

企業情報システムの要求分析のためのドメインモデル

斉藤 康彦^{†,☆} 本位田 真一^{†,☆☆}

ドメインモデルの観点から、企業情報システムの要求分析法を考察する。企業情報システムの要求分析では、1) 企業情報システムの特性をふまえて、ソフトウェアの開発方法論が適用されていないこと、および、2) 企業活動の改善に関する分析の視点とソフトウェアの開発に関する分析の視点との間に存在するギャップが問題になる。本論文では、これらの問題を解決するために、まず、企業内の業務を記述するための語の集合と、業務を支援する情報システムを構成するソフトウェアを記述するための語の集合に基づいて、企業情報システムの特性を表現するドメインモデルを定義する。次に、本モデルが第1の問題の解決に寄与することを示すために、本モデルとデータ中心アプローチ (DOA) における分析の手順の関係を説明する。次に、本モデルが第2の問題の解決に寄与することを示すために、本モデルに基づく業務分析技法を提示する。本技法と DOA は、共通のモデルによって表現される企業情報システムの特性に関係付けられるので、DOA を実践する中で、本技法を適用することができる。さらに、本技法によって、企業活動における改善点に対する改善案の妥当性を確認できることを明らかにする。

A Domain Model for Requirements Analysis of Business Information Systems

YASUHIKO SAITO^{†,☆} and SHINICHI HONIDENT^{†,☆☆}

We discuss requirements analysis methods for business information systems (BISs) from the viewpoint of domain models. The following are problems in requirements analysis of BISs: 1) Applying a software development method without taking features of BISs into consideration. 2) A gap between analysis for improving business activities and analysis for software development. To solve these problems, we define a domain model representing features of BISs on the basis of sets of words used in descriptions of business and in descriptions of software components of BISs. To show that the domain model contributes towards solving the first problem, we explain relations between the domain model and the analysis process in data oriented approach (DOA). To show that the domain model contributes towards solving the second problem, we present an analysis method for improving business activities. The method is based on the domain model. Because both the method and the analysis process in DOA are related to the common domain model representing features of BISs, one can use the method in DOA development. Also, the method allows one to validate plans for improving business activities.

1. はじめに

企業情報システムの要求分析に関して、しばしば、以下の問題が指摘される。

- (1) 構造化分析 (SA)³⁾、データ中心アプローチ (DOA)^{7),11)}、オブジェクト指向分析 (OOA)^{1),15)}、ジャクソン・システム開発法

(JSD)⁹⁾ などの開発方法論が提案され、有効性が実証されてきたが、実際の開発プロジェクトに適用しようとする、必ずしもうまくいかない。

- (2) 事務作業の流れを表す図や伝票の経路を表す図などを用いた業務分析は、企業活動において改善すべき点を検出し、それに対する改善案を検討するうえで有効であるが、ソフトウェアの開発方法論を実践する中で、その分析結果を活かすことが難しい。

第1の問題の原因は、企業情報システムの特性をふまえて、ソフトウェアの開発方法論が適用されていないことであると考えられる。そこで、本研究では、企

[†] 情報処理振興事業協会 (IPA)
Information-technology Promotion Agency, Japan

[☆] 現在、株式会社アイネス
Presently with INES Corp.

^{☆☆} 現在、株式会社東芝
Presently with Toshiba Corp.

業情報システムの特性を明らかにし、さらに、この特性と開発方法論の関係を明らかにする。

第2の問題の原因は、企業活動の改善に関する分析の視点とソフトウェアの開発に関する分析の視点との間に存在するギャップであると考えられる。そこで、本研究では、企業情報システムの特性と開発方法論の関係を明らかにしたうえで、この特性に基づいて、企業活動における改善点に対する改善案を検討するための業務分析技法を開発する。これによって、業務分析技法と開発方法論は、企業情報システムの特性に関係付けられるので、視点間のギャップが小さくなる。

企業情報システムの特性を明らかにするために、いくつかの企業情報システムを開発した経験から得られた知識を整理して、一般化した。これを企業情報システムの特性を表現するドメインモデルとして提案する。ドメインモデルとは、対象システム自身が本来持つ各種の性質や開発上の多様な知識を分析することによって得られた、共通の対象領域（ドメイン）に属する、用語、問題のとらえ方、システムの構造、システムの作り方等の、固有な概念構造である¹⁸⁾。本論文で提案するドメインモデルは、過去に要求分析を行った経験から得られた知識を再利用することによって、特定のシステムの要求分析に際して、上に述べた問題の解決を図るものである。したがって、特定のシステムについて要求分析を行った結果としての要求仕様や、複数のシステムの要求仕様を標準化したものではない。

本論文の2章では、このような企業情報システムのドメインモデルを定義する。本ドメインモデルは、企業内の業務を記述するための語の集合と、業務を支援する情報システムを構成するソフトウェアを記述するための語の集合に基づいて、企業情報システムの特性を表現する。3章では、本ドメインモデルとDOAにおける分析の手順の関係を考察する。企業情報システムのための開発方法論は、DOAに限定されるものではないが、本ドメインモデルは、DOAの考え方を参考にして定義されている。4章では、本ドメインモデルに基づく業務分析技法の例を示す。本技法は、DOAを実践する中で適用していくことが可能であり、情報システムの改善による企業活動の改善を支援する。本技法によって、企業活動における改善点に対する改善案の妥当性を確認することができる。5章では、本研究におけるドメインモデルの役割をより明確にするために、他の研究におけるドメインモデルとの差異について議論する。

