

CDM とデザインパターンを用いた情報システム実装手法の提案

吉田 和正*1 上田 紳一朗*2 高橋 和也*2 金田 重郎*1*2

*1) 同志社大学大学院工学研究科, *2) 同志社大学工学部

情報システム開発分野では、業務担当者へのヒアリングを中心とした開発手法がしばしば適用されてきた。しかし、業務担当者ヒアリングのみでは、対象業務が抱える問題を根本的に解決するシステムの導出は困難である。そこで、あるべき業務の姿を求めるために特定非営利法人技術データ管理支援協会の概念データモデリングを採用し、これに、マイケル・ジャクソンのプロブレムフレームを用いて、実装モデルおよび機能パターンを与える実装手法を提案する。併せて、市民税業務処理へ適用しその妥当性を探る。

Proposal of Information System Implementation Approach using CDM and Design Pattern

Kazumasa Yoshida*1 Shin-ichiro Ueda*2 Kazuya Takahashi*2 Shigeo Kaneda*1*2

*1) Faculty of Engineering, Doshisha University,

*2) Graduate School of Engineering, Doshisha University

On-site hearing approach has been employed in the requirement analysis process of the information system development. However, it is very difficult for this conventional approach to develop the system which solves the essential problem of the target business. Thus, this paper proposes a new approach to provide implementation models such as UML and a necessary function pattern from operation analysis by using Conceptual Data Modeling (CDM), proposed by Manufacturing Architecture for Series Products (MASP), and Michael Jackson's Problem Frames. Also, this paper shows an example of application to the taxation business process in tax for citizens.

1. はじめに

情報システム開発分野では、従来、現場担当者からのヒアリングによって業務フローを分析し、帳票等のデータ項目からER図を作成して情報システムを構築する手法がしばしば用いられて来た。しかし近年、このアプローチには「開発されたシステムが改造に弱い」「業務全体での導入効果が乏しい」等の課題が指摘されている[1]。

上記の課題を解決するアプローチとして、概念データモデリング (Conceptual Data Modeling, 以下 CDM) が知られている。中でも、特定非営利法人技術データ管理支援協会 (MASP) [3]は対象世界の「もの」と、あるべき姿としての「こと」に注目した、独自のCDM手法を提案している[4][5][6]。これにより、業務全体のデータ整合性と、改造が容易な情報システムを構築できるとしている[5]。MASPのCDMについては、KDDI[2]、JFEスチール[7]等の成功例が報告されている。

CDMでは、対象ビジネスに関する価値観を捨象し、

現実世界をあるがままにデータとして写し取る。ただし、CDMで表現されているのは、概念的な上位ビジネス記述であり、そのままでは、情報システムの詳細仕様をそこから導くことはできない。すなわち、CDM・実装間にはギャップが存在する。

そこで本稿は、CDMのモデルで捉えた「もの」「こと」を、情報システムの実装へ繋ぐ手法を提案する。具体的には、ビジネスにおける情報システムに必要な要素の抽出方法と、情報システムの構築に必要な概念クラスや機能種別を導く方法について述べる。情報システムに関するモデリングにはUML[10]を、デザインパターンにはマイケル・ジャクソンのプロブレムフレーム[11]を採用する。

以下2章では既存情報システム開発手法の問題点について述べる。3章ではCDMとプロブレムフレームについて述べる。4章ではCDMとデザインパターンを用いた分析手法を提案する。5章では適用事例を示す。6章では本提案手法の効果について考察する。7章はまとめである。

2. 現状の課題

情報システム開発分野では、従来、情報システムの機能仕様を導くために、現場へのヒアリングや対象業務を分析した業務フローを作成した後、UMLのユースケース図や概念レベルのクラス図などの記法を用いて、実装に直接必要なER図やデータフローダイアグラムを導く手法が多用されてきた。

近年、このアプローチの課題として、「上流からの仕様変更に弱い」「現場の要求は満たすものの、設備やシステムに対する投資効果や、業務全体の効率化への寄与が希薄である」などが指摘されている。そして、このような効果に留まる原因として、ビジネス側からは「ビジネス側のニーズが開発者に正しく伝えられているかを確認できない」、開発者側からは「ビジネスやユーザのニーズが曖昧なままに開発が進み、ニーズが明確になる頃にはシステムの身動きが取れなくなっている」などの指摘がある。

