

経営情報システムの分析のためのドメインモデル

斉藤 康彦[†]

本位田 真一[‡]

情報処理振興事業協会 (IPA) 新ソフトウェア構造化モデル研究本部

経営情報システムの分析では、大規模化したメタデータベースに格納すべきデータ項目の決定が難しい。また、環境の変化に適応してメタデータベースを再編する必要がある。本稿では、情報システムを利用する業務の視点からメタデータベースを分割し、分割された部分の間の相互作用を把握するためのドメインモデルと、それに基づく要求分析の方法論を提示する。ドメインモデルは、(1) 実世界における現象を表現するモデル、(2) 情報システムの機能を表現するモデル、(3) 実世界と情報システムの間で共有される語彙の集合として両者の対応関係を表現するデータ辞書から構成される。要求分析の方法論は、データ辞書の構築と再編を支援する手法である。

A Domain Model for Analysis of Management Information Systems

Yasuhiko Saito

Shinichi Honiden

Laboratory for New Software Architectures,
Information-technology Promotion Agency(IPA),
3-1-38 Shibakoen, Minato-ku, Tokyo 105, Japan

It is difficult for analysts of management information systems to identify items to be stored in large-scale metadatabases, which should be adapted to changes in environments. We presents an analysis method based on a domain model for dividing the metadatabase according to views of the users and for capturing interactions among the parts. The domain model includes: 1) models for phenomena in the real world; 2) models for functions of an information system; 3) a data dictionary, which is a set of vocabularies shared by the real world and an information system. The analysis method supports organization and reorganization of the data dictionary.

[†](株) アイネスより出向.

[‡](株) 東芝より出向.

1. はじめに

経営情報システムのドメインでは、設計と製造の技術がかなり確立されてきた。その一方で、導入はしたが、期待していたほど有効に利用されていないシステムも少なくない。これは、分析が十分でなかったことが原因であると考えられる。

経営情報システムは、「データは資源である」という情報資源管理の考え方を基本的な前提とする。したがって、一般に、データ中心のアプローチで開発される[4]。このようなシステムの分析では、システムの中核となるデータベースに格納するデータ項目を同定すること、すなわち、メタデータベースの定義が重要になる。しかし、企業が管理の対象とする情報資源の量的な増大化と質的な多様化によって、データベースのみならず、メタデータベースの大規模化を余儀なくされている。また、企業組織内外の環境の急激な変化は、メタデータベースの保守をさらに困難なものにしている。

メタデータベースの管理を支援するためのツールとしては、従来から、DD/DS (Data Dictionary / Directory System) が使用されてきた[6]。近年、DD/DSは、ソフトウェアのライフサイクルの支援を目的とするリポジトリとして発展してきている。リポジトリは、情報システムの開発局面における視点を統合する。しかし、大規模なメタデータベースでは、格納すべきデータ項目の決定や格納されたデータ項目の理解が難しいことから、情報システムの利用局面に対応してメタデータベースを分割し、一貫性を保ちながら、それらを再統合しなければならない。

メタデータベースの分割と統合は、3層スキーマ構成によって、ある程度まで実現される。すなわち、対象世界は、個別のアプリケーションシステムに対応する外部スキーマによって、いくつかの部分に分割されるが、同時に、それらの部分は、対象世界全体を表現する概念スキーマとして統合される。しかし、この場合の各部分は、情報システムの機能の集合に対応しており、情報システムを利用する業務に素直に対応しているとは限らない。そのために、業務の内容が徐々に変化していくことによって、実世界と情報システムとの対応関係が次第に曖昧になり、円滑な入力情報の収集や出力情報の活用が阻害されるようになる。こうして、環境の変化に情報システムが追従しなくなる。

本稿では、環境の変化に適応して情報システムを進化させるための方法論を提示する。本方法論は、

情報システムを利用する業務の視点からメタデータベースを分割し、分割された部分の間の相互作用を把握することによって、それらを統合するためのドメインモデルに基づいている。ここで、ドメインモデルとは、あるドメインに属する問題を解決するために利用可能な知識を、認識し組織化するプロセスの結果として、ドメインごとに構築されるモデルのことである[1]。本稿で提示する経営情報システムのドメインモデルは、実世界における現象を表現するモデル、情報システムの機能を表現するモデル、実世界と情報システムの間で共有される語彙の集合として両者の対応関係を表現するデータ辞書から構成される。本稿で提示する要求分析の方法論は、データ辞書の構築と再編を支援する手法である。データ辞書の再編とは、企業組織のある部分の変化による影響がどのように波及するかを解析し、企業組織に対する情報システムの歪みを矯正することである。これによって、企業組織と情報システムを整合的に進化させることが可能になる。