2. ドメインモデル

本章では、企業情報システムの要求分析のためのドメインモデルを定義する。

2.1 業務記述

業務記述は、企業組織の多種多様な業務の内容を業務単位ごとに記述したものである。特に、情報システムが導入される業務において、人間がそのシステムを利用して行う作業が記述の対象となる。業務記述の目的は、業務と情報システムの境界を規定し、情報システムが業務の中でどのように利用されるべきかを明確にすることである。本ドメインモデルでは、業務記述の表記法を特に限定しない。しかし、業務記述の中で用いられるデータ項目とそれらに関する作業の名称が明示されなければならない。

業務集合 B は、業務記述の集合である。 n 個の業務単位のためのシステムを開発する場合、各業務単位についての業務記述を $b_1, b_2, b_3, \dots, b_n$ とすると、

$$B = \{b_1, b_2, b_3, \dots, b_n\}$$

である。

業務記述には、以下の種類がある。

(1) 情報の収集に関する記述

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プロセスを記述したもの。業務集合 B における情報の収集に関する記述の集合を B_c とする。

(2) 情報の加工に関する記述

原価計算規則、在庫管理規則、部品構成表、賃金体系など、情報を加工するための定常的な業務規則を記述したもの。業務集合 B における情報の加工に関する記述の集合を B_p とする。

(3) 情報の活用に関する記述

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、販売予測、生産計画などの意思決定を行うプロセス、および、請求書、領収書、財務諸表、給与明細などによって、報告を行うプロセスを記述したもの。業務集合 B における情報の活用に関する記述の集合を B_u とする。

B, B_c, B_p, B_u の間には、以下の関係がある。

$$B = B_c \cup B_p \cup B_u$$

$$B_c \cap B_p = \phi$$

$$B_p \cap B_u = \phi$$

$$B_u \cap B_c = \phi$$

2.2 ソフトウェア記述

ソフトウェア記述は、業務を支援する情報システム

を構成するソフトウェアの機能をサブシステムごとに記述したものである。ソフトウェア記述の目的は、業務記述によって定義される業務の中で情報システムが利用されるために、情報システムがどのような機能を提供すべきかを明確にすることである。本ドメインモデルでは、ソフトウェア記述の表記法を特に限定しない。しかし、ソフトウェア記述の中で用いられるデータ項目とそれらに対する操作の名称が明示されなければならない。また、記述の抽象レベルとしては、ソースコードや内部仕様書のレベルではなくて、外部仕様書のレベルを目安とする。

ソフトウェア集合 S は、ソフトウェア記述の集合である。 n 個のサブシステムから構成されるシステムを開発する場合、各サブシステムについてのソフトウェア記述を $s_1, s_2, s_3, \dots, s_n$ とすると、

$$S = \{s_1, s_2, s_3, \dots, s_n\}$$

である。

各ソフトウェア記述は、以下の部分から構成される。

(1) 蓄積情報記述

蓄積情報とは、企業活動のための資源としての情報を蓄積する処理に関連する情報である。蓄積情報には、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）、データベースなどに保持されるデータ項目（マスタ情報）、および、業務規則に関連するデータ項目（ルール情報）がある。蓄積情報記述は、蓄積情報について、データ項目の属性やデータ項目間の静的な関係（データベースのスキーマや画面のレイアウトなど）を明確にする。ソフトウェア記述 s についての蓄積情報記述を $s.id$ とする。 $s.id$ には、 s が記述の対象とするサブシステムで扱われるすべての蓄積情報を記述する。

(2) 蓄積処理記述

蓄積処理とは、企業活動のための資源としての情報を蓄積する処理、すなわち、蓄積情報に関する手続きである。蓄積処理記述は、蓄積情報記述におけるトランザクション情報/マスタ情報/ルール情報を区別し、これらの間の関係を明確にする。ソフトウェア記述 s についての蓄積処理記述を $s.ip$ とする。 $s.ip$ には、 s が記述の対象とするサブシステムで扱われるすべての蓄積情報に関する手続きを記述する。

(3) 生成情報記述

生成情報とは、資源として蓄積された情報から新たに生成される情報である。生成情報には、画面や帳票などに出力されるデータ項目がある。

生成情報記述は、生成情報について、データ項目の属性やデータ項目間の静的な関係（画面や帳票のレイアウトなど）を明確にする。情報の活用に関する記述は、これらのデータ項目の基本的な意味や用法を説明する。ソフトウェア記述 s についての生成情報記述を $s.od$ とする。 $s.od$ には、 s が記述の対象とするサブシステムで扱われるすべての生成情報を記述する。

