

# 共通業務を考慮したドメインモデルに基づくプロトタイプング

川 端 亮<sup>†</sup> 伊 藤 潔<sup>†</sup> 熊 谷 敏<sup>††</sup>

ドメイン分析は、対象領域(ドメイン)に固有な、分析方法、用語集、仕様書、ライブラリ、ソースコードなどをドメインモデルとして獲得、再利用することで効率的なシステム開発を行う手法である。ドメインモデルには、帳票処理業務、割当て業務など、業種に依存しない共通業務が存在する。共通業務ごとに、プロトタイプを開発するプロセスをドメインモデルとして獲得することで、他のドメインへも共通業務のプロトタイプングサイクルを適用し、再利用できる。本稿では、共通業務の観点からのドメイン分析の例として、教務ドメインを協調業務、帳票処理業務の観点から分析し、教務情報システムを構成した。また、協調業務、帳票処理業務のプロトタイプングサイクルを実装したプロトタイプングツールを試作した。これらのツールを用いることで、対象ドメインについて、協調業務の観点から分析し、業務の性能評価ができる、また、帳票処理業務の観点から分析し、結果をデータベース言語形式の出力で得ることができるので、この出力を用いてデータベース上で稼働させることで期待どおりに動作するか評価できる。

## Prototyping Based on Domain Model Considering Generic Task

RYO KAWABATA,<sup>†</sup> KIYOSHI ITOH<sup>†</sup> and SATOSHI KUMAGAI<sup>††</sup>

Domain analysis and modeling (DAM) enables efficient system developments. Domain model which is acquired through DAM process takes various forms such as analysis method, glossary, specifications, library and source codes. Since domain model makes the best use of the domain characteristics for the reuse within that domain, it is highly dependent on the domain and needs extra efforts for reuse to another domain. The authors have recognized common facets acknowledged across several domains and called them generic task. Generic task includes trading task, allocation task, monitoring task, and collaboration task, etc. In order to validate the reusability of generic task beyond a particular domain, this paper deals with prototyping cycle of generic tasks as a domain model to be reused. Prototyping cycle is a method which allows to analyze, design, and evaluate a system in an interactive manner while executing a prototype of the system. Prototyping cycle of the generic task can be reused to different domains. We analyze the domain for academic affairs in terms of collaboration task and trading task and develop a prototyping tool for the domain. The tool integrates the two prototyping cycles of the generic tasks. The tool allows to analyze the system in terms of collaboration task, while simulating and evaluating it in terms of efficiency of collaboration. The output of the analysis in terms of the trading task is obtained in the form of SQL description for the evaluation of database behaviors.

### 1. はじめに

システム開発は、多くの場合、過去に作られた類似したシステムの仕様書、設計書、ソースコードなどを再利用して、異なる部分を開発対象システム用に修正することで開発にかかる労力を軽減する。システム開発が行われる対象領域(ドメイン)は、在庫管理、金融、プラント制御、列車や飛行機などの座席予約、運行管理など様々である。各々の領域には、固有の構

造、決まりごとがあり、システム開発は、これらをつまえて行うことが重要である。ドメイン分析・モデリング<sup>1),2)</sup>は、ドメイン固有のこれらの性質を分析・モデル化しておくことで、新規システムの開発を効率的に行うことを目的とした手法である。ドメイン分析・モデリングでは、要求分析をすることで得られた仕様書や、ダイアグラム、プログラムのソースコード、アルゴリズム、データベーススキーマ、パラメータチューニング、開発に用いる手法などをモデル化し、再利用する。

筆者らは、これまで、運輸ドメイン<sup>3)</sup>、診察ドメイン、プラントドメイン、OLTPリアルタイムシステムドメイン、図書館ドメイン、販売ドメイン、レンタル

<sup>†</sup> 上智大学理工学部

Faculty of Science and Technology, Sophia University

<sup>††</sup> 株式会社山武

Yamatate Co., Ltd.

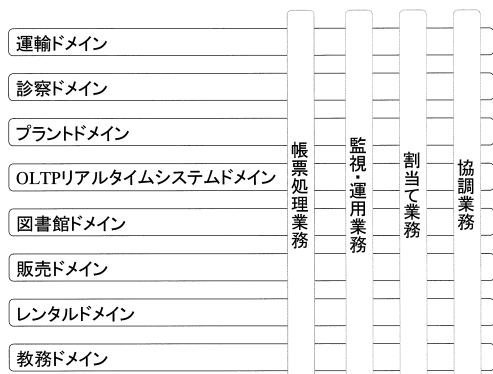


図1 ドメインと共通業務の関係

Fig. 1 Relationships between domains and generic tasks.

ドメイン，教務ドメインなどの主要なドメインの分析を行い，ドメインモデルを明らかにした．これらのドメインモデルには，特定の業種や業態に依存しない共通の要素が存在している．これらの共通の要素を共通業務 (generic task)<sup>4)~6)</sup>と呼ぶ．共通業務には，帳票処理業務，監視・運用業務，割当て業務，協調業務などがある．共通業務の観点でとらえると，運輸，診察，プラントなど，様々な業種のドメインは，いくつかの共通業務の組合せで表現できる (図1)．分析をするとき，この共通業務を1つのドメインととらえることで，より再利用性の高いモデルを得ることができる．

本研究では，共通業務を1つのドメインととらえ，共通業務のプロotypingサイクルに着目した．プロotypingサイクルとは，プロotypingの記述，プロotypingの稼働，プロotypingの評価のサイクルを回りながら開発を行っていく方法である<sup>7)</sup>．業種のドメインを共通業務の観点から分析し，共通業務ごとのプロotypingサイクルをドメインモデルとして定義すれば，その共通業務を要素として含む他の業種のドメインに対しても，共通業務のドメインモデルが再利用可能である．