上記の問題の開発者側からの解決案として、アジャイル開発プロセスやMVCフレームワークを通して、ユーザやビジネス側に早期のプロトタイプレビューをフィードバックし、双方のギャップを埋めようというアプローチが存在する。しかしながら、これらのプロセスの中で用いられるモデルは実装主体のレベルで記述されていることが多く、あくまで開発者が読むためのモデルが主流である。

一方でビジネス側からのアプローチとして、経済産業省を中心にEA（Enterprise Architecture）への関心が高まっている。EAは企業（Enterprise）の活動全体を様々な観点から記述・統合し、企業活動の全体像を明らかにすることで、それを支援する情報システムの明確化を目的とする。EAは図1に示す4つの階層から成る。

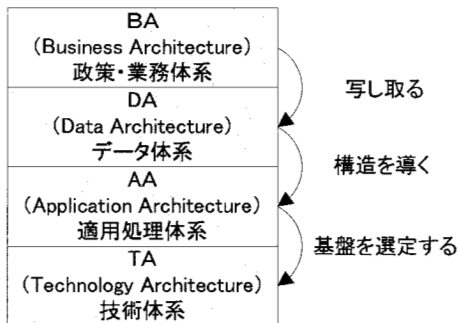


図1 EA (Enterprise Architecture) の構造

EAの視点から見ると、従来の開発手法には、ビジネスの構造から情報システムを適用するまでの工程に問題があることになる。例えば、現場ヒアリングからユースケース図を作成し、その中に現れる名詞と動詞からデータベースや画面の設計を行う方法がある。しかし名詞と動詞等の言語情報は必ずしも情報システム上に実装されるデータとは限らず、対象のビジネスの或る領域を表現しているに過ぎない可能性がある。

ヒアリングや業務分析から得た情報をそのまま実装するのではなく、この情報から対象ビジネスの本質を表現するデータ構造を吟味する工程が必要である。この工程を欠いたままに情報システム開発を行うと、ビジネスレベルの変更を情報システムへ反映することが難しくなる。EA視点からすると、ビジネスと実装との間をうまくつなぐモデル・分析手法が必要である。

3. アプローチ

前章の課題を解決するため、本稿では、技術データ管理支援協会(MASP)の概念データモデリング(CDM) [4][5]と、UMLのクラス図・コミュニケーション図、そして、マイケル・ジャクソンのプロブレムフレームの記法とを融合する。

3.1. 概念データモデリング

技術データ管理支援協会(MASP)が提案する概念データモデリングでは、世の中のビジネスを構成する「もの」と「こと」に着目して、対象業務をモデル化する[4][5]。具体的には、以下の図を作成する。この中で、モデル化において特に重要な役割を果たすのは、3番目までの3つの図である。

・ 静的モデル

業務に関係し、重要な「もの」とそれらの間の関連を表現する。

・ 動的モデル

静的モデルに現れた「もの」の中で、データ状態が変化する原因となる「こと」を順番に記述する。

・ 組織間連携モデル

静的・動的モデルに現れた「もの」「こと」を実際に存在する組織上に貼り付け、データの流れや責任関係の妥当性を検証する。

・ 機能モデル

上記以外の機能や動的モデルでは細かすぎる詳細な機能についてデータフローダイアグラムで記述する。

3.2. プロBLEMフレーム

プロBLEMフレームは、マイケル・ジャクソンにより提案された、問題解決デザインパターンの一つである[11]。デザインパターンは、通常、ソフトウェアの設計上の枠組みとして認識されているように思われる。これに対して、プロBLEMフレームはコンピュータとそれを取り巻く外の世界との間を問題領域とする。すなわち、情報システムで実現すべき外の世界からの機能要求と、そのために必要な情報システム内の機能仕様を橋渡しする。このため、CDM が持っている抽象性を機能仕様に変換するために効果的と期待される。プロBLEMフレームは以下に示す基本の5つのフレームとそれらの組み合わせである変種フレームで構成される。