(4) 生成処理記述

生成処理とは、資源として蓄積された情報から新たな情報を生成する処理、すなわち、生成情報に関する手続きである。生成処理記述は、蓄積情報と生成情報の間の関係を明確にする。これには、照会機能や帳票印刷機能などの記述が含まれる。ソフトウェア記述 s についての生成処理記述を $s.op$ とする。 $s.op$ には、 s が記述の対象とするサブシステムで扱われるすべての生成情報に関する手続きを記述する。

ソフトウェア記述 s は、組 $(s.id, s.ip, s.od, s.op)$ で表される。

2.3 語の集合

集合 $V(b), V(s.id), V(s.ip), V(s.od), V(s.op)$ を以下のように定義する。

$V(b)$

業務記述 b の中で用いられる語の集合。各語は、業務で扱うデータ項目の名称、および、データ項目に関する作業の名称である。

$V(s.id)$

蓄積情報記述 $s.id$ の中で用いられる語の集合。各語は、情報システムが扱うデータ項目の名称である。

$V(s.ip)$

蓄積処理記述 $s.ip$ の中で用いられる語の集合。各語は、情報システムが扱うデータ項目の名称、および、データ項目に対する操作の名称である。

$V(s.od)$

生成情報記述 $s.od$ の中で用いられる語の集合。各語は、情報システムが扱うデータ項目の名称である。

$V(s.op)$

生成処理記述 $s.op$ の中で用いられる語の集合。各語は、情報システムが扱うデータ項目の名称、および、データ項目に対する操作の名称である。

したがって、業務集合とソフトウェア集合に属するすべての業務記述とソフトウェア記述の中で用いられる語の集合 V は、以下の集合の和となる。

$$V_c = \bigcup_{b \in B_c} V(b)$$

$$V_p = \bigcup_{b \in B_p} V(b)$$

$$V_u = \bigcup_{b \in B_u} V(b)$$

$$V_{id} = \bigcup_{s \in S} V(s.id)$$

$$V_{ip} = \bigcup_{s \in S} V(s.ip)$$

$$V_{od} = \bigcup_{s \in S} V(s.od)$$

$$V_{op} = \bigcup_{s \in S} V(s.op)$$

2.4 語の集合における制約

語の集合は、以下の制約を満足する必要がある。

制約 1

すべての蓄積情報は、情報の収集や加工のプロセスで参照される。したがって、 V_c , V_p , V_{id} , $V(b)$, $V(s.id)$ の定義より、以下の関係が成立する。

$$V_{id} \subseteq V_c \cup V_p$$

制約 2

すべての生成情報は、情報の活用プロセスで参照される。したがって、 V_u , V_{od} , $V(b)$, $V(s.od)$ の定義より、以下の関係が成立する。

$$V_{od} \subseteq V_u$$

制約 3

すべての蓄積情報は、蓄積処理で参照される。したがって、 V_{id} , V_{ip} , $V(s.id)$, $V(s.ip)$ の定義より、以下の関係が成立する。

$$V_{id} \subseteq V_{ip}$$

制約 4

すべての生成情報は、生成処理で参照される。したがって、 V_{od} , V_{op} , $V(s.od)$, $V(s.op)$ の定義より、以下の関係が成立する。

$$V_{od} \subseteq V_{op}$$

3. ドメインモデルと開発方法論の関係

本章では、企業情報システムのドメインモデルと DOA における分析の手順の関係を考察する。

2章で示した語の集合における制約から、業務記述とソフトウェア記述の変更（追加や削除を含む）について、以下の規則が導かれる。

規則 1

制約 1 より、情報の収集に関する記述、または、情報の加工に関する記述で用いる語の変更によって、

蓄積情報記述で用いる語の変更が必要になることがある。変更が波及しないのは、 $(V_c \cup V_p) - V_{id}$ に属する語を変更する場合である。

規則 2

制約 1 より、蓄積情報記述で用いる語の変更によって、情報の収集に関する記述、または、情報の加工に関する記述で用いる語の変更がつねに必要なになる。

規則 3

制約 2 より、情報の活用に関する記述で用いる語の変更によって、生成情報記述で用いる語の変更が必要になることがある。変更が波及しないのは、 $V_u - V_{od}$ に属する語を変更する場合である。

規則 4

制約 2 より、生成情報記述で用いる語の変更によって、情報の活用に関する記述で用いる語の変更がつねに必要なになる。

規則 5

制約 3 より、蓄積処理記述で用いる語の変更によって、蓄積情報記述で用いる語の変更が必要になることがある。変更が波及しないのは、 $V_{ip} - V_{id}$ に属する語を変更する場合である。

規則 6

制約 3 より、蓄積情報記述で用いる語の変更によって、蓄積処理記述で用いる語の変更がつねに必要なになる。

規則 7

制約 4 より、生成処理記述で用いる語の変更によって、生成情報記述で用いる語の変更が必要になることがある。変更が波及しないのは、 $V_{op} - V_{od}$ に属する語を変更する場合である。

