

概念データモデリング(CDM)と 責務駆動設計(RDD)に基づく 情報システム実装手法

岡田裕[†] 金田重郎[†]

オブジェクト指向は、情報システム設計のための強力なツールである。しかし、現実には、業務全体のデータ整合性と組織間の連携関係を見出し、かつ業務専門家からの処理要求をうまく設計に活かすことは、とりわけ初学者にとっては難しい。本稿では、MASP アソシエーションが提案している「概念データモデリング(CDM)」を「責務駆動設計(RDD)」と組み合わせることで分析・設計工程をより一貫したものとし、分析結果を設計工程にそのまま生かす手法を提案する。実システム(A自治体向けのCMS)に適用した結果、ビジネスモデルの本質を見直すことができ、それを生かした責務を持つオブジェクト設定が可能となった。結果的に、将来のビジネス機能の追加に伴う改造が容易なモデル実装が可能となったと考える。

Proposal of Information System Implementation Approach using Conceptual Data Modeling and Responsibility-Driven Design

Yu Okada[†] and Shigeo Kaneda[†]

Object-oriented is a powerful tool to design information systems. However, it is very difficult for the less-experienced software engineers to keep data consistency in the application, to find appropriate relationship among stakeholders, and to reflect demands of application domain experts to the design. To resolve the problem, this paper proposes a new approach that combines Conceptual Data Modeling and Responsibility-Driven Design into one consistent and systematic development method. As a result, the output of the Object-oriented analysis is directly reflected into the final implementation. The proposed method has been experimentally applied into a design process of a real content management system design process. The experimental evaluation shows that the proposed approach clarifies the essential business targets and the appropriate object responsibilities.

1. はじめに

オブジェクト指向は、情報システム開発の最も強力な手段の一つとなっており、様々な開発手法が提案され利用されている。しかし、初学者にとっては、オブジェクト指向は、分かりやすいものではない。ややもすると、現実社会の「もの」のアナロジーに囚われて、保守・拡張が困難な情報システムを設計してしまう恐れがある。

この問題を解決する方法として、Rebecca Wirfs-Brock の提唱する責務駆動設計(Responsibility Driven Design, 以下 RDD)が注目されている。RDD では「なぜオブジェクト指向なのか」「そもそもオブジェクトは何なのか」「機能中心やデータ中心とそもそも根本的に違う点は何なのか」という原理原則にこだわり、ユースケースを完全に網羅するだけのモデリングではなく、業務全体の「責務(responsibility)」を再度、設計工程でも考えるモデリングを行う。具体的には、ロールステレオタイプやCRCカードなどを使用して、責務に沿ったオブジェクト識別・オブジェクト定義を行い、それらの疑問を明快にするアプローチ方法である[1]。

しかし、RDD はあくまでもソフトウェアの設計手法であり、オブジェクト指向分析手法としては不十分であることは否めない。RDD はオブジェクト指向における設計工程に重きを置き、設計ストーリーやCRCカードを使用し、システムの責務について改めて言及する。つまり、システムのステークホルダを設定し、ビジネスの対象世界の構造と振る舞いを捉え、組織間連携の仕組みを理解し、ビジネス改革の可能性を言及するオブジェクト指向分析の工程に関しては、RDD が有効とは考えられない。

そこで本稿では、RDD を行う前に、技術データ管理支援協会(MASP) [2] が提案する概念データモデリング(Conceptual Data Modeling, 以下 CDM)を適用する。CDM では、要(かなめ)の「こと」「もの」に注目し分析することで、業務全体のデータ整合性と組織間の連携関係を見出し、改造容易な情報システムを構築できる[3]。MASP のCDM については、KDDI, JFE スチール等におけるモデリングに適用した成功例が報告されている[4]。このCDM をオブジェクト指向分析で用いることで、RDD では導出しにくいビジネスの要求の本質を見出すことができ、要の「こと」「もの」を網羅したRDD を実施できる。A自治体から委託されて開発を行っているCMSシステムにCDMとRDDを適用した結果、その親和性を確認できた。

以下第2章では本稿で用いるRDDとCDMについて簡単に説明し、第3章では提案手法について述べる。第4章では設計例を示す。第5章では、本手法の有効性に関する考察を述べ、最後に第6章では本稿に関するまとめと今後の課題を述べる。