・ 必要とされる振る舞いフレーム

外の世界や情報システム内の特定の条件に対して情報システムが制御される部分の問題を明確にする。

・ 命令された振る舞いフレーム

外の世界のオペレータ（情報システムの操作者）が発行するコマンドに対して情報システムが制御される部分の問題を明確にする。

・ 情報表示フレーム

外の世界の状態や振る舞いについて、特定の情報が継続的に必要な場合に、情報の入手方法と表示に必要な形式と適切な場所の問題を明確にする。

・ 仕掛品フレーム

コンピュータ上で処理可能なテキストやグラフィック等をユーザが作成・編集・検索するような特定の機能にはツールが必要となる。その機能が必要となる領域の問題を明確にする。

・ 変換フレーム

コンピュータが読み書きできる入力ファイルに対し、特定の形式で変換した出力が必要となる部分の問題を明確にする。

4. 提案手法

EA (Enterprise Architecture) に基づく情報システム設計では、対象ビジネスのモデルから情報システムを導く過程において、ビジネスの要求からデータ構造を導く必要がある。そこで、先ず、対象ビジネス領域の「もの」「こと」をCDMで取り出し、更に、「もの」から概念クラス図を作成し、最後に、「こと」からシーケンス図とプロBLEMフレームの機能パターンを導出する手法を提案する。

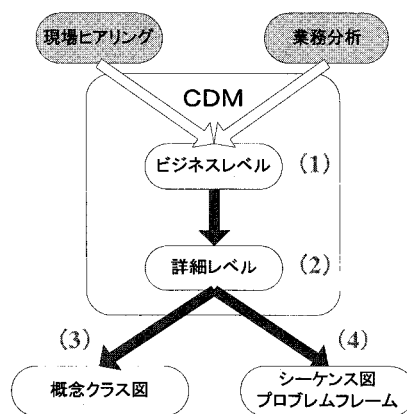


図2 提案手法のフロー

今回の提案手法は図2の中の(1)～(4)に示すような段階に分かれる。以下は各段階の概略について述べる。

- (1) MASPにより提案されたCDMの実施である。現場ヒアリングや業務分析で得たデータからステークホルダを導き、対象ビジネスにおける「事業領域と使命」を決定する。その中から現実社会に存在する「もの」から静的モデルを、「こと」から動的モデルを作成し、それぞれから組織間連携モデルを導く。組織間連携モデルと現実社会を照らし合わせ、「もの」「こと」の過不足に応じて静的・動的モデルを修正する。
- (2) ビジネス分析のためのCDMでは実装には概念的すぎる。このため、最初に実装を意識したCDMの詳細化が必要となる。具体的には、CDMの各図において、主に以下の点に留意して追記修正し、情報システムが扱うドメインを決定する(CDMが表現したビジネスのすべてが情報システム化されるわけではない)。* 静的モデルでは、対象業務の中で「もの」を扱う「人」を必要に応じて追加する。「もの」の属性と識別子は、実装する情報システムが管理するものに絞り込む。* 動的モデルでは、静的モデルに現れた「もの」のデータ状態が変化するような「こと」を詳細化し、「こと」の識別子と属性を絞り込む。その際、静的モデル上の「もの」によって「こと」が能動受動問わず発生するようにする。また、データ状態の変化はないが、業務上必要な「こと」についても、この詳細化プロセスで動的モ

デルに追記する。そして、最後に複数の動的モデルを並べ、「もの」－「こと」－「もの」が連鎖している構成とする。シーケンス図を生成する準備である。

* 組織間連携モデルでは、現実社会の組織上に「もの」「こと」を貼り付け、「もの」の発生源や「こと」による移動する様子を捉える。最終的に、理想として成し遂げたい業務を写し取った to be モデルになっているかに留意する。

(3) 詳細化された静的モデルと動的モデルに基づき、概念クラス図を作成する。具体的には、「もの」を概念レベルのクラスに、「もの」の属性をフィールドに変換する。ただし、「もの」の持つ「こと」をメソッドへ変換する際、静的モデル上の関連が「こと」を集約した表現となっている点に注目し、関連をメソッドへ変換しても良い。尚、『対象業務の中で「もの」を扱う主な「人』』の詳細動的モデルには、見かけ上データ状態が変化しない「こと」が記述されていることがある。この種の「こと」については、関係する「もの」のメソッドとする。「もの」および関連の変換については以下の点に注意する。