本稿の構成は、次のとおりである。第2章と第3章では、経営情報システムのドメインモデルと、それに基づく要求分析の方法論を示す。第4章では、人事情報システムの分析を例題として、現実的な問題に対して本方法論をどのように適用するかを示す。第5章では、他の研究との関連を示す。

2. ドメインモデル

経営情報システムの要求分析を目的とするドメインモデルの概念を以下のように定義する。

【業務モデル】

業務モデルは、企業組織の多種多様な業務を個別に記述したものである。特に、情報システムを導入しようとする業務において、人間がそのシステムを利用して行なう情報処理がモデル化の対象となる。したがって、情報システムの機能は、業務モデルとして記述されない。なお、業務モデルの形式や記法は、各業務に依存する。

【要求モデル】

情報システムの機能は、要求モデルとして記述される。要求モデルは、いわゆるソフトウェアの機能仕様書や利用者マニュアルに記述される内容である。なお、要求モデルの形式や記法は、各情報システムに依存する。

【問題領域】

問題領域 P は、業務モデルの集合である。各業

務モデルは、問題領域のセグメントになる。問題領域のセグメントには、以下の種類がある。

- 情報を収集する (collect) モデル
ある事象が発生したときに、伝票や届出用紙などの書式に記入し、それを情報システムに登録するような業務を記述したモデル。
- 情報を加工する (process) モデル
仕訳規則、原価計算規則、人事評価規則、給与決定規則など、情報を加工するための定常的な業務規則を記述したモデル。
- 情報を活用する (use) モデル
情報システムから出力される帳票類に基づいて、原価管理や生産計画などの意思決定を行なうプロセス、あるいは、財務諸表や給与明細書などを特定の目的のために配布するプロセスなどを記述したモデル。

情報を収集するモデルの集合を P_c とし、情報を加工するモデルの集合を P_p とし、情報を活用するモデルの集合を P_u とする。

[解領域]

解領域 S は、要求モデルの集合である。各要求モデルは、解領域のセグメントになる。解領域のセグメントは、以下の要素から構成される。

- 入力情報 (input data : id) モデル
入力画面などで指定されるデータ項目 (トランザクション情報) と、データベースなどに格納されるデータ項目 (マスタ情報)、および、変更される可能性のある業務規則に関連するデータ項目を規定する。原料として管理される情報のモデルである。
- 入力処理 (input process : ip) モデル
入力情報モデルによって定義されるデータ項目に関する手続きを規定する。データベース更新処理や入力画面インタフェース処理などが含まれる。
- 出力情報 (output data : od) モデル
画面や帳票などに出力されるデータ項目を規定する。製品として生産される情報のモデルである。
- 出力処理 (output process : op) モデル
出力情報モデルによって定義されるデータ項目に関する手続きを規定する。帳票作成処理や出力画面インタフェース処理などが含まれる。

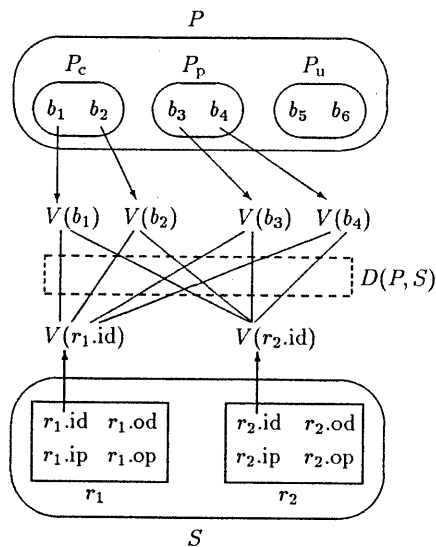


図 1: 経営情報システムのドメインモデル

要求モデル r についての各構成要素を $r.id$, $r.ip$, $r.od$, $r.op$ で参照すると、解領域のセグメントの構成要素は、以下のように分類される。

$$\begin{aligned} Sid &= \{r.id \mid r \in S\} \\ Sip &= \{r.ip \mid r \in S\} \\ Sod &= \{r.od \mid r \in S\} \\ Sop &= \{r.op \mid r \in S\} \end{aligned}$$