[†] 同志社大学大学院工学研究科
Graduate School of Engineering, Doshisha University

2. RDD と CDM

2.1 RDD

RDD では「なぜオブジェクト指向なのか」「そもそもオブジェクトは何なのか」「機能中心やデータ中心とそもそも根本的に違う点は何なのか」という原理原則にこだわり、オブジェクト識別・オブジェクト定義の中心に「責務(responsibility)」を置くことで、それらの疑問を明快にする [1]。RDD を行う際に、具体的な方法として、「ロール、責務、コラボレーション」、「ロールステレオタイプ」という概念を考え、「CRC カード」を用いる。これらについては後述する。これらの概念や方法は、あくまでもソフトウェアの設計が目的である。即ち、システム全体の責務を考えることはあっても、システムの明確なステークホルダを設定し、ビジネスの対象世界の構造と振る舞いを捉え、組織間連携の仕組みを理解し、ビジネス改革の可能性を言及するといったオブジェクト指向分析工程は、RDD は持っていない。

2.1.1 ロール、責務、コラボレーション

RDD における「責務」とは、何かを知っている、あるいは何かをおこなうなど、オブジェクトが満たさなければならない責任である。「ロール」は関連する責務の集合として捉えた役割を表す。そして、「コラボレーション」とはオブジェクトもしくはロールの相互作用を示す。以上のことを踏まえ、責務駆動設計では、アプリケーションは以下の①～⑥の関係性を果たすように設計される。

- ① アプリケーションは属するドメインの大きな責務を果たす必要がある。
- ② 大きな責務から複数のオブジェクトを考え出し、それぞれに責務を割り当てる。
- ③ 各オブジェクトが果たすべき責務を新たに考える。
- ④ 責務の集合としてロールを設定し、オブジェクトのあるべき姿を再度考える。
- ⑤ ロールは責務を果たすために、他のオブジェクトとコラボレートを行う。コラボレートすることで、場合により設計モデルを発展させる形で③を再度行う。
- ⑥ これらのオブジェクトが集まり、協調することで最終的にアプリケーションが持つ、より大きな責務を果たす。

以上の①～⑥の関係を表した概念図を図1に示す。図1のように、責務駆動設計を行う上で、これらのロール、責務、コラボレーションという概念をそれぞれ設計作業で考え続けることで、アプリケーションの要求を満足する現実的なソフトウェアをどれだけ明確に作り上げるかという事に注目し続けることができる。

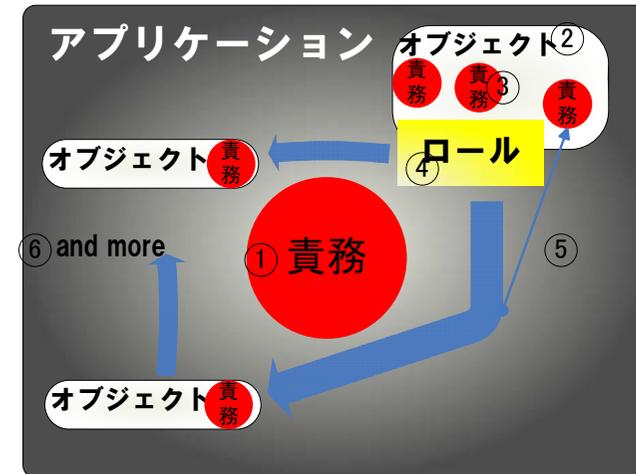


図1：責務、ロール、コラボレーションの関係概念図

2.1.2 ロールステレオタイプ

適切に定義されたオブジェクトは明確にロールをサポートする。意図的に思い切り単純化したロールステレオタイプという概念を用いることで、オブジェクト責務に注目しやすくなる。設計の目的は、一貫した使いやすいオブジェクトを作り上げることであり、オブジェクトにステレオタイプを与えることは有効である。つまり、オブジェクトの振る舞いの細かい違いを気にせず、抽象度を高くすることで思い切り性格付けをし、設計者はより簡単にオブジェクトの性質を考えられるようになる。以下のような六つのステレオタイプを責務駆動設計で用いている。

- 情報保持役(Information Holder)：情報を知り、情報を提供する。
- 構造化役(Structure)：オブジェクト間の関係と、その関係について情報を維持する。
- サービス提供役(Service Provider)：仕事を行うが、一般に演算サービスを提供する。
- 調整役(Coordinator)：イベントを他のオブジェクトに委譲する。
- 制御役(Controller)：判断を行い、④他のオブジェクトのアクションを指示する。
- インターフェース役(Interfacier)：システム内の異なる部分で情報と要求の変換をする。