規則 8

制約 4 より、生成情報記述で用いる語の変更によって、生成処理記述で用いる語の変更がつねに必要なになる。

一般に、要求される蓄積情報は、要求される生成情報に依存して決定するので、生成情報記述で用いる語の変更によって、蓄積情報記述で用いる語の変更が必要になる場合がある。その場合には、たとえば、情報の活用に関する記述で用いる語が変更されると、規則 3 より、生成情報記述で用いる語の変更が必要になることがあり、これによって、蓄積情報記述で用いる語の変更が必要になる。ここで、蓄積情報記述で用いる語が変更されると、規則 2 より、情報の収集に関する記述、または、情報の加工に関する記述で用いる語の変更が必要になり、規則 6 より、蓄積処理記述で用い

る語の変更が必要になる。

以上の考え方は、要求の変更に対する波及解析の手順を規定するが、要求を変更する場合だけでなく、新たに要求を定義する場合にも適用できる。なぜならば、新たな要求の定義は、追加を中心とする要求の変更としてとらえられるからである。

前章のドメインモデルから導かれたこれらの規則に基づいて、企業情報システムの特性と DOA における分析の手順の関係を説明する。DOA の開発方法論は、これまでいくつか提案されているが、ここでは、以下の分析の手順⁷⁾について考える。

- (1) 業務プロセスモデリング
データの発生、変更、消滅のライフサイクルを支配する業務をモデル化する。
- (2) データの標準化
データ名称と形式、データ項目の内容記述、データ正規化、データの分類体系について標準を設定する。
- (3) データライフサイクル処理設計
データごとに、その値を決定するプロセスを再配置する。

業務プロセスモデリングでは、 Bc , Bp , Bu を決定し、それに基づいて、 Vc , Vp , Vu を決定する。データの標準化では、 Vid , Vod を決定する。データライフサイクル処理設計では、 Vip , Vop を決定し、それに基づいて、 S を決定する。ここで、業務プロセスモデリングを第 1 のステップにすると、規則 1 と規則 3 より、データの標準化が第 2 のステップになり、さらに、規則 6 と規則 8 より、データライフサイクル処理設計が第 3 のステップになる。これが基本的な順序であるが、通常は、手戻りがある。すなわち、規則 2 と規則 4 より、データの標準化の結果が業務プロセスモデリングに影響を及ぼす場合や、規則 5 と規則 7 より、データライフサイクル処理設計の結果がデータの標準化に影響を及ぼす場合がある。

このように、ドメインモデルと分析の手順の関係が明示されることによって、実際の開発プロジェクトへの手順の適用が容易になると考えられる。特に、DOA では、規則 1～規則 8 が明示されていないが、要求の変更が多い開発プロジェクトでは、これらの規則によって、手順の柔軟な適用が可能になる。また、著者のひとは、企業情報システムの開発に DOA を適用する理由を、社内の開発技術者に理解してもらうために、本ドメインモデルを用いて、企業情報システムの特性と DOA の関係を説明することを試みている。

4. ドメインモデルに基づく業務分析技法

本章では、企業情報システムのドメインモデルに基づく業務分析技法が、DOA を実践する中で適用していくことが可能であること、そして、本技法によって、企業活動における改善点に対する改善案の妥当性を確認できることを示す。

4.1 業務記述とソフトウェア記述の間の距離

はじめに、2 章で示した語の集合に基づいて、業務記述 b とソフトウェア記述 s の間の距離 $d(b, s)$ を、以下のように定義する。

$$d(b, s) = 1/|V(b) \cap V(s.id)|$$

ただし、 $b \in Bc \cup Bp$, $s \in S$ である。 $|V(b) \cap V(s.id)|$ は、共通集合 $V(b) \cap V(s.id)$ の要素の数であり、業務記述 b とソフトウェア記述 s の間の関連性の強さを表す。そこで、業務記述 b とソフトウェア記述 s の間の距離を、その逆数として定義する。

DOA では、データ項目の名称を統一することが重要であるが、業務記述とソフトウェア記述の間の距離を適切に設定するには、やはり、データ項目の名称を統一する必要がある。これは、次の理由による。 $d(b, s)$ を適切に設定するためには、任意の s_1 と s_2 について、 $V(b) \cap V(s_1.id)$ と $V(b) \cap V(s_2.id)$ を適切に決定しなければならない。したがって、 $V(b)$, $V(s_1.id)$, $V(s_2.id)$ は、以下を満足する必要がある。

- (1) $V(s_1.id)$ に属する語 w_1 と $V(s_2.id)$ に属する語 w_2 が、業務上の同じ対象を指示するならば、 $V(b)$ に属する語 w_3 について、 $w_1 = w_3$, かつ、 $w_2 = w_3$ である。このとき、 $w_1 = w_2$ となるので、 w_1 と w_2 は、同じ名称のデータ項目である。すなわち、業務上の同じ対象を指示するデータ項目の名称が同じであることが保証されている (synonym の解決)。
- (2) $V(s_1.id)$ に属する語 w_1 と $V(s_2.id)$ に属する語 w_2 が、業務上の異なる対象を指示するならば、 $V(b)$ に属する語 w_3 と w_4 について、 $w_1 = w_3$, かつ、 $w_2 = w_4$, かつ、 $w_3 \neq w_4$ である。このとき、 $w_1 \neq w_2$ となるので、 w_1 と w_2 は、異なる名称のデータ項目である。すなわち、業務上の異なる対象を指示するデータ項目の名称が異なることが保証されている (homonym の解決)。