[語彙]

語彙 $V(m)$ は、モデル m を記述するために用いる語の集合である。

[共通語彙]

共通語彙 $C(b,r)$ は、 $V(b)$ (ただし、 $b \in P_c \cup P_p$) と $V(r.id)$ (ただし、 $r \in S$) の共通集合である。このとき、 $V(b)$ と $V(r.id)$ の間の概念上の対応を $V(b) \times V(r.id)$ の部分集合として定義したものが $\{(x,x) \mid x \in C(b,r)\}$ となるように、問題領域と解領域の間で語彙を統一することによって、セグメント間で語彙を標準化しておく。

[データ辞書]

データ辞書 $D(P,S)$ は、共通語彙から構成される。たとえば、 $P_c \cup P_p = \{b_1, b_2, b_3\}$, $S = \{r_1, r_2\}$ である場合のデータ辞書は、 $C(b_1, r_1)$, $C(b_1, r_2)$, $C(b_2, r_1)$, $C(b_2, r_2)$, $C(b_3, r_1)$, $C(b_3, r_2)$ によって定義される。

これらの概念の間の関係を図 1 に示す。

3. 方法論

前章で示したドメインモデルに基づく要求分析の方法論を示す。

3.1 語彙間の包含関係とモデル間の界面

問題領域と解領域を記述するための全語彙は、以下のように分類される。

$$\begin{aligned} V_c &= \bigcup_{b \in P_c} V(b) \\ V_p &= \bigcup_{b \in P_p} V(b) \\ V_u &= \bigcup_{b \in P_u} V(b) \\ V_{id} &= \bigcup_{r \in S} V(r.id) \\ V_{ip} &= \bigcup_{r \in S} V(r.ip) \\ V_{od} &= \bigcup_{r \in S} V(r.od) \\ V_{op} &= \bigcup_{r \in S} V(r.op) \end{aligned}$$

語彙 $V_c, V_p, V_u, V_{id}, V_{ip}, V_{od}, V_{op}$ は、それぞれ、モデルの集合 $P_c, P_p, P_u, S, S, S, S$, Sod, Sop の要素を記述するための語彙である。すなわち、これらの語彙とモデルの集合は、 $V_c \leftrightarrow P_c, V_p \leftrightarrow P_p, V_u \leftrightarrow P_u, V_{id} \leftrightarrow S, V_{ip} \leftrightarrow S, V_{od} \leftrightarrow S, V_{op} \leftrightarrow S$ のように対応付けられる。本方法論では、以下の包含関係が成立するようにモデルと語彙を定義する。

- $V_{id} \subseteq V_c \cup V_p$
(入力情報は収集または加工の対象となる)
- $V_{od} \subseteq V_u$
(出力情報は活用の対象となる)
- $V_{id} \subseteq V_{ip}$
(入力情報は入力処理の対象となる)
- $V_{od} \subseteq V_{op}$
(出力情報は出力処理の対象となる)

これらの語彙間の包含関係から、モデル間の界面が明確になる。すなわち、

- V_{id} を P_c と P_p から抽出できるので、情報システムと実世界の界面が明確になる。
- V_{od} を P_u から抽出できるので、情報システムと実世界の界面が明確になる。
- V_{id} を S_{ip} から抽出できるので、情報システムにおけるデータと手続きの界面が明確になる。
- V_{od} を S_{op} から抽出できるので、情報システムにおけるデータと手続きの界面が明確になる。

3.2 モデル間の依存関係とモデル化の手順

語彙 V_1 が語彙 V_2 を包含するならば、 V_1 に対応するモデルの集合 M_1 は、 V_2 に対応するモデルの集合

$$\begin{array}{ccccccc} P_c \cup P_p & \implies & S_{id} & \longrightarrow & S_{od} & \longleftarrow & P_u \\ & & \uparrow & & \uparrow & & \\ & & S_{ip} & & S_{op} & & \end{array}$$

図 2: モデル間の依存関係

M_2 に依存する ($M_1 \implies M_2$) と考える。また、要求される出力情報が変化すれば、それを得るために必要な入力情報も変化する。したがって、 S_{id} は S_{od} に依存する ($S_{id} \longrightarrow S_{od}$) と考える。そこで、図 2 に示すモデル間の依存関係が成立する。