2.1.3 CRC カード

CRC カードとは候補(Candidates)、責務(Responsibilities)、コラボレータ(Collaborators)をカードに記述し、ドメインよりのオブジェクトを責務とともに識別を行うものである。クラスのように関係や多重度を明示的に書かず、UML 表記の知識を持たない人であったとしても一緒にモデリングを行えるので、アプリケーションにおけるドメインの専門家も含めたモデリングが可能となる。従って、開発者だけの意見によらない様々な意見から直感的に理解できる責務を突き詰めることができる。

カードは手軽なものなのでうまく機能し、位置変えや修正が容易に可能である。また、実際にユースケースを想定して動かすので、全員のシステムに関する理解の向上にもつながり、設計のアイデアを洗練する手段としても非常に有効である。

これらの候補の責務はクラスの属性と操作に利用し、コラボレータはオブジェクト間のメッセージの送信先を表現する。コラボレートするカードは互いに近くに置き、CRC カードの最終的な配置はクラスモデルの基盤となる。

2.2 CDM

技術データ管理支援協会(MASP) が提案する CDM は、主要な「もの」、「こと」に着目して対象業務を分析し、本質的な業務プロセスをデータモデルとして写し取る手法である。具体的には、以下の四つのモデルを作成する。この中で、業務プロセスの本質を表現する上で特に重要な役割を果たすのは、上から三モデルであり、本稿でもこの三図を分析に利用する。

- ①実体関連図(静的モデル)：業務に関係する「もの」(エンティティ)とそれらの関係を記述
- ②実体状態変化過程図(動的モデル)：静的モデル中の「もの」それぞれに対して、「もの」が持つ属性値の状態変化を与える原因となる「こと」を、時間的順序を追いながら記述
- ③組織間連携図：上記の静的モデル・動的モデルを実際に存在する組織の上に貼り付けて、データの流れの妥当性を分析
- ④機能領域図・機能連鎖図(機能モデル)：上記以外の細かな機能についてデータフローダイアグラムを用いて記述

「もの」(エンティティ)については、業務を分析する際、粒度の決定が課題となる。MASP の CDM では、動的モデル上で状態変化が同一のものは、同一の「もの」として扱う。細かな状態の差は、識別子の値で区別する。結果的に、どの範囲の業務の「こと」を含むかが、「もの」の粒度に影響を与える。

「こと」については、業務が進捗した際に、「もの」の属性値を書き換える特定の

「こと」にまず着目する。例えば、「一覧表の作成」「領収書の作成」と言った、単なる情報提示の機能は分析に含めない方がよい。変更のみに着目することで、本質的部分のみに注目し、結果的にモデルを簡明化している。結果として「あるべき情報の流れと組織」を分析する手段である組織間連携図が簡明化される。そして、組織間連携図において、どこの組織がデータに責任を持つのかという視点を用いて、「as is」ではない、「to be」の業務の流れを見出しつつ、データの整合性を保証する。何れにせよ、この動的モデルも組織間連携図も、どの範囲の「こと」を分析に含めるかで大きく結果が変わってくる。

以上見てきたように、CDM は情報システムの仕様を記述することを目的としたものではないため、そのままでは情報システムの設計に使えない。そこで情報システムに実装すべきプロセスモデルの構築が必要である。

3. 提案手法

RDD ではオブジェクトの選定は、アプリケーションに関する設計ストーリーを書き、CRC モデリングにより行う。設計ストーリーはアプリケーションの重要な側面について記述することである。しかし、この重要な側面はシステムに対する分析の過程で導出することは非常に難しい。また、CRC カードモデリングで、アプリケーションとビジネスの両方に関する責務を混在して考えるというのも困難である。

つまり CDM でビジネスの要求の本質を導くことで、その本質を捉えたビジネスロジックに関する責務を導出することに専念する。その結果を RDD で利用し、再度設計ストーリーを書くことで、オブジェクト指向設計に活かす。

本稿で提案する CDM と RDD を併用したオブジェクト分析・設計手法のプロセスについて示し(図 2)、その具体的なステップを以下に示す。

[STEP 1]

MASP により提案された CDM を実施する。現場の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、それら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・動的モデルを修正し、業務全体のコンセプトとともに、情報システムが扱うドメインに対する属性、識別子および関連を導き出す。これはドメインオブジェクトの責務を洗い出すことに他ならない。