* 『対象業務の中で「もの」を扱う主な「人』』と他の「もの」との間の関連の場合は、他の「もの」にメソッドを加える。対象業務の中で「もの」を扱う主な「人」を関連と共に1つのクラスに変換すると、システムの柔軟性を損なう結果になることがあるためである[10]。この際、クラスにとって自然な使役関係を表現するメソッド名をつける。

* 『業務に関係する人という「もの」－対象業務の中で「もの」を扱う主な「人』』の関連の場合には、前者をクラスに置き換え、「取り込みチェックメソッド」を付与する。また間に中継データクラスを作り、「入力更新メソッド」を付与する。この関連は、データ連携に関するものなので、この他のメソッドは必要ない。このような関連では必ずデータの授受が必要で、データクラスを出現させる。

* 人という「もの」と所有の関連をもつ「もの」は、両者をクラスへ置き換えるが、その関連をメソッドへは変換せず、リンクのみを張る。

(4) 動的モデルからのシーケンス図の作成プロセスである。以下の点に留意してシーケンス図を作成する。

- ① CDM の動的モデルに表された、業務に必要な「こと」に着目し、それらの「こと」が起こりうる順序を検討したシーケンス図を、一つの「もの」ごとに作成する。その際、順序のあいまいさが残らないよう、綿密に検討を行う。
- ② シーケンス図に表れたそれぞれの「こと」を、プロブレムフレームの5つの基本フレームに当てはめる。その際、必要に応じて1つの「こと」を複数の基本フレームに分割する。
- ③ 作成した基本フレームそれぞれについて、各フレームに応じたフレーム考慮事項やフレームによらない考慮事項について検討し、その「こと」を実際に行うために運用上必要となる補助的な機能を導き出す。シーケンス図から得られた順序制約についても、併せて検討する。

考慮事項には5つの基本フレームに対応するもの、フレームの変種に対応するもの、そしてフレームによらず検討が必要なものがある。フレームによらず検討が必要なものには、超過考慮事項、初期化考慮事項、信頼性考慮事項、同一性考慮事項、完全性考慮事項がある。これらの項目について検討すると、業務に必要な「こと」を実際に運用していく上で、どのような機能が必要となるかを導き出すことができる。

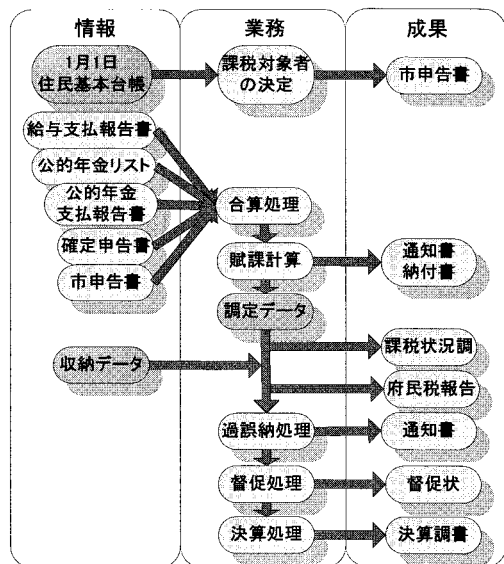


図3 年次税務処理の流れ

5. 税務処理システムへの適用

今回は、或る市の市民税に関する年次処理業務一連の業務を分析対象とした。ヒアリングにより得た年次処理の概要は図3に示すとおりである。今回の分析における業務の事業領域と使命は、市職員の方と協議した結果、「市民税の調査・課税・収納・滞納を遂行する」となり、この目的で CDM の各図の作成を行った。

5.1 ビジネスレベル～詳細レベル CDM

市職員と共に CDM モデリングを行い、45 個の「もの」から成る静的モデルを作成した。「もの」は「税務課」「企業」といった組織や、「会計表」「納税証明」などの帳票・証明、「課税データ」「収納データ」といったデータが主となった。関連は「送る」「印刷する」など作業内容レベルが主であった。動的モデルでは、静的モデルの関連（作業内容）に比して詳細な「1月1日住民の照合」「チェックリストの作成」などの作業手順レベルの「こと」が現れている。これをビジネスレベルでの CDM と扱った。