モデル化の手順は、 S_{od} を開始点とするならば、モデル間の依存関係に基づいて決定する。しかし、実際には、利用者のニーズを表現する P_u と P_p に適合するように、 S_{od} と S_{id} を定義することになるので、以下の手順になる。

1. P_u を定義する。
2. P_u と整合的な S_{od} を定義する。
3. S_{od} にしたがって、 P_u を調整する。
4. P_p を定義する。
5. S_{od} にしたがって、 P_p と整合的な S_{id} を定義する。
6. S_{id} にしたがって、 P_p を調整する。
7. S_{id} にしたがって、 P_c を定義する。
8. S_{od} にしたがって、 S_{op} を定義する。
9. S_{id} にしたがって、 S_{ip} を定義する。

3.3 セグメント間の依存関係に基づく波及解析

情報システム全体を整合的に変更するためには、セグメント間の相互作用を把握し、企業組織のある部分の変化による影響が、情報システム、および、企業組織の他の部分にどのように波及するかを解析する必要がある。 $C(b, r) \neq \emptyset$ であるセグメント $b (\in P_c \cup P_p)$ とセグメント $r (\in S)$ について、対 (b, r) を定義する。 b と r を頂点とし、 (b, r) を辺とするグラフは、セグメント間の依存関係を表現する。各辺に対応する $C(b, r)$ の要素に基づいて、相互に依存する業務モデルと要求モデルを対照させることによって、具体的な影響の内容を調べることができる。

3.4 データ集約図

$C(b, r)$ の要素数を $|C(b, r)|$ とする。セグメント間の依存関係を表現するグラフの各辺に、重み $1/|C(b, r)|$ を与える。このとき、セグメント間を結ぶ経路が含む辺の重みの総和の最小値をセグメント間の距離とみなし、距離の行列に対して多次元尺

度法を適用して作成した布置図が、データ集約図である。データ集約図は、企業組織と情報システムとの関係を概観することを目的とする。

4. 例題

前章で示した方法論にしたがって、人事情報システムを分析する例題を示す。人事管理に関わる職務を情報システム化した人事情報システムは、採用管理、就業管理、給与計算、人事評価、福利厚生、人材開発などのサブシステムから構成される [9]。

4.1 モデルの定義

本例題では、データ辞書を定義する上で重要な、Pc, Pp, Pu, Sid, Sod について説明する。

1. Pu の定義

情報システムから出力される情報を活用する現象をモデル化する。たとえば、

- 各社員に給与を支給し、明細を報告する。
- 所属別や期間別の勤怠状況を把握する。
- 組織を改編し、人事異動を行なう。

などの現象がある。情報活用上の便宜の観点から、関連する現象をセグメントとしてまとめる。本例題では、情報を活用するモデルのセグメントとして、< 給与計算 >, < 就業管理 >, < 人事評価 > を定義する。

2. Sod の定義

本例題では、Pu に属するセグメントに、解領域のセグメントが対応すると考えて、解領域のセグメントとして、< 給与計算サブシステム >, < 就業管理サブシステム >, < 人事評価サブシステム > を定義する。これらのセグメントごとに、要求される出力データ項目を決定する。たとえば、給与明細書 (給与計算サブシステム)、勤怠状況表 (就業管理サブシステム)、昇格/昇給/異動対象者一覧 (人事評価サブシステム) といった帳票のデータ項目を明らかにする。

3. Pp の定義

本例題では、情報を加工するモデルのセグメントとして、< 賃金体系 > を定義する。

4. Sid の定義

解領域のセグメントごとに、要求される出力データ項目を得るために必要な入力データ項目を決定する。これらのデータ項目は、入力画面で指定されるものとデータベースに格納されるものに大別される。また、運用後に賃金体系が変更

されることを考慮して、賃金体系に関するデータ項目を Sid として扱う。

5. Pc の定義

情報システムに入力する情報を収集する現象をモデル化する。たとえば、

- 人事記録表に記入する。
- 出勤簿に記入する。
- 人事考課表に記入する。
- 自己申告書に記入する。

などの現象がある。情報収集上の便宜の観点から、関連する現象をセグメントとしてまとめる。本例題では、情報を収集するモデルのセグメントとして、< 基本属性 >, < 勤怠状況 >, < 人事考課 >, < 自己申告 > を定義する。