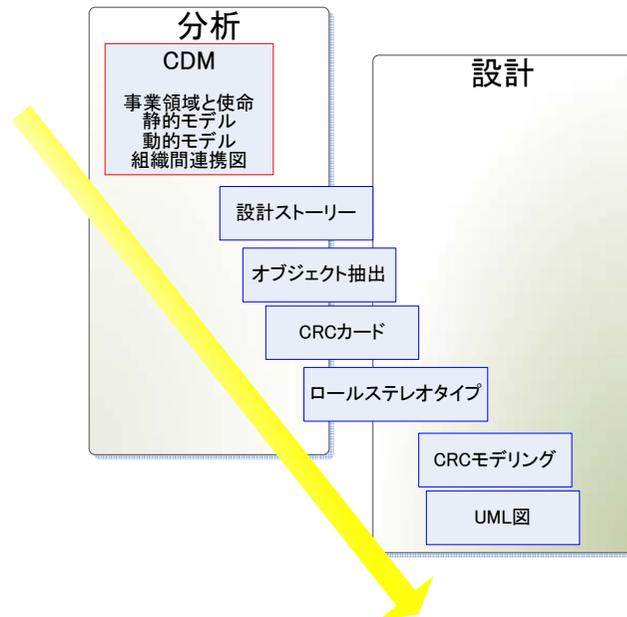


図 2：提案手法のプロセス

[STEP 2]

静的モデルと動的モデルに基づき、RDD を実施する。具体的にはユースケース、設計ストーリーを描く。設計ストーリーを書く際に、静的モデルの「もの」の説明をすることで、業務に関する設計ストーリー部分は補える。こうすることで、設計者はアイデア導出に集中することができる。また、設計者は静的モデルを読むことで業務の理解にもつながり、設計のアイデア導出のヒントになる。

設計ストーリーを描いた後、システムロジックを含めた重要なテーマをもう一度識別する。こうすることで分析・設計の両工程の繋ぎについて考える場を与え、CDM で導出したビジネス要求をどうシステムに生かすかという、システム全体の責務についてもう一度言及することができる。そのテーマをシステムの振る舞い、アーキテクチャ、性能、構造などの観点から対象領域の概念を表したオブジェクト候補の抽出を行う。また、オブジェクト候補の抽出の際には CDM であらわされた要の「もの」は責務を持ったドメインオブジェクトであるため、可能な限りオブジェクト候補にする。

[STEP 3]

STEP 2 で得られたオブジェクトから設計に役立てるオブジェクトへの転換を行う。まず、CRC カードを使用し、初期の設計のアイデアを作成し、実行するためのオブジェクトとドメインオブジェクトを関連付ける。また、CRC カードにはロールステレオタイプの記述を行う。同時に、ロールステレオタイプを用いてオブジェクトの責務に注目も行う。そして、広く動的な視点で不確かな部分も土台にした CRC モデルを作成する。CRC モデリングする際に以下の点に注意する。

- ※ CDM で導出された、オブジェクト候補としての「もの」を設計オブジェクトとして使う必要があるのなら、その「もの」をロールステレオタイプの情報保持役、構造化役、サービス提供役に割り当てる。これは、制御役と調整役がシステム制御に関すること、また、インターフェース役はシステムと外部との接続部であるため、CDM からは導出されないことに起因する。具体的には、静的モデルで表された要の「もの」であるエンティティの役割は情報保持役、それらの集合としての役割を持つものは構造化役、「こと」を実現するために目的として作られたエンティティは「サービス提供役」とする。
- ※ 責務を記入する際に、「もの」の属性と識別子はオブジェクトの持つ責務として記述する。また、「もの」の持つ「こと」をメソッド変換する際、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ことに注目し、関連も責務として考えられる。これらの責務を、箇条書きで CRC カードに記述する。敢えて箇条書きで書き込む理由として、後に考えられる責務との粒度を統一させるためである。

以上を注意しながら、多くの詳細な働きはシミュレーションを行い、抜けがないかを確認することで導出する。そして初期のクラスやシーケンス図を定義する。ここで不確かな部分が生じた場合、再度コラボレーションの妥当性を繰り返し検討する。

オブジェクトが一貫性を保つように相互作用しており、システムの自然な分割が保たれているのが示された場合に、コラボレーションは妥当なものとなり、STEP4 へ移行する。

[STEP 4]

設計をさらに予測可能な、一貫した、柔軟性のある、理解しやすいものにする作業を行う。アプリケーションに関する制御を割り当て、オブジェクト間の静的、動的な参照関係を決定し、詳細なクラス図とシーケンス図を作成する。