一方、業務で導入すべき情報システムを設計するには、業務上何がどのような目的で扱うか、を通して誰がどのデータをどの範囲までアクセスすることができるかを規定する必要がある。そこで、CDM の詳細化が必要である。静的モデルでは税務上の役割に応じて「税務課」に集約されていた役割と担当職員を「税務課職員」「会計課職員」など4つの担当課職員という「もの」に分けた。また、組織については組織名がついた「もの」を「税務署担当者」など組織内で業務に関係する担当者という「もの」に変更した。

静的モデルの関連については、「税務課職員は1月1日付けの住所を調査する」など、ものが主語・

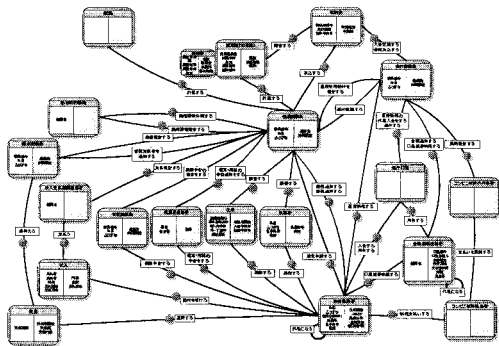


図4 詳細レベルの静的モデル

目的語となり、挟まれる関連は「もの」が業務上最終的に達成すべき「こと」を集約した動詞になるよう変更した。以上の変更から「もの」が20個となり、図4の静的モデルを得た。●は関連における多重側を表示しており、破線による囲いは「人」を表す「もの」である。

次に、動的モデルでは、細かい作業レベルの「こと」を逐一捉えるのではなく、対象とした静的モデル上の「もの」自身との関連で表現される業務上の目的に必要な「こと」を、細分化した。また、「こと」による「もの」自身が受ける変化と、変化によって他の業務上の「こと」が起きる条件を整理した。例えば、納税義務者からの異動申告は市民課職員が受け取り、税務課へ報告される。税務課への報告は、税務処理上、納税義務者となる人数の増減に関わるため、変化として捉えた。扱う納税義務者の人数変化は、後の合算処理を行う人数に影響を及ぼす。また、証明書の発行などは見かけ上、データ状態に変化のない「こと」である。この場合、証明書を要求する「こと」は発行の責務を発生させる、として扱うことで変化を捉えた。以上の変更から、静的モデル上の全ての「もの」について図5の動的モデルを得た。

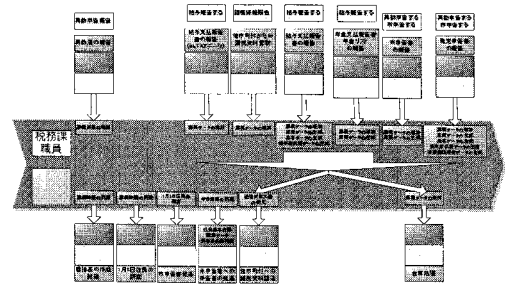


図5 詳細レベルの動的モデル（一部抜粋）

組織間連携モデルでは、金融機関など、現実社会の組織名と静的モデル上の「もの」の名前が同一になる場合が存在した。この場合、業務上必要な組織内での「もの」「こと」が捉えられなくなるため、内部の担当者を「もの」として静的モデルの修正することになる。また、情報システム内で扱う情報の種別として賦課は調定額と収納金の情報を統合した「もの」とするなど、業務上の作業と情報システムで行う作業の分別も行った。

5.2 詳細レベル CDM～概念クラス図

次に、詳細レベルの CDM から概念クラス図への変換を行った。静的モデルの「もの」をクラスへ変換する。今回の適用事例では、納税義務者の識別子として「氏名、生年月日、性別や現住所」があり、属性「配偶者等の血縁関係、課税区分」を持つため、これらをフィールドへ置き換えた。概念クラス図では、オブジェクトを一意に識別するフィールドを作る必要はないが、実装レベルのクラス図では識別子の代わりに「市民個人コード」を設ける必要がある。