4.2 データ辞書の定義

前節で定義したセグメントごとに、データ辞書に収録する語を選び出し、問題領域のセグメントと解領域のセグメントの対について、共通語彙を定義する。データ辞書の例を以下に示す。

C(< 基本属性 >, < 給与計算サブシステム >) = {
社員番号, 氏名, 所属,
役職, 給与年齢, 最終学歴,
健康保険証番号, 雇用保険番号, 厚生年金番号,
住居区分, 通勤経路, 配偶者,
家族状況 }

C(< 基本属性 >, < 就業管理サブシステム >) = {
社員番号, 氏名, 所属,
勤務地 }

C(< 基本属性 >, < 人事評価サブシステム >) = {
社員番号, 氏名, 所属,
勤務地, 職務, 役職,
性別, 年齢, 勤続年数 }

C(< 勤怠状況 >, < 給与計算サブシステム >) = {
社員番号, 氏名, 所属,
年, 月, 残業時間数,
欠勤日数, 遅刻日数, 早退日数 }

C(< 勤怠状況 >, < 就業管理サブシステム >) = {
社員番号, 氏名, 所属,
勤務地, 年, 月,
残業時間数, 有給休暇日数, 特別休暇日数,
振替出勤日数, 振替休日数, 欠勤日数,
遅刻日数, 早退日数 }

C(< 勤怠状況 >, < 人事評価サブシステム >) = {
社員番号, 氏名, 所属,
勤務地 }

C(< 人事考課 >, < 給与計算サブシステム >) = {
社員番号, 氏名, 所属,
人事考課 }

C(<人事考課>, <就業管理サブシステム>) = {
社員番号, 氏名, 所属 }

C(<人事考課>, <人事評価サブシステム>) = {
社員番号, 氏名, 所属,
職務, 速度, 持続性,
正確さ, 丁寧さ, 協調性,
積極性, 規律性, 責任感,
勤怠状況, 服装, 職務知識,
関連知識, 洞察力, 折衝力,
表現力, 実績, 態度,
能力, 人事考課, 評価者 }

C(<自己申告>, <給与計算サブシステム>) = {
社員番号, 氏名, 所属,
資格 }

C(<自己申告>, <就業管理サブシステム>) = {
社員番号, 氏名, 所属 }

C(<自己申告>, <人事評価サブシステム>) = {
社員番号, 氏名, 所属,
仕事の量, 仕事の難易度, 希望勤務地,
希望職務, 資格, 特技,
趣味 }

C(<賃金体系>, <給与計算サブシステム>) = {
基本給, 残業手当, 役職手当,
資格手当, 家族手当, 住宅手当,
通勤手当, 役職, 給与年齢,
最終学歴, 住居区分, 通勤経路,
配偶者, 家族状況, 残業時間数,
欠勤日数, 遅刻日数, 早退日数,
人事考課, 資格 }

C(<賃金体系>, <就業管理サブシステム>) = {
残業時間数, 欠勤日数, 遅刻日数,
早退日数 }

C(<賃金体系>, <人事評価サブシステム>) = {
役職, 人事考課, 資格 }

4.3 環境の変化への適応

本節では、環境の変化に適応するように、モデルと語彙を変更する手順を説明する。

変化-1: 職能制度の導入

個人の能力を客観的に評価するために、職能体系を規定する。

- (1) 情報を活用するモデルである<人事評価>を変更する。
- (2) 問題領域に、情報を加工するモデルである<職能体系>を追加する。<職能体系>の語彙には、“職務知識”、“関連知識”、“洞察力”、“折衝力”、“表現力”、“職務”、“職能”が含まれる。
- (3) <人事評価サブシステム>を変更する。“職能”、“滞留年数”を語彙に追加する。

(4) <人事評価サブシステム>の変更から波及して、情報を収集するモデルである<人事考課>を変更する。“職能”、“滞留年数”を語彙に追加する。

(5) <人事評価サブシステム>の変更から波及して、情報を収集するモデルである<基本属性>を変更する。“職能”を語彙に追加する。

(6) <職能体系>の追加から波及して、<給与計算サブシステム>を変更する。“基本給”を削除し、“年齢給”、“職能給”、“職能”を追加する。

(7) <給与計算サブシステム>の変更から波及して、情報を加工するモデルである<賃金体系>を変更する。“基本給”を削除し、“年齢給”、“職能給”、“職能”を追加する。