4. 適用事例

提案手法の有効性を検証するため、京都府内 A 自治体から依頼された観光支援システムについて本提案手法を試行した。A 自治体は日本の歴史上の非常に重要な部分を担ってきたと言える都市である。そこで、もっと身近に重要な歴史文化資産を社会資産として活用することを目的としてあるプロジェクトが立ち上げられた。具体的な手段としては、昨今インターネットがどの家庭においても普及しているといった背景を利用し、これらの情報を集約してより多くの人に発信するという手法をとった。

これまでに、観光イベントの一元管理システムと、史跡情報の html 自動生成システムを作成した。前者は運用段階前、後者は実際に運用してはいる。しかし、両方のシステムには課題を抱えており、アクセス数も多くない。そこで、これまでの問題を踏まえつつ、二つのシステムを統合し、A 自治体の史跡とイベント両方の観光資源を使った、A 自治体を効果的に PR することができるシステムを作りたいという要望が新たに浮上した。このような経緯で作成するに至った観光支援システムが評価対象の CMS(Content Management System)システムである。

開発には動的な Web アプリケーション開発に適している Ruby On Rails を用いて開発を行った。このシステムの特徴として、各コンテンツは相互作用し、複雑である。扱うものが観光資源である以上、コンテンツは日常的に変更される。観光情報について、必要項目を入力するだけで同一フォーマットの HTML ファイルを自動的に生成する Contents Management System (CMS) とする必要があり、管理部分も含めて、オブジェクトは複雑なコラボレーションを行うと予想される。

また、本プロジェクトは地域連携を図っており、今後コンテンツが増えることも期待される。また、今度、異なる要求が発生する可能性もあり、システムの保守・拡張は必然である。従って、拡張性の高いシステム基盤を設計するためにも、責務に則った設計が必要となり、RDD が役立つと予想される。

しかし、A 自治体を効果的に PR することができるシステムを作りたいという要望はあるものの、Web サイトの具体的な構成など、効果的なページ構成案や、システムに対するニーズの分析は明確には未着手であった。従って RDD を行う前に、分析工程として CDM を適用することが必要と考えた。

4.1 RDD のみによるモデリング

当初、CDM は利用せず、自治体の観光振興関係の業務担当者や、コンテンツを提供する歴史学の専門家などからのヒアリングにより業務分析を行った。業務分析結果から、以下のコンテンツを設定した。

- 時代旅行(メインコンテンツ)
 - 各時代の絵図と現代地図の対応
 - A 自治体の昔を知ることができる
 - 絵図上の史跡情報を閲覧可能
- A 自治体地図
 - A 自治体の観光資源を Googlemaps で表示
 - 各コンテンツから詳細を閲覧可能
- 散歩道
 - 歴史に造詣の深い専門家の進める A 自治体の歩き方の提案
 - 各散歩道における近隣のイベントの表示

上記の分析結果を受けて、当初、作成したトップ画面イメージ(図 3)を以下に示す。トップ画面イメージは図 3 のように、各コンテンツともに並列に上記タブの中に格納された。京都と時代旅行という事項を前面に PR するトップデザインといえる。その後、ユースケースを記述し、RDD を行った。最終的に以下のクラス図(図 4)が得られた。



図 3 : RDD のみによるトップ画面イメージ

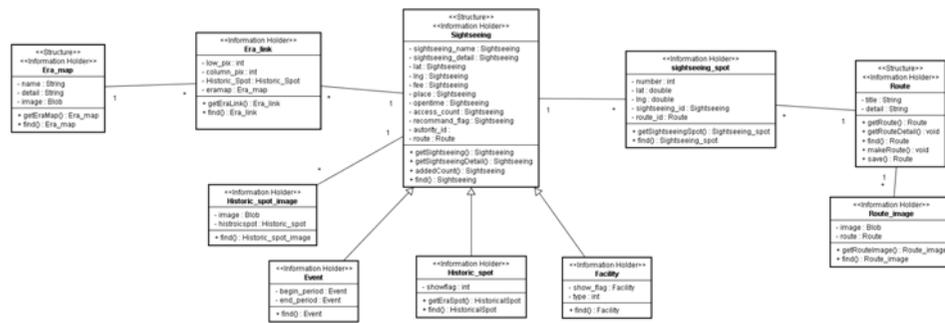


図 4 : RDD のみによるクラス図

図 4 に示すように、クラスの数は 11 個、属性の数は 29 個、操作の数は 23 個という結果となった。モデル層が数多く属性を持ち、操作を行うという結果が得られ、RDD を行った結果としては十分なものができたと考える。しかし、ビジネスロジックを満たしたものができたかということは、ヒアリングのみであるため、分析事項の設定の妥当性に不安が残る。