メソッドの付与に関しては、「税務署担当者—納税義務者」および「収入支払機関担当者—納税義務者」の場合、中継データクラスとして「収入支払データ」と「確定申告データ」クラスを作成し、ともに「入力するメソッド」を与えた。プログラム上では、例えば確定申告データの取り込みを税務課が行う場合、確定申告データクラスが「create()」され、「入力 (the 納税義務者, the 確定申告データ) メソッド」を起動し、納税義務者クラスの「確定申告可能メソッド」で入力を許可し、確定申告クラスの「入力実施メソッド」で作業は終了する。

「税務署担当者—税務課職員」の「確定・相続の申告通知をする」といった関連は、特にメソッドへ変換しなかった。この関連を詳しく確認するために詳細動的モデルを確認すると、「通知およびデータの送信」という業務であることがわかる。これは納税義務者の情報を照会するために必要となる業務であり、システム上では、「納税義務者—税務署担当者」の「確定・相続の申告をする」という関連をメソッド化することで実現できる。

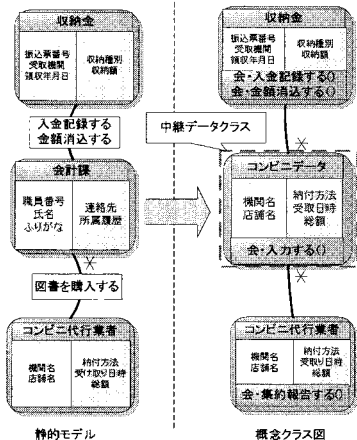


図6 「税務担当者—納税義務者」の変換

また「コンビニ収納代行業者—会計課職員」のような場合にも、「コンビニデータ」という中継データができる。静的モデルでは、「コンビニ収納担当者」と「コンビニ収納代行業者」を別の「もの」として扱っていたが、クラス変換の際には「コンビニ収納代行業者—コンビニ収納担当者」の関連である「支払いを集約する」は、コンビニ担当者側のシステムで行う業務であり、今回はこの2つを集約した。

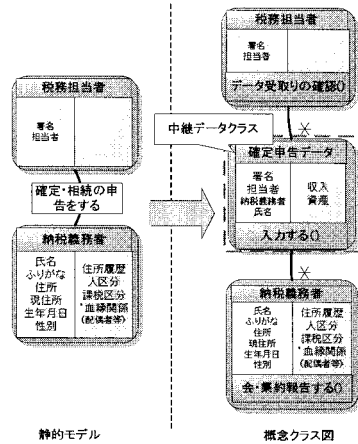


図7 「会計課職員—コンビニ収納代行業者」の変換

「納税義務者—資産」については、所有という関連で結ばれているため、概念クラスとしてメソッドを付与する必要はない。ただし、資産は納税義務者の属性でもあり、リンクを切ってはならない。また資産は差し押さえ対象となるので、「徴収課担当者—資産」の関連をもとに「差し押さえ調査をする()」というメソッドが付与される。

その他に、収納業務に関する「もの」として「金融機関担当者」や「銀行口座」が存在する。この2つは「所有」の関連であり、特別なメソッドは存在しない。しかし、これら2つの「もの」と「会計課職員」の間の関連は業務上重要なものが多く、変換作業は注意を要した。静的モデルでこの2つを分割したことによって、金融機関担当者が行う業務が明確になり、半自動化された口座経由の入金・還付作業をメソッドとして扱いやすくなることができた。本手法を用いてクラスへの変換を行う作業は、業務上のデータの流れを改めて確認し、加えてシーケンス図を利用してオブジェクトの起動、そして消滅までを大まかに考える作業でもある。作業を進める中

で、静的モデルの識別子は必ずしもキー属性として利用することはなく、新たに実装レベルでオブジェクトの識別コードを付与する場合があることが示唆された。また業務担当職員を静的モデルへ盛り込んだ詳細 CDM によって、システムの大まかな振る舞いを定義することは、概念クラス図への変換に大きく寄与したと考えられる。