本稿では，共通業務の例として，協調業務と帳票処理業務を取り上げ，それぞれのドメインモデルを実装した共通業務のプロotypingツールを試作し，協調業務と帳票処理業務の両側面を持つドメインの例として，教務ドメインに適用し，教務情報システムを構成した．

## 2. 共通業務の分析

ドメインの分析を行うとき，我々は，いろいろな視点から分析を行う．たとえば，「そのドメインを構成しているものは何であるか？ 組織構造はどのように

なっているか？ どのような制約条件があるか？ どのような業務がどのようなプロセスで行われているか？」などについて分析する．このとき，運輸ドメイン，診察ドメインなど，業種ごとにドメインをとらえ分析を行う．これらの業種のドメインを見ると，異なる業種のドメインでも，類似した業務が行われていることが多い．たとえば，コンビニエンスストアなどの販売ドメインでは，伝票や台帳を用いて，商品の受発注，在庫管理を行う業務があるが，プラントドメインでも，スペアの配管やポンプなどの在庫を調べ，不足していたら注文するという同種の業務が行われている．また，販売ドメインでは，店員がどの日のどの時間帯の販売を担当するかというスケジュール割当ての業務があるが，プラントドメインにおいても，操業や点検のスケジュール管理と，それらを行う要員の割当てという同種の業務が行われる．要員は作業を行う人のことである．

筆者らは，これら同種の業務を共通業務 (generic task)と呼ぶ．この共通業務を1つのドメインととらえ，分析することで，より再利用性の高いモデルを得ることができる．共通業務を考慮したドメイン分析では，共通業務ごとに，構成要素，分析手順などを明らかにし，モデル化しておけば，異なる業種のドメインでも，そのモデルを利用できる．たとえば，帳票処理業務では，伝票，台帳，要員が主要な構成要素である．分析するときには，まず，これらの構成要素を認識し，次に「要員が伝票に記入する」などの，構成要素間の関連を明らかにする．そして，得られた関連をもとに，「商品台帳の在庫数量から発注伝票の発注数量を引く」などのトランザクションを明らかにする．この分析手順は，プラントドメインや，販売ドメインなどにも利用可能で，それぞれの業種のドメインで，どれが伝票，台帳，要員であるかを考え，分析を進めていけばよい．現在，筆者らは，共通業務として帳票処理業務，協調業務，割当て業務，監視・運用業務を認識している．

共通業務は，それぞれの業種のドメインにおいて，要素として占める割合が異なる．たとえば，販売ドメインでは，帳票処理業務が中心に行われ，要員配置などの割当て業務は少しいである．また，鉄道ドメインでは，列車のダイヤ作成という割当て業務が大きな割合を占めている．モデルを再利用するとき，ドメインを業種の観点からとらえて，似ている業種のドメインモデルを再利用することも考えられるが，それぞれの業種のドメインがどれだけ似ているのか，また，用語の意味や，ビジネス慣習が，ドメインどうしてどれだけ違うのかを判断するのが難しい．モデルを共通業務ご

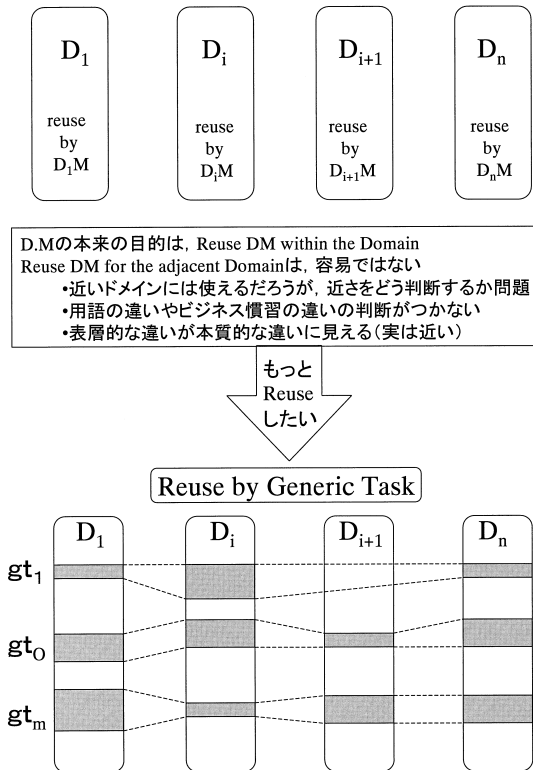


図2 共通業務を活用した再利用

Fig. 2 Reuse from the viewpoint of generic task.

とに明らかにし、それらを組み合わせていくほうが、より正確なモデルを得ることができ、再利用性も高まる。図2で、 $D_1 \sim D_n$  は業種ドメイン、 $D_{1M} \sim D_{2M}$  はドメインモデル、 $gt_1 \sim gt_m$  は共通業務を表す。図2の下に示すとおり、共通業務は、それぞれのドメインで異なった割合で行われている。

### 3. プロトタイピング

ソフトウェア開発やシステム開発のとき、分析や設計を支援する手法としてプロトタイピングがある。プロトタイピングは、製品の原型(プロトタイプ)を作り、開発の各段階で、要求仕様、設計仕様が機能面、性能面、コスト面などについてユーザや設計者の意図したものとは異なっていないかを検証する開発手法である。一般に、仕様や設計の変更は、開発の後の段階になるほど、変更にかかる時間や労力が大きくなる。プロトタイピングは、製品開発の早い段階で誤りが入り込むのを防ぐことができるため、最終段階での大きな仕様の変更、再設計が少なくなり、効率的に設計・開発を行うことができる。特に開発の初期段階でプロトタイピングを行うことは有効である。