これらを満足することは、データ項目の名称を統一することである。

また、DOA では、名称を統一するデータ項目として、特に、複数のサブシステムの間で共有されるデー

データベースに格納するデータ項目が重要になるが、これらのデータ項目の名称は、蓄積情報記述、および、 $V(s.id)$ の定義より、 $V(s.id)$ に属する語である。さらに、これらの語は、語の集合における制約 1 より、情報の収集に関する記述、または、情報の加工に関する記述で用いられているので、業務記述 b は、 Bc 、または、 Bp に属する。したがって、 $b \in Bc \cup Bp$ と $s.id$ に基づいて、 $d(b, s)$ を定義している。

4.2 手順

距離の概念に基づく業務分析の手順を以下に示す。

- (1) 距離を定義できるすべての業務記述 b とソフトウェア記述 s について、 b と s を頂点とし、対 (b, s) を辺とするグラフを描く。このグラフの各辺に距離 $d(b, s)$ を与える。
- (2) 頂点と頂点を結ぶ経路が含む辺に与えられた距離の総和の最小値を、頂点間の距離と見なす。グラフのすべての辺に対応する距離の行列に、多次元尺度法⁵⁾を適用して、頂点、すなわち、業務記述とソフトウェア記述の布置図を描く。
- (3) ある一定の意図に基づいて、業務記述とソフトウェア記述を少しずつ変更していく。各変更に対して、布置図を描き直す。ここで、ある一定の意図とは、たとえば、「在庫量を減少させるために、生産業務と販売業務をリアルタイムに連携させる」、「管理部門の規模を縮小するために、管理部門の機能の一部を個別の生産部門に委譲する」といった企業活動の改善のための意図である。また、業務記述とソフトウェア記述は、たとえば、伝票への記入欄の追加、発注方式の変更、課の統合などともなっており変更される。この変更によって、語の追加、削除、置換といった、局所的な変化が生じる。
- (4) これらの布置図を時系列に並べる。布置図上の点間の距離は、それらの点に対応する業務記述やソフトウェア記述の間の関連性の強さを視覚的に表現する。したがって、時系列に並べた布置図から、以下のような、業務記述とソフトウェア記述における関連性の時系列の変化をとらえることができる。

- 点のクラスタが次第に凝集する、すなわち、そのクラスタに含まれる業務記述とソフトウェア記述における関連性が強まりつつある。これは、それらの業務記述とソフトウェア記述に対応する業務単位とサブシステムの間関係が、密になりつつあることを意味する。

- 点のクラスタが次第に拡散する、すなわち、そのクラスタに含まれる業務記述とソフトウェア記述における関連性が弱まりつつある。これは、それらの業務記述とソフトウェア記述に対応する業務単位やサブシステムの間関係が、疎になりつつあることを意味する。
- 点のクラスタの凝集や拡散から離れて動かない点がある、すなわち、その点に対応する業務記述やソフトウェア記述が他から孤立しつつある。これは、その業務記述やソフトウェア記述に対応する業務単位やサブシステムが、組織の中で孤立しつつあることを意味する。

これらは、業務記述とソフトウェア記述における局所的な変化によって生じる大局的な変化であり、たとえば、「生産部門、販売部門、生産管理システム、販売管理システムの間関係が、密になりつつある」といった変化である。

- (5) こうした大局的な変化が、局所的な変化を生じさせる動機になった本来の意図、たとえば、「在庫量を減少させるために、生産業務と販売業務をリアルタイムに連携させる」といった意図に適合しているかどうかを評価する。適合している場合は、その業務記述とソフトウェア記述の変更が妥当であると判断され、適合していない場合は、妥当でないと判断される。

4.3 特徴

本技法の特徴は、以下のように整理される。

- (1) 業務とそれを支援する情報システムの局所的な変化によって生じる、企業組織における大局的な変化を、視覚的に把握することができる。
- (2) 業務とそれを支援する情報システムを、相互に関連するものとして把握することができる。
- (3) 企業活動を期待する方向に変化させるために、情報システムをどのように変更すべきかについて、いくつかの改善案がある場合に、それらの妥当性を確認することができる。一方、そのような改善案が何もない状況では、本技法を適用することができない。
- (4) DOA における分析の手順と交互に反復して適用することができる。すなわち、まず、DOA における分析の手順にしたがって、業務記述とソフトウェア記述を作成する。次に、作成した業務記述とソフトウェア記述に対して本技法を適用する。さらに、本技法を適用した結果を反

映させるために、DOA における分析の手順にしたがって、業務記述やソフトウェア記述を変更する。

なお、本技法を適用する際に注意すべき点として、記述の粒度の問題がある。すなわち、業務集合とソフトウェア集合に属する業務記述とソフトウェア記述における記述の粒度は、統一されていなければならない。なぜならば、記述の粒度にばらつきがあると、業務記述とソフトウェア記述の間の距離が適切に設定されないからである。また、本技法では、データ項目の名称を統一するために、synonym と homonym の解決を前提とするが、そのためにも、記述の粒度の統一が重要である。