4.2 提案手法によるモデリング

ビジネスモデルの本質を改めて見直すため、CDM による業務分析を改めて行った。まず、自治体職員と歴史学専門家と協議し、前提事項の確認を行った。その結果、事業領域と使命は「様々な情報を PR する事により自治体を活性化する」となった。この事業領域と使命に基づいて分析し、各モデルを作成した後、最終的に以下の静的モデル(図 5)を作成した。

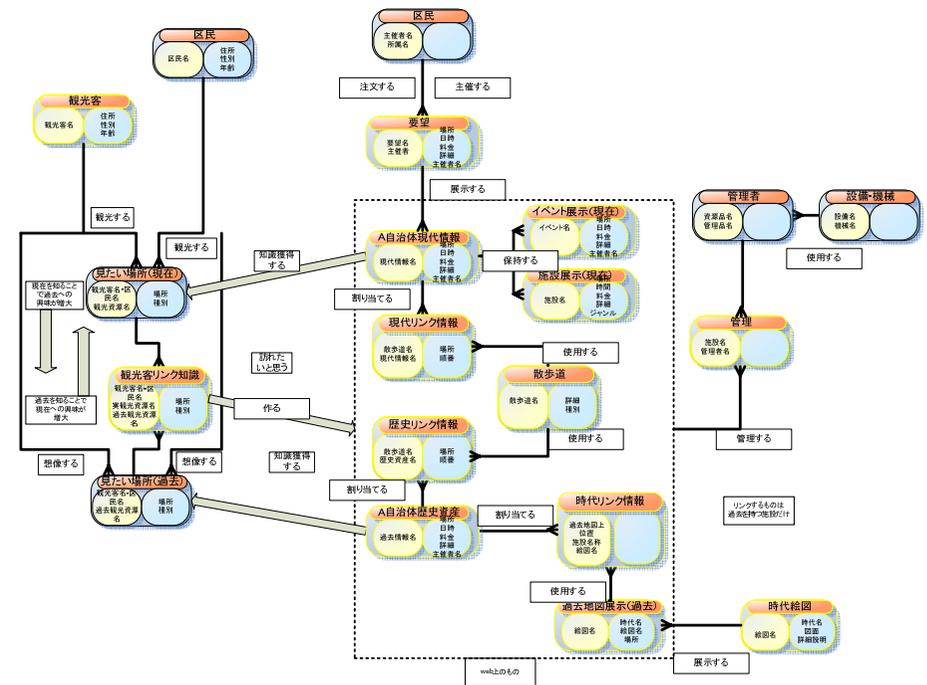


図 5 : 静的モデル

このモデルから、ビジネスの対象世界の構造を捉えたところ、「観光客や住民の方に、A 自治体における過去と今の情報とを関連付けた知識を獲得して頂くことで、A 自治体を PR する」というコンセプトが導きだされた。また、A 自治体の観光支援 Web システムとして表現する上で、ビジネスの観点を以下のように整理できた。

- Web システムにおける情報の流れは、過去と今の情報をリンクさせるものが必要であること
- ユーザごとのリンク知識を使って、新たにユーザに提示できる機能を追加できる可能性があること
- ユーザ自身が新たに散歩してみたい道を作るといったような、サイトによって得た知識を「表現する場」を与えることも必要なのではないかということ^a

^a 実際には悪意のある書き込みが自治体のサイトに行われた場合の問題などがあり、実現性については、別途検討が必要である。

- ・ 現代のイベントや施設は主催者がいるはずで、区民が主催する旨を伝える場も設ける必要があるということ

また、このビジネス目的を明確に意識して、図3のトップ画面の設計を再度、やり直した。新たに作成したトップ画面イメージ(図6)を以下に示す。

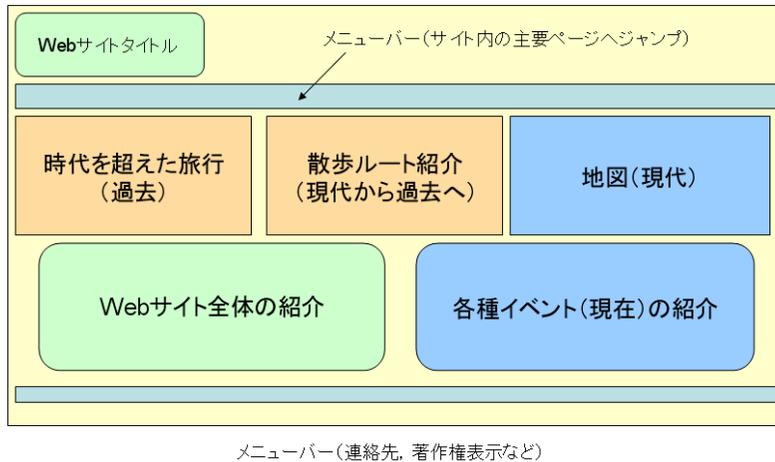


図6: CDMにより得たトップイメージ図

トップ画面イメージは図6のように、各コンテンツは(図3に比べて)大きな「コンセプト」として表示された。図3では、京都のA自治区における全体的な様を表現していたが、図6のような配置は業務の本質を新たに整理し、その結果を表現するものとして必然な結果である。