プロトタイピングを行うときは、プロトタイプの記述、稼働、評価のサイクルを回りながら最終的な製品の仕様を固めていく。これをプロトタイピングサイクルと呼ぶ。記述されたプロトタイプを実際に動かすことで、その振舞いや性能を視覚的、または、定量的に確認し、要求との差異を小さくするよう記述を修正することで、より精密なプロトタイプを作る。プロトタイピングを行うとき、プロトタイプを作るために必要な情報、プロトタイプの構成要素、チューニングパラメータなどは、ドメインに依存している。プロトタイピングは、それぞれのドメインに固有のプロセスに従って行われる。つまり、プロトタイピングサイクルのモデルは、そのドメインでのみ再利用可能である。

プロトタイピングサイクルをさらに再利用するために共通業務の概念を導入する。共通業務ごとに、プロトタイピングサイクルを明らかにしモデル化しておく、異なる業種のドメインでも、共通業務ごとに、主な構成要素、振舞いなどは同じであり、異なる業種のドメインで、プロトタイピングサイクルの再利用が可能となる。ドメインごとに異なる用語、構造、知識、制約条件などは、ライブラリとして持たせ、共通のプロトタイピングサイクルに、対象となるドメインのライブラリをリンクすることでドメインに応じたプロトタイピングを行うことができる。一度、プロトタイピングサイクルやライブラリをモデルとして獲得しておけば、新しくプロトタイプを作成するとき、要求分析の労力を軽減できる。また、新たなドメインのプロトタイプを作成する場合も、そのドメインのライブラリを獲得すれば、プロトタイピングサイクルは共通であり再利用可能である。

共通業務のプロトタイピングサイクルは、まず、対象について共通業務の観点から分析を行う、分析を行った結果、その共通業務を構成する構成要素や、関連、制約条件などが得られる。これらは、それぞれの共通業務を記述するのに適した表や図面などの形式で記述される。次に、得られた分析結果が正しいか評価する。前述の図面などは、静的な記述であるため、これを動的な記述(プロトタイプ)へと変換し、実際に動かしてそれが期待どおりであるか評価する。もし、期待どおりでない場合には、また分析の段階へ戻る。このサイクルを繰り返し、プロトタイピングを進めていく(図3(a))。共通業務のプロトタイピングの例として、協調業務では、分析の結果、業務にかかわる、要員、作業、もの、情報と、ものや情報の流れ、要員とものや情報の共有関係などを得ることができる。その結果を評価するために、作業にかかる時間や、割

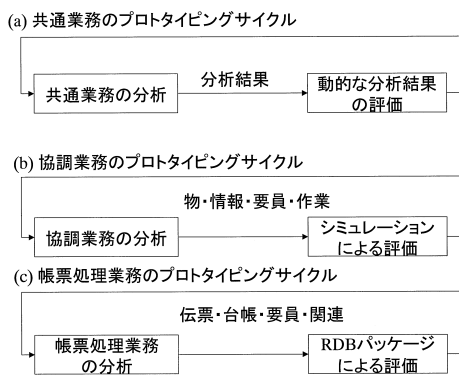


図3 共通業務におけるプロトタイプング

Fig. 3 Prototyping in generic task.

り当てることのできる要員数などの具体的な値を付加することで、シミュレーションモデルに変換し、シミュレーションを行うことで、性能面から評価を行う(図3(b))。また、帳票処理業務では、分析の結果、伝票、台帳、要員とそれらの関連などが得られる。それをデータベース言語に変換し、データベース上で稼働させ、期待どおりに動作するかどうか、機能面から評価を行う(図3(c))。

#### 4. 共通業務分析の例

本稿では、共通業務分析の例として、協調業務分析と帳票処理業務分析の2つを取り上げ説明する。協調業務<sup>8)</sup>は、複数の要員、もの、設備が相互に情報交換しながら、ある特定の目的を遂行する業務である。この業務では、業務にかかわる要員、もの、情報の複雑な相互作用が存在している。したがって、システム化対象となる業務をまず協調業務の観点から分析することで、対象となる業務活動の位置づけが他の相互作用をとみなわない関連業務との関係から明確にできると考えられる。一方、帳票処理業務の観点から分析することで、その業務が必要とする情報や加工し蓄積すべきデータを明確にできる。これらの情報やデータは監視・運用業務<sup>9)</sup>、割当て業務にも利用できる。

分析対象ドメインとして、教務ドメインを取り上げる。このドメインは、講義科目の管理、成績表作成、進級査定、学生証発行など、学校の教務情報を扱うドメインである。

教務ドメインでは、教員が記入した成績を職員が集め成績のデータベースを更新する業務を見ると、教員の記入した成績表は伝票、成績のデータベースは台帳と考えることができる。この業務は、主に帳票類を扱うことから帳票処理業務と考えることができる。また、別の共通業務の観点では、教員と職員の間では協調作

業が行われているととらえることができる。職員は、学生の履修登録などの作業を行い、作成されたリストを教員に渡し、教員は、そのリストに評価を記入し職員に渡す。また、講義や試験の時間割の作成、教員と職員の仕事の割当てを行う割当て業務も存在する。

本稿では、教務ドメインでの履修登録から成績評価を行うまでの一連の作業を行う教務情報システムを対象とする。教務情報システムの開発では、協調業務と帳票処理業務の観点からの分析が必要である。次節以降で、まず、協調業務、帳票処理業務の観点からの分析プロセスを示し、教務ドメインにそれぞれの分析プロセスを適用する。

##### 4.1 協調業務の観点からの分析

協調業務の観点からの分析は、まず、業務の中に存在する情報、もの、要員、作業を明らかにする。次に、作業にかかる時間やものの量、要員数などの情報を付加することでシミュレーションモデルへ変換し、シミュレーションを行い評価する。この評価結果が期待どおりでなければ、また、分析から始める(図3(b))。