企業活動の改善を支援する本技法は、ドメインモデルに基づくものである。前章では、DOA における分析の手順とドメインモデルの関係を示した。これによって、企業活動の改善を支援する技法と DOA における分析の手順は、共通のドメインモデルに関係付けられたことになる。したがって、このようなドメインモデルには、企業活動の改善に関する分析の視点とソフトウェアの開発に関する分析の視点との間に存在するギャップを小さくする役割があるといえる。

4.4 例題

本技法を人事情報システム¹⁷⁾の要求分析に適用する例題によって、企業活動における改善点に対する改善案の妥当性を確認するうえで、本技法が有効であることを示す。

B_c , B_p , S を以下のように与える。

$$B_c = \{b_1, b_2, b_3, b_4\}$$

$$B_p = \{b_5\}$$

$$S = \{s_1, s_2, s_3\}$$

ただし、 b_1 は人事記録表に記入する業務の記述、 b_2 は出勤簿に記入する業務の記述、 b_3 は人事考課表に記入する業務の記述、 b_4 は自己申告書に記入する業務の記述、 b_5 は賃金体系の記述、 s_1 は給与計算サブシステムの記述、 s_2 は就業管理サブシステムの記述、 s_3 は人事評価サブシステムの記述である。これらの業務記述とソフトウェア記述の間の距離を設定するための語の集合を図 1 に示す。

ここで、「能力主義の人事制度に移行する」という意図に基づいて、以下の順に、業務記述とソフトウェア記述を変更していく。

(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。この変更によって、業務記述 b_6 として、職能体系の記述が B_p に追加される。

$V(b_1) \cap V(s_1.id) = \{$	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 }			
$V(b_1) \cap V(s_2.id) = \{$	社員番号,	氏名,	所属 }	
$V(b_1) \cap V(s_3.id) = \{$	社員番号,	氏名,	所属,	勤務地,
	職務,	役職,	性別,	年齢,
	勤続年数 }			
$V(b_2) \cap V(s_1.id) = \{$	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 }			
$V(b_2) \cap V(s_2.id) = \{$	社員番号,	氏名,	所属,	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 }			
$V(b_2) \cap V(s_3.id) = \{$	社員番号,	氏名,	所属 }	
$V(b_3) \cap V(s_1.id) = \{$	社員番号,	氏名,	所属,	人事考課 }
$V(b_3) \cap V(s_2.id) = \{$	社員番号,	氏名,	所属 }	
$V(b_3) \cap V(s_3.id) = \{$	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 }
$V(b_4) \cap V(s_1.id) = \{$	社員番号,	氏名,	所属,	資格 }
$V(b_4) \cap V(s_2.id) = \{$	社員番号,	氏名,	所属 }	
$V(b_4) \cap V(s_3.id) = \{$	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 }		
$V(b_5) \cap V(s_1.id) = \{$	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 }
$V(b_5) \cap V(s_2.id) = \{$	残業時間数,	欠勤日数,	遅刻日数,	早退日数 }
$V(b_5) \cap V(s_3.id) = \{$	役職,	人事考課,	資格 }	

図 1 業務記述とソフトウェア記述の間の距離を設定するための語の集合

Fig. 1 Word sets for calculating distances between business descriptions and software descriptions.

本記述では、「職務知識」、「関連知識」、「洞察力」、「折衝力」、「表現力」、「職務」、「職能」などの語が用いられている。語の集合に関するこれ以外の変更を示す。

- b_1 で用いる語として、「職能」を追加する。
- b_3 で用いる語として、「職能」、「滞留年数」を追加する。

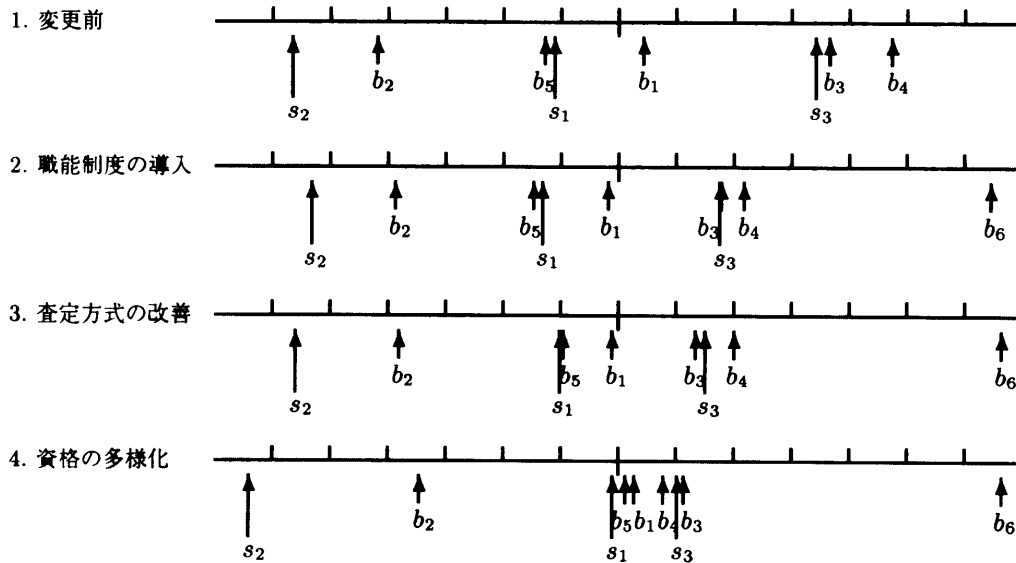


図2 業務記述とソフトウェア記述の布置図の変化

Fig. 2 Transitions of a scattered diagram of business descriptions and software descriptions.