次に、上記のアプローチを生かした設計ストーリーを作成し、オブジェクト候補を抽出した。オブジェクト候補には図5の中に現れる主要な「もの」を「情報保持役」「構造化役」、主要な「こと」ならば「サービス提供役」としてロールステレオタイプに割り当てて、CRCカードに書き込む。またそれらの属性、識別子および関連を箇条書きで責務であるとして、カードに記述する。ロールステレオタイプでオブジェクトに性格づけを行い、CRCモデリングでコラボレーションを妥当なものとし、作成された詳細なクラス図(図7)を以下に示す。

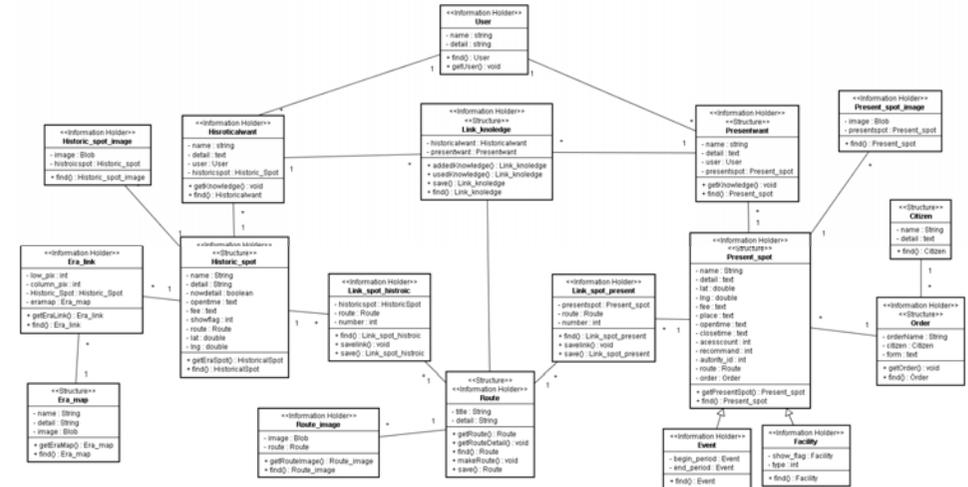


図7: CDMを行ったクラス図

図7に示すように、クラスの数、属性の数、操作の数は64個という結果となった。属性の数、操作の数は同じクラスにおいても、RDDのみによるモデリングよりも多少増えている。クラスに至っては新しく7つのクラスが導出された。

まず、「Historicspot」「Presentspot」というクラスについて言及する。この二つのクラスは、データ項目が似ているという観点からRDDでは図4のように「Sightseeing」という親クラスを設定し、一緒に分類していた。しかし、これは歴史コンテンツと現代コンテンツという明らかにビジネスターゲットが違うものである。管理の面から見ても、歴史コンテンツと現代コンテンツでは明らかに視点が違う。つまり、現代コンテンツに限っては、イベントや施設や店舗など載せるにも関わらず、RDD単体ではその最たるターゲットである「自治体住民」に対するオブジェクトが出てきていない。CDMの結果、図7のように「Order」「Citizen」クラスは新たに導出するに至ったのと当然の結果である。

また、「link_knowledge」「Historicknow」「Presentknow」という概念的なクラスも導出された。このクラスは実際にキャッシュを用いてユーザ判別をし、ユーザのリンク知識をサイト内で表示させるもの、ユーザに得た知識をフィードバックさせるものとしての機能を実現する可能性を秘めたクラスである。これは、得た知識を表現できるものとして、実際に散歩道をユーザ自身で作る事も可能である。

5. 考察

前章の適用事例から、RDD のみによるモデリングと、本手法とでは大いに異なる結果が得られた。この違いは、CDM でオブジェクト指向分析を行った結果が影響を与えていると考えられる。