以下に協調業務の一般的な分析プロセスを示す。

##### Step 1 マテリアルの列挙

協調作業にかかわるマテリアル(もの)を列挙する。マテリアルとは、図面、台帳、患者などである。

##### Step 2 要員の列挙

協調作業にかかわる要員を列挙する。職員、教員などが協調作業にかかわる要員である。

##### Step 3 作業の列挙

業務の中で行われている、協調作業を列挙する。診察、受付、履修登録などが作業として列挙される。

##### Step 4 要員間でやりとりされる情報の列挙

協調作業を行うときに、その作業にかかわっている要員の間でやりとりされている情報を列挙する。診察ドメインでは、患者に関する情報や、処方する薬の情報など、教務ドメインでは、履修に関する情報や、成績などである。

協調業務の観点から分析したものを記述する手法として、MCM(Multi-Context Map)やCLM(Collaborative Linkage Map)<sup>10)</sup>などがある。MCMで教務情報システムを記述したものが図4である。

MCMでは、作業を箱(CM: Context Map)の1つ1つで表現する。協調作業には、様々な人、組織、装置がかかわり、それらは、異なったものの方や立場を持つ。これをパースペクティブと呼ぶ。その箱の左上に作業を依頼する側のパースペクティブ(左側パースペクティブ)、右上に作業を行う側のパースペクティブ(右側パースペクティブ)を書く。両パースペクティブ

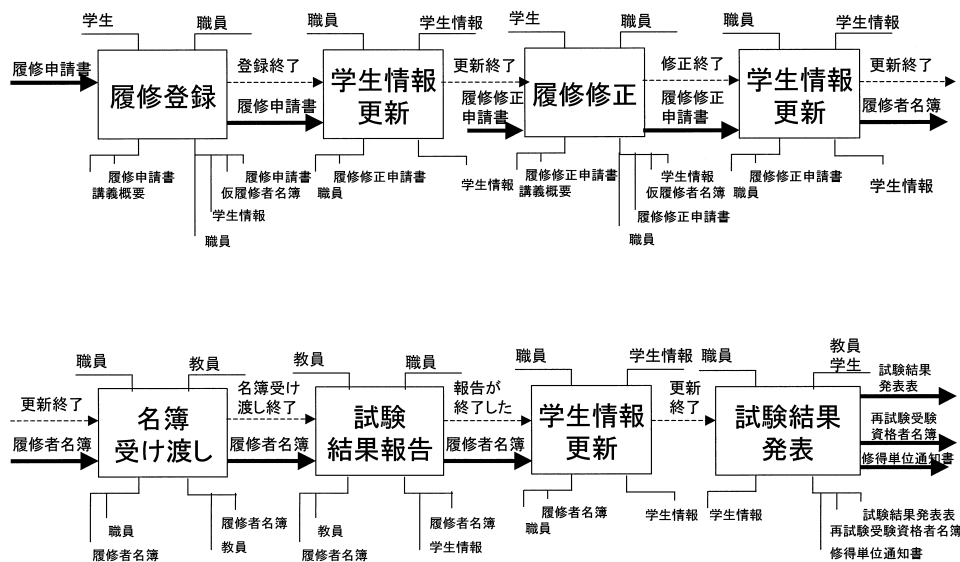


図 4 教務情報システムの MCM  
Fig. 4 MCM in academic affairs information system.

ブ間では、情報（インフォメーション）やマテリアル、パースペクティブの間で認証される事実（トークン）を発生する。MCMでは、インフォメーションの流れは太線、マテリアルの流れは細線、トークンの流れは破線で表される。また、CMの右下、左下には、作業で使用される設備やものなどの資源マテリアルを記述する。MCMでは、要員が認証するトークンの流れ、マテリアルの流れ、インフォメーションの流れが区別して記述できるため、それぞれの要員がどのトークンやインフォメーションやマテリアルを受け取ったかが明確になる。また、同じマテリアルに対して各要員が認識する情報の違いも記述可能である。このようにMCMは対象ドメインを協調業務の観点から記述するのに適したチャートである。

教務情報システムをMCMで記述した例を図4に示す。図4は、講義の履修についての協調業務を表している。まず、学生が履修登録申請書を作成し、職員との間で履修登録が行われる。職員は、学生情報のデータベースに履修登録の内容を反映させる。次に履修の修正や追加登録を行うために、学生は履修修正申請書を作成し、履修登録と同様に、職員が履修修正を学生情報に反映させ、履修者名簿が作成される。作成された履修者名簿は職員から教員に渡される。教員は学生に対して講義と試験を行う。教員は試験の結果を履修者名簿に記入して職員に報告する。職員は、報告をもとに学生情報に成績の登録を行い、試験結果発表表、再試験受験資格者名簿、修得単位通知書を作成する。これらの作業は、学生-職員間、職員-学生情報を

扱う装置間、職員-教員間の協調作業であり、図4のMCMのCM1つ1つに対応している。

#### 4.2 帳票処理業務の観点からの分析

帳票処理業務の観点からの分析は、要員、伝票、台帳の3つを認識することから始める。帳票処理業務では、業務にかかわる要員が、売買に関する情報を記入した伝票をやりとりし、伝票に書かれた内容と在庫状況を台帳を用いて管理する。教務情報システムでは、伝票、台帳という用語は使われていないが、履修申請書や修得単位通知書は伝票、学生情報は台帳と考えることができる。要員、台帳、伝票を認識し、これら3つの関連を探し出すことで帳票処理業務を分析する。この分析結果は、データベース言語に変換することで、実際に動かすことが可能で、正しく動くか評価できる（図3(c)）。以下に、帳票処理業務一般の分析プロセスを示す。この分析プロセスは、異なったドメインにおいても利用可能である。