変化-2: 査定方式の改善

人事考課の結果をより適切に賞与に反映させるために、より緻密な査定方式を採用する。

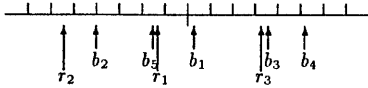
- (1) 情報を加工するモデルである<賃金体系>を変更する。“人事考課”を削除し、“実績”、“態度”、“能力”を追加する。
- (2) <賃金体系>の変更から波及して、<給与計算サブシステム>を変更する。“人事考課”を削除し、“実績”、“態度”、“能力”を追加する。

変化-3: 資格の多様化

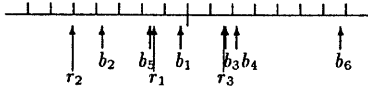
事業の拡大によって、必要な資格が増加する。

- (1) 情報を活用するモデルである<人事評価>を変更する。
- (2) <人事評価サブシステム>を変更する。“資格”を削除し、“資格-1”~“資格-8”を追加する。
- (3) <人事評価サブシステム>の変更から波及して、情報を収集するモデルである<自己申告>を変更する。“資格”を削除し、“資格-1”~“資格-8”を追加する。
- (4) <人事評価サブシステム>の変更から波及して、情報を加工するモデルである<賃金体系>を変更する。“資格”を削除し、“資格-1”~“資格-8”を追加する。
- (5) <賃金体系>の変更から波及して、<給与計算サブシステム>を変更する。“資格”を削除し、“資格-1”~“資格-8”を追加する。

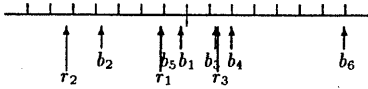
1. 変更前



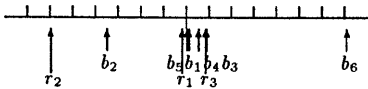
2. 職能制度の導入による変更後



3. 査定方式の改善による変更後



4. 資格の多様化による変更後



- b₁ : < 基本属性 >
- b₂ : < 勤怠状況 >
- b₃ : < 人事考課 >
- b₄ : < 自己申告 >
- b₅ : < 賃金体系 >
- b₆ : < 職能体系 >
- r₁ : < 給与計算サブシステム >
- r₂ : < 就業管理サブシステム >
- r₃ : < 人事評価サブシステム >

図 3: 人事情報システムの進化の過程

4.4 情報システムの進化

前節における、職能制度の導入→査定方式の改善→資格の多様化という環境の変化は、能力主義の人事制度への移行を意味する。このような企業組織の進化は、データ集約図によって、情報システムの進化との関係から捉えることができる。図3は、環境の変化に対して変更を行なったデータ辞書について、データ集約図を時系列に並べたものである。データ集約図の変化から、以下のことが導かれる。

- 変更前には独立した位置にあった< 基本属性 >、< 人事考課 >、< 自己申告 >、< 賃金体系 >が、< 給与計算サブシステム >と< 人事評価サブシステム >の関係が緊密化するにつれて、次第に接近し、最終的に統合される。能力主義の人事制度への移行を期待しているならば、望ましい方向への進化であるといえる。したがって、この分析を情報システムを変更する前に行なえば、能力主義の人事制度への移行が、どのような変更によって実現されるかを予測できる。
- < 職能体系 >は、能力主義の人事制度に向け

ての、情報システムの進化をもたらす。しかし、< 職能体系 >そのものは、いわば触媒的な存在であって、情報システムの進化の影響を受けずに、安定した位置を保っている。

- < 勤怠状況 >と< 就業管理サブシステム >については、ほとんど変化が認められない。したがって、安定した業務でありアプリケーションシステムであると判断されるが、場合によっては、変化を促すための何らかの対策が必要である。

5. 関連研究

要求工学におけるドメインモデルは、情報システムと相互作用する対象のモデルである[2]。情報システムの要求仕様は、その情報システムを利用する組織の業務に関する知識を表現するドメインモデルに基づいて定義されなければならない[7]。本研究における業務モデルは、この意味でのドメインモデルに相当し、要求者と分析者の間のコミュニケーションの基盤となる[8]。