- b_5 で用いる語として，“職能”を追加し，“基本給”を“年齢給”と“職能給”に分ける。
 - s_1 で用いる語として，“職能”を追加し，“基本給”を“年齢給”と“職能給”に分ける。
 - s_3 で用いる語として，“職能”，“滞留年数”を追加する。
- (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。この変更によって， b_5 で用いる語として，“人事考課”を削除し，“実績”，“態度”，“能力”を追加する。 s_1 で用いる語についても同様とする。
- (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。この変更によって， b_4 で用いる語として，“資格”を“資格-1”～“資格-8”に分ける。 b_5 ， s_1 ， s_3 で用いる語についても同様とする。

各変更に対する布置図を時系列に並べたものを図2に示す。布置図の変化から，次の解釈が導かれる。すなわち，変更前には散らばった位置にあった業務記述 b_1 ， b_3 ， b_4 ， b_5 ，および，ソフトウェア記述 s_1 ， s_3 が，次第に接近して，1つにまとまる。このことは，能力と給与の関連性が強くなっていくことを意味し，「能力主義の人事制度に移行する」という本来の意図に適合しているので，これらの変更は妥当であると判断される。このように，本技法によって，企業活動における改善点に対する改善案の妥当性を確認すること

ができる。

5. 関連研究

企業情報システムの特性をドメインモデルとして表現することによって，企業情報システムの特性と業務分析技法，および，開発方法論との関係が明確になる。本章では，このようなドメインモデルの役割と，他の研究におけるドメインモデルの役割を比較する。

ソフトウェア開発におけるドメインモデルの役割は，多岐にわたるが，その多様性を説明する枠組みとして，Triadic Domain Modelが提案されている⁸⁾。この枠組みでは，ソフトウェア開発の各フェーズごとに，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 という3種類の概念に基づいて，ドメインモデルを分類する。要求分析フェーズについて考えると，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形であり，Domain Process Modelは，実世界における業務のインフォーマルな表現からフォーマルに記述したソフトウェアの要求仕様を導く手順のひな形である。本研究におけるドメインモデルは，要求分析フェーズにおける Domain Problem Modelと Domain Product Modelを融合したものであるといえる。また，本ドメインモデルとDOAにおける分析の手順の関係を明らかにすることによって，DOAにお

る分析の手順を、企業情報システムのための Domain Process Model として位置付けることができると考えられる。

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⁴⁾。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¹³⁾。また、ドメインモデルは、要求者と分析者の間のコミュニケーションの基盤としても利用される¹⁶⁾。本研究における業務記述は、この意味でのドメインモデルに相当するといえる。ここで注意すべき点は、この意味でのドメインモデルは、再利用されることを前提として、標準化されているとは限らないことである。

一方、再利用されることを前提として、業務知識の標準を表現した業務記述は、要求分析フェーズにおける Domain Problem Model としてとらえることができる。同様に、機能要求の標準を表現したソフトウェア記述は、要求分析フェーズにおける Domain Product Model としてとらえることができる。しかし、本研究におけるドメインモデルは、業務知識や機能要求の標準を表現するものではなくて、企業情報システムにおける業務とソフトウェアの一般的な特性を表現するものである。

R. Prieto-Diaz と P. Freeman は、ドメインの特性を表現するドメインモデルの形式として、ファセット分類のスキーマを提案した¹⁴⁾。ファセットとは、対象領域に関連する用語の集合である。いくつかのファセットのそれぞれに属する用語を組み合わせ、当該ドメインの製品の特徴を記述する。このドメインモデルは、ドメインの特性に基づいて、製品を体系的に管理することを目的とする。

これに対して、A. Davis は、

- (1) 問題の難しさ
 - (2) データと処理の間の時間的な関係
 - (3) 同時に実行されるタスクの数
 - (4) 問題の難しさがデータとコントロールとアルゴリズムのいずれの側面にあるか
 - (5) システムの挙動が決定的か非決定的か
- というファセットでドメインの特性をとらえ、各種の要求分析法がどのような特性を有するドメインに適しているかを考察した²⁾。これは、本研究と同様に、ドメインの特性と要求分析法の関係に注目している。しかし、本研究におけるドメインモデルでは、ドメインモデル自体に基づいて、両者の関係を説明することができるが、ファセットによるドメインの特性の記述で

は、それができない。

特定のドメインにおける要求分析を支援するドメインモデルとしては、これまでに、リアクティブシステムのためのドメインモデルで、開発環境 Asdreas の基礎となる STD Triad¹⁰⁾、建築設計のためのドメインモデルで、建築部材記述言語 BDL の基礎となるもの⁶⁾、事務処理分野のソフトウェアにおける要求仕様の再利用のためのドメインモデル¹²⁾などが提案されている。