##### Step 1 要員を認識

対象業務にかかわる要員を認識する。

##### Step 2 台帳を認識

対象業務に含まれる台帳を認識する。

##### Step 3 伝票を認識

要員の間で受け渡しされているものを伝票として認識する。

##### Step 4 要員と台帳の関連を認識

要員と台帳の関連を認識する。関連には以下の2通りがある。

- (1) 台帳から情報を読み取る。

(2) 台帳に情報を記入する(更新する)。

#### Step 5 要員と伝票の関連を認識

要員と伝票の関連を認識する。関連には以下の4通りがある。

- (1) 受け渡し：伝票は、ある要員から渡され、ある要員が受け取る。
- (2) 読み取り：伝票を受け取った要員は、伝票から何らかの情報を読み取る。
- (3) 作成：伝票を渡す要員は、その伝票を作成する。渡そうとしている伝票が、他の要員から受け取ったものである場合には、作成されていない。
- (4) 記入：伝票を渡すときには、その伝票に対して情報が記入されている。

#### Step 6 トランザクションの認識

Step 4, Step 5 で得られた関連をもとに、要員が伝票に従って台帳を更新したり、台帳に従って伝票を発行したりする一連の操作の流れをトランザクションとして認識する。

以上の分析で、要員の間で、どのような伝票がやりとりされ、その伝票に対してどのような操作が行われているかという関連が明らかになる。

4.3 協調業務と帳票処理業務の統合分析プロセス  
帳票処理業務の分析は、4.2 節の分析プロセスに従って行うことで分析できるが、このとき必要となる情報は、4.1 節の協調業務の分析で得られた結果に含まれている。そこで、まず、4.1 節の協調業務の観点からの分析を行い、その結果を利用することで、よりスムーズな分析を行うことが可能となる。以下では、MCMを用いた協調業務の観点からの分析結果をもとにした帳票処理業務の分析プロセスを示す。図4のMCMをもとに、帳票処理業務の観点から教務ドメインを分析する。

#### Step 1 要員を認識

パースペクティブから学生、職員、教員を要員として認識する。

#### Step 2 台帳を認識

パースペクティブで、Step 1 で要員として認識したものの以外を台帳として認識する。ここでは、学生情報を台帳として認識する。

#### Step 3 伝票を認識

資源マテリアルから伝票を認識する。資源マテリアルには、職員などの要員、学生情報などの台帳も含まれているが、ここでは、資源マテリアルの中でコンテキスト間をマテリアルとして流れている履修申請書、履修修正申請書、履修者名簿、試

験結果発表表、再試験受験資格者名簿、修得単位通知書を伝票として認識する。

#### Step 4 要員と台帳の関連を認識

要員と台帳がパースペクティブになっているCMに注目して、要員と台帳の関連を認識する。

「学生情報更新」のCMから「職員は、学生情報を更新する」という関連を認識する。

#### Step 5 要員と伝票の関連を認識

- 学生は、履修申請書を作成。
- 学生は、履修申請書を職員に渡す。
- 職員は、履修申請書を受け取り、情報を読み取る。
- 学生は、履修修正申請書を作成。
- 学生は、履修修正申請書を職員に渡す。
- 職員は、履修修正申請書を受け取り、情報を読み取る。
- 職員は、履修者名簿を作成。
- 職員は、履修者名簿を教員に渡す。
- 教員は、履修者名簿を受け取り情報を読み取る。
- 教員は、履修者名簿に情報を記入し、職員に渡す。
- 職員は、履修者名簿から情報を読み取り、学生情報を更新する。
- 職員は、試験結果発表表、再試験受験資格者名簿、修得単位通知書を作成。

という関連が得られる。

#### Step 6 トランザクションの認識

トランザクションの認識は、Step 4 で得られた要員と台帳の関連と Step 5 で得られた要員と伝票の関連をもとに行う。Step 4 で「職員は、学生情報を更新する」という関連が得られている。そこで「学生情報更新」のトランザクションを考える。この学生情報更新のCMについては、Step 5 で「職員は、履修申請書を受け取り、情報を読み取る」、「職員は、履修修正申請書を受け取り、情報を読み取る」、「職員は、履修者名簿を受け取り、情報を読み取る」という3つの関連が得られているので、履修申請書、履修修正申請書、履修者名簿の情報をもとに学生情報を更新していると考えられる。そこで、履修申請書の情報をもとに学生情報に、履修科目の登録を行う「履修登録」をトランザクションとして認識し、同様に、「履修修正」、「成績登録」のトランザクションを認識する。さらに、履修者名簿作成、試験結果発表表作成、再試験受験資格者名簿作成、修得科目一

覧表作成、修得単位通知書作成をトランザクションとして認識する。

#### 4.4 協調業務の性能プロトタイピング

MCM で記述された協調業務の観点からの分析結果は、どのような要員、作業、もの、情報があるか、また、それらがどういう関係を持っているかという静的な記述で、業務全体がどのように行われているかをとらえることができるが、全体の作業時間などを定量的に評価することは難しい。そこで、協調業務を性能面から分析し評価するために、トランザクション型シミュレータの GPSS<sup>11)</sup> 上に協調業務のモデルを構築し、シミュレーションを行う。シミュレーションを行うために、MCM で記述された協調業務モデルをシミュレーションモデルへ変換する。このモデルをシミュレータ上で動かすことで性能を評価できる。シミュレーションモデルへは、協調業務のものや情報の流れをトランザクションの流れととらえ、要員や組織によって、ものや情報を使っている状態を、要員というリソースが