5.3 詳細レベル CDM～シーケンス図・プロブレムフレーム

次に動的モデルから、シーケンス図などを導く必要がある。詳細レベルの CDM からシーケンス図・プロブレムフレームへの変換は、まず動的モデルに着目し、「こと」が実行される順序が明確になるよう注意しながら、シーケンス図を作成する。実際に作成した図の例が以下の図 8 である。

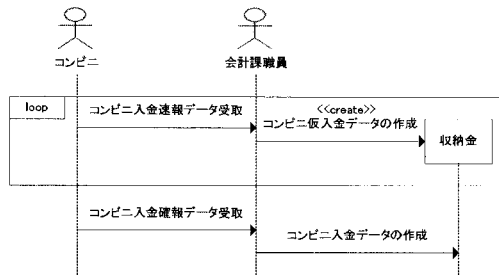


図 8 収納業務のシーケンス図 (一部)

図 8 からは、コンビニ入金速報データが送られてくるたびにコンビニ仮入金データを作成し、最後にそれらをまとめたコンビニ入金確認データを受け取って最終的なコンビニ入金データを作成するという業務の流れがわかる。この図をもとに検討を行うことで、例えば、コンビニ入金データの作成には、コンビニ入金確認データに対応するコンビニ仮入金データが存在している必要があることなどが導き出される。

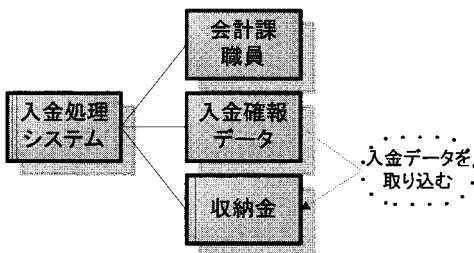


図 9 コンビニ入金確認データの取込のフレーム図

このように、シーケンス図に「こと」の順序関係を明確に表すことで、それぞれの「こと」がどのような前提条件を満たしていなければならないか、また順序関係があやふやな箇所は無いかを確認することができる。

次に、シーケンス図に表されたそれぞれの「こと」を、プロブレムフレームに当てはめる。フレーム図は図 9 のように表される。大抵の場合、一つの「こと」の実行のためには、インタフェース表示部、データ変換処理部といったように、複数の基本フレームへの分割が必要となる。「こと」をフレームに当てはめる際には、これ以上問題を分割する必要が無いかに十分に注意する。

最後に、それぞれのフレームに対し、プロブレムフレームの考慮事項をもとに運用上必要となる機能を検討していく。図 9 のコンビニ入金確認データの取込について実際に検討したところ、その一部として次のような事項について考える必要があることがわかった。

- ・ 入金確認データにデータ形式を満たさないデータが混ざっている可能性はあるか
- ・ 入金確認データに誤りがあった場合、どのように対処すべきか

またシーケンス図から得られた前提条件として、入金確認データと作成済みの仮入金データの対応が取れている必要があることがわかっている。この前提条件を満たすため、以下についても検討が必要である。

- ・ コンビニ入金確認データの個々のデータに対応する仮入金データが無い場合はあるか。また逆に、仮入金データに対応する確認データが無い場合はあるか。
- ・ 上記の対応関係を満たさなかった場合、支払い情報の消失等を防ぐためどのような対応を行う必要があるか。

以上のように税務の収納業務に対し検討を行った結果、シーケンス図とプロブレムフレーム収納業務の「こと」を実際に運用する際に必要となる機能を体系的に導出できることが確認できた。

6. 考察

前章の適用事例で、今回の提案手法では、税務処理業務における「もの」や「こと」から、業務上必要となる情報システム内のクラスや機能種別、付随データがある程度得られることが分かった。

CDMの各図を詳細レベルに変換する部分では、例えば税務上では税務課職員が1月1日の調査を必ず行い、課税条件として重要な要素として扱われるため、住所などは「もの」として静的モデル上に現れた。業務に関する制度や制約と、業務上必要な「もの」と「こと」の関連から分析することで、データが扱われる意図や業務上の役割を考慮し易くするための設計が行えたと考えられる。