6. おわりに

本論文では、企業情報システムの特性を表現するドメインモデルを定義した。そして、企業情報システムの特性をふまえて、DOA を実際の開発プロジェクトに適用できるように、本ドメインモデルと DOA における分析の手順の関係を明らかにした。次に、本ドメインモデルに基づく業務分析技法を提示し、本技法と DOA が共通のドメインモデルに関係付けられることによって、DOA を実践する中で、本技法を適用できることを示した。また、企業活動における改善点に対する改善案の妥当性を確認するために、本技法が有効であることを示した。

今後の課題としては、まず、実際の開発プロジェクトにおいて、本ドメインモデルを積極的に利用していく必要があると考えている。そのために、業務記述とソフトウェア記述の作成を支援する CASE ツールと連携して、本論文で提示した業務分析技法の適用を支援する環境を提供することを検討している。既存の上流 CASE ツールが支援する特定の表記法にしたがって、業務記述とソフトウェア記述が作成されているならば、本技法を適用するために必要な情報を自動的に抽出できると考えられる。

本論文で提示したドメインモデルは、企業情報システムのすべての特性を表現しているわけではない。本ドメインモデルによって表現されない重要な特性としては、たとえば、情報システムに組み込まれている業務規則に関する特性、ユーザ・インタフェースに関する特性、ハードウェアを含むシステムの構成に関する特性などがある。また、業種に応じてドメインを細分化することによって、細分化されたドメインに対して、固有の特性を付加することができる。これらの特性を明らかにすることも、今後の課題である。

これまでの企業情報システムは、SA や DOA を適用して開発されたものが多かったため、これらのシステムを開発した経験に基づいて定義された本ドメインモデルは、これらの開発方法論に適合したものになっ

ている。しかし、今後は、OOAを適用して開発されるシステムも多くなると考えられるので、OOAの考え方に適合したドメインモデルについても検討していきたい。

謝辞 本研究は、著者が情報処理振興事業協会(IPA)に在職中に、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、IPAが新エネルギー・産業技術総合開発機構から委託を受けて実施したものである。また、有益な助言をいただいた査読者の方々に感謝する。

参考文献

- 1) Coad, P. and Yourdon, E.: *Object-Oriented Analysis*, Yourdon Press (1990).
- 2) Davis, A.M.: *Software Requirements: Objects, Functions and States*, Prentice-Hall (1993).
- 3) DeMarco, T.: *Structured Analysis and System Specification*, Yourdon Inc. (1978).
- 4) Finkelstein, A.: Requirements Engineering: A Review and Research Agenda, *Proc. First Asia-Pacific Software Engineering Conference*, pp. 10-19 (1994).
- 5) 林知己夫, 飽戸 弘(編): 多次元尺度解析法, サイエンス社 (1976).
- 6) 廣田豊彦, 橋本正明, 長澤 勲: 応用ドメインに特化した概念モデル記述言語に関する一考察, *情報処理学会論文誌*, Vol.36, No.5, pp.1151-1162 (1995).
- 7) 堀内 一: データ中心システム設計, オーム社 (1988).
- 8) 伊藤 潔, 田村恭久, 杵嶋修三: 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 論文集, pp.81-88 (1994).
- 9) Jackson, M.: *System Development*, Prentice-Hall (1983).
- 10) 杵嶋修三, 伊藤 潔: リアクティブシステムの分析・設計向きドメインモデル: Asdreas STD Triad, *情報処理学会論文誌*, Vol.34, No.9, pp.2025-2036 (1993).
- 11) Martin, J.: *Information Engineering*, Prentice-Hall (1989).
- 12) 名取万里, 加賀谷聡, 本位田真一: 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).
- 13) 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, *情報処理学会研究報告*, 94-SE-96-13 (1994).
- 14) Prieto-Diaz, R. and Freeman, P.: *Classifying Software for Reusability*, *IEEE Software*, Vol.4, No.1, pp.6-16 (1987).
- 15) Rumbaugh, J., et al.: *Object-Oriented Modeling and Design*, Prentice-Hall (1991).
- 16) 齊藤康彦, 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, pp.2469-2479 (1994).
- 17) 下崎千代子: 人事情報システム, 日科技連出版社 (1993).
- 18) 田村恭久, 伊藤 潔, 杵嶋修三: ドメイン分析・モデリング技術の現状と課題, *情報処理*, Vol.35, No.10, pp.952-961 (1994).

(平成8年1月10日受付)

(平成8年7月4日採録)

齊藤 康彦 (正会員)



1961年生。1984年早稲田大学教育学部卒業。同年,(株)協栄計算センター(現,(株)アイネス)入社。1991年より情報処理振興事業協会(IPA)に出向。新ソフトウェア構造化モデル研究本部にて、要求分析法の研究に従事。現在は,(株)アイネスにて、ソフトウェア開発環境の研究に従事。人工知能学会会員。

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年度大阪大学大学院非常勤講師兼任。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。著訳書10点。