ものや情報というトランザクションによって占有された状態と考えることで変換できる<sup>12)</sup>。

4.1 節で分析された、教務情報システムの協調業務のシミュレーションを、GPSS を用いて行った結果を図 5 に示す。学生 1 人あたりの履修登録にかかる時間を平均 30 秒、学生情報更新に平均 6 秒、学生は 1000 人で、平均 10 秒の指数分布に従って到着するものとし、すべての学生の履修登録と学生情報更新の処理にかかる時間が、職員数によって、どのように変わるかを見た。この負荷では職員が 4 人必要であることが分かる。履修修正以降の処理は、すべての学生の情報更新が終わって 1 日以上間隔をおいてから行うので、学生情報の更新までで 1 つのシミュレーションとした。

このように、協調業務の観点からの分析結果をシミュレーションモデルへ変換しシミュレーションすることで、協調業務について、性能面から定量的に評価を行うことができる。シミュレーションの結果では、業務全体の効率だけでなく、作業ごとの稼働率や、待ちの長さなどについても評価できる。

MCM で記述されたモデルから GPSS 上で動くシミュレーションモデルへは、表 1 に示された対応関係で変換する。この変換作業をコンピュータで支援するトランスレータを作っている。作業にかかる時間や、要員数などは、実際の業務のものを入れる。

#### 4.5 帳票処理業務のプロトタイピング

4.2 節の帳票処理業務の観点からの分析結果は、データベース用言語の SQL 形式で記述される。SQL 形式の記述は、データベースソフトに入力することで、実

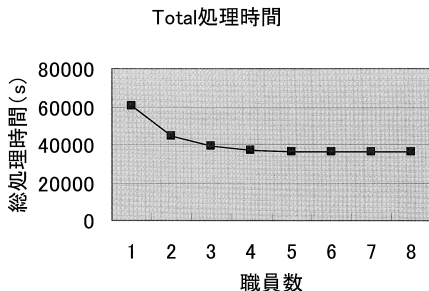


図 5 協調業務のシミュレーション結果  
Fig. 5 Simulation of collaboration task.

表 1 MCM の構成要素と GPSS の構成要素の対応

Table 1 Relationships between components of MCM and components of GPSS.

MCM の構成要素	GPSS の構成要素	備考
コンテキスト パースペクティブ 資源マテリアル	STORAGE	STORAGE は、トランザクションが使用するエンティティで、1 以上の容量を持つことができる。STORAGE に空きがあれば、後続のトランザクションも、STORAGE を使用できる。パースペクティブや資源マテリアルの容量が制約条件であるので、コンテキストには、パースペクティブや資源マテリアルより大きい容量の STORAGE を割り当てる。
トークン マテリアル インフォメーション	トランザクション	<ul style="list-style-type: none"> <li>トランザクションは 1 つ発生させ、それを複製してトークン、インフォメーション、マテリアルの区別をするパラメータをトランザクションに持たせる。コンテキストに入るときには、3 つを 1 つに合成する。つまりトークン、マテリアル、トークンの 3 つがそろわないと次のコンテキストに進むことができない。</li> <li>最後のコンテキストや Junction を出たトランザクションは、消滅 (Terminate) させる。</li> </ul>
Serialization	-	トランザクションは順番に到着するので、そのまま目的のコンテキストの待ち行列 (queue) に並ぶ。
Synchronization	GATE	複数の経路の合流するところで、それぞれの入り口に GATE を置き、互いの GATE の状態を監視することで同期をとる。
Branch	TEST	トークン、マテリアル、インフォメーションによって分岐先が決まっている場合は、条件分岐。そうでない場合は、確率で分岐。

```

create table 授業料(学生番号 number(8,0),債権金額
number(8,0),履行期限 date,減少額 number(8,0));
create table 学科(学科名 char(50));
create table 学部(学部名 char(50));
create table 教官(教官番号 number(8,0),教官氏名
char(50),所属 char(50));
create table 科目(科目番号 number(8,0),科目名
char(50),単位区分 char(50),単位数 number(8,0),開講
年次 number(8,0),学期区分 char(50),教室番号
number(8,0));
create table 履修申告(年度 number(8,0));
create table 成績処理(日付 char(50));
create table 個人履修(科目番号 number(8,0),得点
number(8,0),単位数 number(8,0),取得年次
number(8,0),単位区分 char(50),開講年次
number(8,0),教官番号 number(8,0));
create table カリキュラム(卒業・進級要件 char(50));
.
.
.
***学生情報登録***

/*学生テーブルをまず作成*/
create table 学生(
  氏名 char(20),国籍 char(20),性別 char(2),住所
char(50),
  電話番号 number(10),改姓年月日 date,生年月日 date,
  出身学校 char(16),
  出身都道府県 char(6),入学区分 char(10),入試結果
char(50),学年 number(1),
  学科 char(20),学籍区分 char(6),卒業退学年月日 date,
  卒業退学学位
  char(6),保証人氏名 char(20),続き柄 char(6),保
  証人住所 char(50),
  保証人電話番号 number(10),入学年度
  number(4),在学年限 number(4),
  学生番号 char(6),学生証発行枚数 number(2));
/*学生登録のプロシージャ作成*/
begin
insert into 学生 (
  氏名,国籍,性別,住所,電話番号,改姓年月日,生年
  月日,出身学校,出身都道府県,
  入学区分,入試結果,学年,学科,学籍区分,卒業退
  学年月日,卒業退学学位,
  保証人氏名,続き柄,保証人住所,保証人電話番号,
  入学年度,在学年限,学生番号,
  学生証発行枚数)
values (
  p_name,p_nationality,p_sex,p_address,p_tel,p
  _changenamedate,p_birthday,

  p_school,p_location,p_enterdev,p_enterexam,p
  _gakunen,p_department,

  p_bmddev,p_graddate,p_gradgakui,p_hname,p
  _relation,p_haddress,p_hitel,

  p_enteryear,p_gradlimit,p_studentid,p_stuidp
  rintno);
commit;
end AddNewStudent;
/

```

図 6 SQL 出力の例

Fig. 6 Example of output of SQL form.