詳細レベルの CDM から概念クラス図の変換においては、「もの」を構成する情報と業務上必要とされる振る舞いが「こと」と関連から捉えられている。静的モデルにおいて「もの」は関連を持つので、データとリレーションという概念と照らし合わせれば、ER図との対応を連想する。しかし、動的モデルで「もの」には変化を及ぼす「こと」が存在しており、静的モデルにおける関連は「こと」を集約した動詞で表現される。ER図における関連はエンティティ同士の集合としての論理性を表現するが、言語表現としての動詞はそれを含む主語の目的語に対する振る舞いを表現する。従って、ER図ではなくクラス図への変換が適していると考えられる。

詳細レベルの CDM からシーケンス図・プロブレムフレームの変換では、業務上の「こと」について情報システム内に必要な機能が体系的に捉えられている。これは、動的モデル上の「こと」が、シーケンス図において実際の業務で情報システムを運用する際の順序・選択・反復を整理され、「こと」毎に起き得る業務上の要求と情報システムに必要な機能との仕様に関する問題がプロブレムフレームに沿って考慮されたことによると考えられる。

詳細レベルの CDM 以降のステップに共通するのは、業務上必要な「もの」と「こと」を情報システムで扱うべきデータ表現として捉えられることである。しかし、情報システムに関する検索・照会・履歴の機能などは捉えきれない。これは、情報システム内で、各ステップから導かれたクラスや機能をコンピュータ上で制御するにはどのようなデータが必要か、は捉えきれないということである。この問いには例えば、動的モデルで現れた「こと」には、さらに細かい「こと」が達成には必要である、とした

動的モデルからの分析を CDM で行うなどのアプローチが必要である。その結果と、MVCフレームワークなど実装アーキテクチャのデザインパターンから、今回のステップで導かれた概念クラス図は制御上必要なデータを補って実装クラス図へ変換されると考えられる。

7. おわりに

今回の提案手法では、税務処理業務の内、業務を行う際に扱う「もの」が更新されるようなデータについて、実装すべき情報システムの概念クラスと機能種別については一定の情報を抽出することができた。しかし、完全な実装レベルの設計には、今回の結果で得たデータに加え、業務レベルに必要な「もの」「こと」以外の、情報システム内でデータを制御するための情報まで導かねばならない。

そのためには、詳細レベルの CDM で表現される業務上の「こと」の達成に必要な「こと」を導出するようなプロセスや、今回の結果に加え、MVCモデルのような実装アーキテクチャ上で考えた場合の概念クラス図やシーケンス図プロブレムフレームを通じた「もの」「こと」の位置付けを行う手法を考える必要がある。

参考文献

- [1] 経営情報学会 システム統合特設研究部会[編]、「成功に導くシステム統合の論点」日科技連合、2005
- [2] 前掲書、p121,「KDDIの事例-概念データモデルによるシステム統合-」
- [3] 特定非営利法人 技術データ管理支援協会 (MASP) Web サイトは以下の通り。
<http://www.masp-assoc.org/>
- [4] 手島歩三、「概念データモデル設計によるソフトウェアのダウンサイジング」、日本能率協会マネジメントセンター、1994
- [5] 手島歩三「ビジネス情報システム工学概説—概念データモデリングに基づく情報システム構築と運営—」技術データ管理支援協会 (MASP) ・内部資料 (非売品)、2006
- [6] 中村善太郎、「もの・こと分析で成功するシンプルな仕事の構想法」、日刊工業新聞社、2003
- [7] 杉原明、白崎俊行、森弘之「J-Smileを支えるITイノベーション (メソドロジ) —柔軟なシステム構築、短工期開発を実現する設計開発方法—」、JFE技報、No.14、pp.25-28、11月、2006
- [8] H・ウィリアム・デトマー著、内山春幸・中井洋子訳、「ゴールドラット博士の論理思考プロセス」同友館、2006
- [9] 吉澤憲治、星翔太、金田重郎「TOCとCDMを用いた業務分析手法の提案」、情報システムと社会環境研究会、2008
- [10] 児玉公信、「UMLモデリングの本質—良いモデルを作るための知識と実践—」、日経BP社、2004
- [11] Michael Jackson、「プロブレムフレーム—ソフトウェア開発問題の分析と構造化—」、翔泳社、2006