図3と図6を比較すると、トップページサイトのコンセプトがコンテンツになっていた。これはこのシステムの明確な目的というのが一般的で抽象的なもの、「A 自治体を PR するもの」としてだけであったからである。CDM を適用することで、閲覧する可能性のあるユーザを明確に意識できるようになり、このシステム唯一の魅力を遺憾なく表現するトップデザインになった。また、ページフローについては、今回の評価対象は、比較的小規模の CMS であるが、CDM は、あるべきサイトの姿を導出する手段となることが十分に表現された結果である。

RDD は、MVC モデルにおいては、「C」に過度にビジネスロジックを集中させることを防ぎ、「M」に多くの責務(クラスの中における言葉で表すならば、属性とメソッドのこと)を割り当てる手法である。しかし、RDD は「当該情報システムのユーザに何を理解して欲しいか」、「ビジネスにおける将来性」について深く言及する場を持たない。今回の評価対象の Web サイトのページ構造についても、RDD では単純に独立したコンテンツを作るということしか気づけなかった。ページ構造は、CMS における VIEW としての役割に留まらず、ビジネスの流れを最大限に表現するものである。しかし、実際には現在と過去に対象性を持つ「知識」をユーザに最大限に提供したいはずであるものの、RDD 単独では、各コンテンツが相互作用するようなページ構造になっていなかった。しかし、CDM をすることで、各コンテンツが相互作用するようなページ構造となり、図6のような適切なコンテンツ配置になりえた。

ドメイン層の設計においては、ステークホルダ内における全体の業務の流れを意識し、ビジネスにおける本質的な「もの」「こと」を明らかにし、責務を取り出した。そして、設計ストーリーを描き、設計よりの視点を加え、CDM の要の「もの」に枝葉(システムにおける責務、責務の抜け)をつけるのが、本手法における CRC モデリングの最も重要な役割といえる。

結果、RDD にとっては弱点である「ビジネスの視点」を導入した RDD を実現し、図7のように RDD では導出しなかったビジネスロジックを持ったクラス、属性と操作が新たに設定された。そして、分析工程と設計工程を繋げる手法として、CDM と RDD のお互いの役割を崩すことなく、かつ親和性の高い手法であることが確認できた。

6. おわりに

オブジェクト指向は、今日、情報システム開発の最も強力な手段となっており、現場でも、様々な開発手法が使われている。しかし、初学者にとっては、オブジェクト指向は、決して分かりやすいものではない。現実社会の「もの」のアナロジーに囚われて、保守・拡張が困難な情報システムを設計してしまう恐れがある。そこで、現在 Rebecca Wirfs-Brock の提唱する RDD(責務駆動設計)というモデリング手法が注目を浴びている。しかし、RDD はオブジェクト指向における設計工程に重きを置くものである。つまり、ビジネス改革の可能性を言及するオブジェクト指向分析工程は、RDD 自体には従来と同様に的確に定める機能を持っていないことになる。

そこで、特定非営利法人技術データ管理支援協会(MASP)の提唱する、CDM(概念データモデリング)の分析結果を、RDD へと繋げる手法を提案した。そして、A 自治体の観光支援システムへの適用を行った。本手法では CDM で種々のあるべき情報の流れ、つまり業務全体の「to be」を発見するに至った。そして、導出された要の「もの」「こと」を RDD の工程につなぐ手法を提案した。結果、分析工程で導出したものを生かした RDD を行うができ、情報システムのモデリングにおける分析工程と設計工程のギャップを繋ぐ有効なアプローチ方法であることを確認できた。本稿では、まだ、ワンポイントの評価である。本稿では比較的小さい CMS についてモデリングを行ったが、CDM はビジネス対象をある一定の粒度で見ってしまうものなので、より規模の大きいシステムについてはどのような結果になるのか、今後の課題として考えていく必要がある。

謝辞 本稿で CDM と RDD による業務分析にご協力いただいた自治体職員、関係者各位に、深謝致します。

参考文献

- 1) レベッカ・ワーフスブラック, アラン・マクキーン 著, 藤井拓 監訳, 辻博靖, 井藤晶子, 山口雅之, 林 直樹 翻訳: オブジェクトデザイン, 翔泳社(2007)
- 2) 特定非営利法人技術データ管理支援協会(MASP)
<http://www.masp-assoc.org/>
- 3) 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)
- 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) デイブ・トーマス, デイビッド・ハインマイアー著, 前田修吾 監訳: Rails によるアジャイル WEB アプリケーション開発, オーム社(2007)
- 6) 児玉公信 著: UML モデリングの本質, 日経 BP 社(2006)