際に稼働させることができ、期待通りの動作をするかどうか評価可能である。図 6 は、教務情報システムの SQL 形式での出力の例である。共通業務のプロトタイプをより効率的に行うために、プロトタイプをコンピュータを用いて支援することが考えられる。本稿では、共通業務のプロトタイプツールとして、帳票処理業務のプロトタイプツール「NAVI」を作成した。「NAVI」には、前述の帳票処理業務の分析プロセスを実装し、いろいろなドメインの帳票処理業務のプロトタイプを支援する。NAVI では、分析プロセスに従って分析対象の要員、台帳、伝票を認識し、これらの関連を定義する。次に台帳、伝票のデータ構造、トランザクションを定義する。分析結果のプロトタイプは、オブジェクト指向ののった RSL 表現<sup>13),14)</sup>、または SQL 表現で出力される。出力された SQL は、データベースソフトに入力して実行できる。ユーザは、この実行結果からシステムの設計が要求にあっていないかを確かめる。NAVI では、販売ドメインの帳票処理業務、プラントドメインの帳票処理業務など、複数のドメインモデルを持ち、それぞれのドメインモデルには、A 社のモデル、B 社のモデルなどの、インスタンスライブラリを持つ。ユーザは、このイン

スタンスライブラリの中から自分のシステムに近いモデルを選び、ユーザのシステムで使用している用語への置き換えや、データ構造の変更を行い、プロトタイプを作成する。得られたプロトタイプは、新たなドメインモデルライブラリやシステムインスタンスライブラリとして NAVI に蓄積でき、また再利用が可能である。この NAVI を用いた帳票処理業務のプロトタイプでは、NAVI に分析プロセスが実装されていることで、効率的に分析できる。

図 7、図 8 は、NAVI の画面例である。図 7 は、学生情報の属性の定義画面である。図 8 は、トランザクションの定義画面である。先に示した図 6 は、NAVI によって出力された SQL である。

出力された SQL を利用して作成した教務情報システムのアプリケーションの例として、履修一覧表示と学生情報登録を示す。図 9、図 10 は、それぞれの画面例である。履修一覧表示では、データベースから学生の履修一覧を SQL の select 文を用いて取得している。一覧の取得部分は NAVI で出力された SQL をアプリケーションに組み込んでいる。学生情報登録では、入力された名前や住所などの情報を引数として、データベースに格納された procedure を呼び出している。





学生番号	科目番号	科目名	学字学科区分	単位区分	単位数	開講年次	学期区分	教員番号	履修
950000	719310	電気工学	学科	選択	2	1998	前期	720002	A
950001	1071	法学	全学	必修	4	1998	適年		1 A
950001	90117	保健教育	全学	必修	1	1998	前期		2 A
950001	90251	保健理論	全学	必修	1	1998	前期		18 A
950001	90700	体育実技A	全学	必修	1	1998	前期		3 A
950001	90701	体育実技B	全学	必修	1	1998	後期		3 A
950001	850271	英語会話A	全学	外国語	2	1998	前期		4 B
950001	850371	英作文	全学	外国語	2	1998	後期		4 A
950001	852071	英語	全学	外国語	2	1998	適年		5 C
950001	852271	英語会話B	全学	外国語	2	1998	適年		6 A
950001	102110	経営学	全学	選択	2	1998	後期		9 A
950001	713371	化学	全学	選択	2	1998	前期		1 B
950001	730116	物理	全学	選択	2	1998	後期		10 A
950001	53402	生物学A	全学	選択	2	1998	前期		17 B
950001	700104	微分方程式	学科	必修	2	1998	後期	700006	B
950001	703101	工業力学	学科	選択	2	1998	後期	700017	B
950001	712701	CGアニメーション	学科	選択	2	1998	後期	700009	A
950001	719310	量子力学	学科	選択	2	1998	前期	720002	A
950001	30512	憲法	全学	選択	2	1998	前期		12 A
950001	30698	経済学	全学	選択	4	1998	適年		13 B
950001	34100	国際政治学	全学	選択	2	1998	後期		14 B
950001	53403	生物学B	全学	選択	2	1998	後期		17 B

図 9 履修一覧表示画面

Fig. 9 Screen of making a list of courses.

学生番号: 950000 学年: 3

学科: 設計工 学籍区分: 学部

氏名: ○○△× 国籍: 日本

性別: 男

住所: 神奈川県横浜市

電話番号: XXXX-YYYY

改姓年月日: 生年月日: 76/01/01

出身学校: A高校 出身都道府県: 神奈川県

入学区分: 一般 入試結果: 合格

卒業退学年月日: 卒業退学学位:

入学年度: 1995 在学年限: 2003 学生証発行枚数: 1

保証人氏名: 続柄: 父

保証人住所:

保証人電話番号: YYYY-ZZZZ

図 10 学生情報登録画面

Fig. 10 Screen of registration of students.

