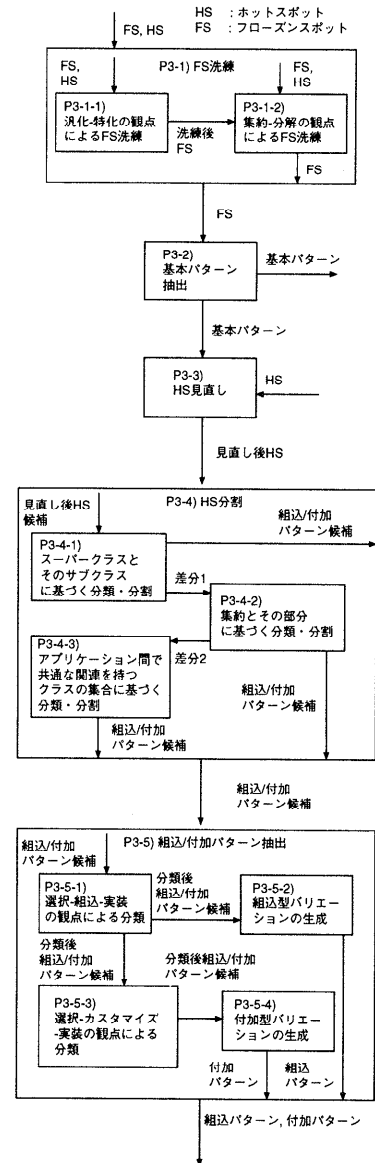


図 17 P3) フレームワークパターン抽出その1のデータフロー

Fig. 17 P3) A dataflow diagram for extracting framework patterns (1).

を決定する。また、データ項目の特性も決定する。

- P1-1-6) リレーションシップ検討：画面・帳票ごとに、P1-1-5) で抽出したエンティティ間の関連を洗い出す。
- P1-1-7) 画面帳票ごとのデータモデル構築：画面帳票ごとにエンティティおよびそれらの関連を実体関連図で表す。
- P1-1-8) アプリケーションシステムのデータモデル構築：画面帳票の実体関連図を、アプリケーションシステムごとに分類する。分類された実体関連

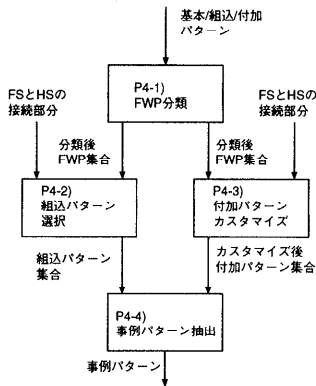


図 18 P4) フレームワークパターン抽出その 2 のデータフロー
Fig. 18 P4) A dataflow diagram for extracting framework patterns (2).

図群において、エンティティおよびデータ項目名の標準化を行い、共通部分を洗い出し、キー項目や属性に過不足がないかどうかを検討する。また、これらの実体関連図を統合し、第 1～第 3 正規化を行い、アプリケーションシステムの実体関連図を作成する。

- P1-1-9) アプリケーションシステム群のデータモデル構築：P1-1-8) で構築したアプリケーションシステムの実体関連図群から、エンティティおよびデータ項目名の標準化を行い、共通部分を洗い出し、キー項目や属性に過不足がないかどうかを検討する。これらの実体関連図を統合し、第 1～第 3 正規化を行い、アプリケーションシステム群の実体関連図を作成する。P1-1-8) で作成した各アプリケーションシステムの実体関連図に対して、データ項目の標準化の結果を反映させる。

(平成 8 年 9 月 24 日受付)

(平成 8 年 12 月 5 日採録)



名取 万里 (正会員)

1967 年生。1989 年早稲田大学理工学部数学科卒業。1991 年同大学院理工学研究科数学専攻修士課程修了。同年 (株) 東芝入社。現在、同社研究開発センターシステム・ソフトウェア生産技術研究所オブジェクト指向技術センターに所属。ソフトウェア生産技術の研究に従事。ドメイン分析技術およびソフトウェア再利用方法論に興味を持つ。人工知能学会会員。



加賀谷 聡 (正会員)

1960 年生。1983 年東北大学理学部物理学科卒業。1988 年同大学院理学研究科博士課程修了 (原子核理学専攻)。理学博士。東北大学理学部附属原子核理学研究施設勤務を経て、1989 年 (株) 東芝入社。現在、同社研究開発センターシステム・ソフトウェア生産技術研究所オブジェクト指向技術センターに所属。ソフトウェア生産技術の研究に従事。データ中心アプローチ (DOA) によるシステム開発法、事務処理分野を対象としたオブジェクト指向技術に興味を持つ。



本位田真一 (正会員)

1953 年生。1976 年早稲田大学理工学部電気工学科卒業。1978 年同大学院理工学研究科電気工学専攻修士課程修了。工学博士。同年 (株) 東芝入社。現在、同社研究開発センターシステム・ソフトウェア生産技術研究所に所属。1989 年より早稲田大学非常勤講師を兼任。1996 年より大阪大学大学院および筑波大学非常勤講師兼任。主としてエージェント指向技術、オブジェクト指向技術の研究に従事。1986 年度情報処理学会論文賞受賞。日本ソフトウェア科学会理事。著訳書 10 点。