要求分析を含むソフトウェア開発の諸フェーズに適用できるドメインモデルとして、問題群を表現する Domain Problem Model、解群を表現する Domain Product Model、問題群から解群を導出する手順を表現する Domain Process Model から構成される、Triadic Domain Model が提案されている[5]。これを要求分析に適用する場合、Domain Problem Model は、実世界における業務のインフォーマルな表現のひな形であり、Domain Product Model は、フォーマルに記述したソフトウェアの要求仕様のひな型である。本研究では、業務モデルが Domain Problem Model に相当し、要求モデルが Domain Product Model に相当する。しかし、データ辞書の構築と再編を支援する要求分析の方法論は、問題領域と解領域の間のギャップを埋めるためのものであるが、問題群から解群を導出する手順にはなっていない。

多くの経営情報システムは、企業の各部門ごとに独立して開発されてきたために、異なる組織単位間での協調的問題解決が困難になっている。一貫性を欠いたシステム群を統合するためには、組織単位ごとの業務知識を共有するためのオントロジを整備しなければならない。たとえば、TOVE プロジェクトでは、異なる組織単位間で利用可能な共通の用語の集まりを定め、各用語に正確で曖昧さのない意味を与えることを目標とする[3]。本研究におけるドメインモデルでは、各要求モデルが各業務モデルによっ

て表現される知識を共有すると同時に、各業務モデルが各要求モデルによって表現される知識を共有する。これによって、協調的問題解決が可能になる。

6. おわりに

本稿では、情報資源管理の考え方を前提とする経営情報システムのドメインモデルに基づく要求分析の方法論を提案した。経営情報システムの分析では、大規模なメタデータベースを環境の変化に適應させるために、企業組織のある部分の変化による影響がどのように波及するかを捉えること、および、情報システムの進化を企業組織の進化の観点から捉えることが重要である。本方法論がこれらを支援することを、例題によって示した。

しかし、本例題は、文献から得た知識に基づいて、机上で分析したものである。したがって、今後の課題は、現実的なソフトウェア開発プロジェクトにおいて、本方法論の適用を試みていくことである。そのためには、本方法論の適用を支援する環境を開発技術者に提供する必要がある。また、業務モデルの定義には、かなりの手間を要すると考えられる。そこで、特定の業務(たとえば、在庫管理業務など)への適用を積み重ねた経験に基づいて、当該業務を情報システム化するための標準的な業務モデルを定義し、これを再利用できるようにしていきたい。

謝辞

本研究は、産業科学技術研究開発制度「新ソフトウェア構造化モデルの研究開発」の一環として情報処理振興事業協会(IPA)が新エネルギー・産業技術総合開発機構から委託をうけて実施したものである。

参考文献

- [1] Arango,G. and Prieto-Diaz,R.: Introduction and Overview: Domain Analysis Concepts and Research Directions, in Prieto-Diaz,R. and Arango,G. (eds.), *Domain Analysis and Software Systems Modeling*, IEEE Computer Society Press, 1991.
- [2] Finkelstein,A.: Requirements Engineering: A Review and Research Agenda, *Proceedings of the First Asia-Pacific Software Engineering Conference (APSEC'94)*, 1994.
- [3] Fox,M.S.: The TOVE Project: Towards a Common-Sense Model of the Enterprise, in Petrie,C.J. (ed.), *Enterprise Integration Modeling: Proceedings of the First International Conference*, MIT Press, 1992.
- [4] 堀内一: データ中心システム設計, オーム社, 1988.
- [5] 伊藤潔, 田村恭久, 杵島修三: Triadic Domain Model に基づくシステムの分析・設計, 第1回 ソフトウェア工学の基礎 ワークショップ (FOSE'94) 論文集, 1994.
- [6] Leong-Hong,B.W. and Plagman,B.K.: *Data Dictionary / Directory Systems*, John Wiley, 1982.
- [7] 大槻繁: アプリケーション・ソフトウェア・システムの要求仕様決定方法, 情報処理学会研究会報告, 94-SE-96-13, 1994.
- [8] 斉藤康彦, 本位田真一: オブジェクト指向モデルを用いた要求獲得の支援, 情報処理学会論文誌, Vol.35, No.11, 1994.
- [9] 下崎千代子: 人事情報システム, 日科技連出版社, 1993.