## 5. おわりに

ドメインを共通業務に着目して分析・モデル化することで、ドメインモデルの再利用や、近隣ドメインへの転用が容易になる。本稿では、共通業務の帳票処理業務、協調業務について、分析プロセスを提案し、教務ドメインを協調業務、帳票処理業務の観点から分析した。また、それぞれの共通業務について、分析プロセスを実装したプロトタイピングツールを試作した。これらのツールを用いることで、協調業務の観点からは、分析結果をシミュレーションモデルに変換し、実際の作業の時間や要員数などの量を与えることで、個々の作業時間を見積もることができる。また、帳票処理業務の観点から、データベース言語のSQL記述の出力を得てデータベースで稼働させることで期待どおりの振舞いをするかどうかを確認できる。

本稿で例題として扱った教務ドメインは、いくつかの大学の教務作業を分析して得たものである。

今後、他の共通業務についてもプロトタイピングを支援するツールの開発を検討し、いろいろなドメインでこれらのツールを利用して、再利用性の高いモデルを獲得することが重要である。現段階で認識している共通業務は、帳票処理業務、協調業務、監視運用業務、割当て業務の4つであるが、多くのドメインを分析していく過程で、新たな共通業務が明らかになることも予想される。

今後の課題として、他の共通業務のプロトタイピング支援ツールの開発と、NAVIの改良を検討している。現在のNAVIでは、構成要素の不足や、関連の矛盾などを見つけてあげることができない。これらの欠落や矛盾を調べ、警告を出す仕組みを取り入れることで、より効率的なプロトタイピングの支援ができると期待する。

謝辞 本研究は、一部、文部省ハイテクリサーチセンタープロジェクトの補助による。

本稿の NAVI の原型は、一部 EAGL の補助による。この原型の設計に助力いただいた、本学の田村恭久氏、1995 年度修士の平井俊光氏、1996 年度修士の各務辰哉氏に感謝する。

MCM の設計を行っている、博士課程在籍の長谷川明子氏、全体の試作に助力いただいた 1998 年度 4 年生の城野孝氏、1999 年度 4 年生の戸波幹氏に感謝する。

### 参考文献

- 1) Arango, G. and Prieto-Diaz, R.: Introduction and Overview: Domain Analysis Concepts and Research Directions, *Domain Analysis and Software Systems Modeling*, Prieto-Diaz, R. and Arangoed, G. (Eds.), pp.9-26, IEEE (1991).
- 2) 伊藤 潔, 杵島修三, 田村恭久, 廣田豊彦, 吉田裕之: ドメイン分析・モデリング, 共立出版 (1996).
- 3) 川端 亮, 田村恭久, 伊藤 潔, 富井規雄: 割当て業務のドメイン分析・モデリングのための支援チャート, ソフトウェア工学の基礎シンポジウム (1997).
- 4) Kawabata, R. and Itoh, K.: System Analysis by Navigator for Collaboration Task and Trading Task, *SCS ECEC'98 (European Concurrent Engineering Conference)*, pp.307-309 (1998).
- 5) 川端 亮, 長谷川明子, 熊谷 敏, 伊藤 潔: 共通業務のドメイン分析に基づくプロトタイピングシステム, ソフトウェア工学基礎論文集, Vol.5 (1998).
- 6) Kawabata, R., Itoh, K. and Kumagai, S.: Prototyping based on Domain Analysis and Modeling, *4th IDPT* (July 1999)
- 7) 伊藤 潔, 本位田真一: プロトタイピング支援ツール, *情報処理*, Vol.30, No.4, pp.387-395 (1989).
- 8) Kumagai, S., Hasegawa, A., Kawabata, R. and Itoh, K.: Building Workflow for Collaboration Task using Multi-Context Map, *Journal of Society of Design and Process Science*, Vol.4, No.3, pp.49-61 (2000).
- 9) Kumagai, S. and Itoh, K.: Domain Analysis and Modeling for Concurrent Engineering in Plant Monitoring and Operations, *IEEE Compsac'97*, pp.450-453 (1997).
- 10) Hasegawa, A., Kumagai, S. and Itoh, K.: Collaboration Task Analysis by Identifying Multi-Context and Collaborative Linkage, *Journal of Concurrent Engineering Journal*, Vol.8, No.1/2 (2000).
- 11) GPSS/H Reference Manual, Wolverine Software (1994).
- 12) 川端 亮, 戸波 幹, 熊谷 敏, 伊藤 潔: トランザクション型シミュレータを用いた協調業務の性能評価, サマーワークショップイン小樽 (1999).
- 13) 原田 実: SOME における動的モデリングの詳細化と設計図の自動レイアウト CAMEO/D と C++ プログラムからの設計図の逆生成, 情報処理学会 OO' 96 シンポジウム論文集, pp.111-118 (1996).
- 14) 原田 実, 田村浩樹, 野村佳秀: オブジェクト指向分析システム CAMEO/A と帰納推論によるルールの学習, 情報処理学会 OO' 96 シンポジウム論文集, pp.57-64 (1996).

(平成 11 年 10 月 22 日受付)

(平成 12 年 7 月 5 日採録)



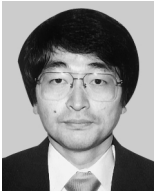
川端 亮 (正会員)

1996 年上智大学理工学部機械工学科卒業。1998 年同大学大学院理工学研究科機械工学専攻修士課程修了。同年より上智大学に勤務。現在、同理工学部機会工学科情報システム講座所属。ソフトウェア工学、ドメイン分析・モデリング等の研究に従事。



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 年同大学大学院理工学研究科情報工学専攻博士課程満期。京都大学工学博士。1979 年より上智大学理工学部勤務。現在、理工学部機械工学科情報システム講座教授。ソフトウェア工学、人工知能、ドメイン分析・モデリング、コラボレーションエンジニアリング、シミュレーション手法の研究に従事。本学会学会誌・論文誌の編集委員歴任。「ドメイン分析・モデリング：これからのソフトウェア開発・再利用基幹技術」(共著, 共立出版)「Domain Oriented Systems Development: Principles and Approaches」(共著, 情報処理学会英文図書第 1 巻) IEEE, ACM, SDPS 等会員。



熊谷 敏

1986 年慶応義塾大学工学部管理  
工学科卒業．1988 年同大学大学院  
工学研究科修士課程管理工学専攻修  
了．同年山武ハネウエル(株)入社．  
現在(株)山武研究開発本部に勤務．

システムエンジニアリング，業務プロセスモデリング  
等の研究開発に従事．工学博士．

---