ドメイン知識表現が要求獲得に与える影響の分析

長田　晃[†]　小澤　大伍[†]　海谷　治彦[†]　海尻　賢二[†]

† 信州大学 大学院　〒 380-8553 長野市 若里 4-17-1

あらまし　ドメイン知識は要求獲得作業に必須である。要求獲得において効率的にドメイン知識を使用するために、幾つか考慮すべき点が挙げられるが、我々は適切な表現形式を選択することに注目する。なぜなら、ドメイン知識自体は非明示的なもので、実際にはある表現形式によって明示化されて我々に示されると考えられるからである。よって、ドメインに合った、適切な表現形式を選択することが、有効なドメイン知識の利用につながると考えられる。本研究は、その第一段階として、表現形式が分析者に与える効果を実験を通して調査した結果を報告する。

The Role of Domain Knowledge Representation in Requirements Elicitation

Akira OSADA[†], Daigo OZAWA[†], Haruhiko KAIYA[†], and Kenji KAIJIRI[†]

† Graduate School of Science and Technology, Shinshu University
4-17-1, Wakasato, Nagano city, 380-8553, Japan

Abstract　We aim to investigate the effect of representation style on the process of requirements analysis. We performed a comparative experiment using two representations style: UML based style and dictionary style. We analyzed requirements documents using defined metrics. As a result, dictionary style seemed to be adequate for the comprehensive description and UML style seemed to be adequate for the as-is reuse of the domain knowledge. The remaining problem is to show the dependency of the effectiveness on the target domain.

1. はじめに

情報システム開発は、その機能や振る舞い、構造といった側面はもちろん、規模や開発期間、開発コストにいたるまで、非常の多くのバリエーションを持ち、千差万別である。すなわち、システムがどの種類のどの業務のどの役割で、どの程度の仕事を行うかに関する決定に、多くのバリエーションがあるということである。よって、要求獲得を成功させるためには、決定すべき事柄について多くの知識を持つ、熟練したアナリストが必要とされている。しかし、ある分野において熟練したアナリストが、別の分野においても熟練したアナリストであるとはいえない。それは、餅は餅屋という格言と同様の現象である。しかしながら、彼らはアナリストとしての高いスキルを持っており、共通知識 (一般常識や技術情報) 以外の、その分野固有の知識さえ補完できれば、十分に分析作業を行えると考えられる。それは、熟練したプログラマーにとって、新しい開発言語を使用する時に、その言語のリファレンスマニュアルが大きな助けとなるのと同様である。我々はそのような固有の特徴を持つ分野を、ドメインまたは問題ドメインと呼び、分野固有の知識をドメイン知識と呼ぶ。基本的にドメイン知識はドメインの全ての情報 (対象ビジネスの文書、豆知識、ビジネスルールなど) を含むものとする。

ドメイン知識を得るためには、ドメインの熟練者 (実務者やアナリスト) に尋ねること [13] や、実際に対象のビジネスに参加する [9] など、幾つかの方法がある。しかし、それらは時間と労力が掛かり現実的ではない。そこで我々の研究の最終目標を、ドメイン知識を体系的にデータ化する手法の確立とした。本論文は、その始めとして、ドメイン知識のデータ化に際してどのような表現にどのような効果があるのかについて研究している。なぜならば、情報のデータ化にはそのメタモデルが必要であるが、メタモデル作成のためには、出力としての表現形式がアナリストに与える影響について考慮し、ふさわしい表現形式を選択しなければならないからである。例えば、図は文章よりも直感的で理解しやすいが、逆に詳細な情報を表現するのに不向きであるといわれている。すると、図を基本とするデータ化と文章を基本とするデータ化では、その構造が大きく違うものとなる。また、我々は以前の研究 [10] において、ドメイン知識を要求獲得に使用した実験を行っているが、その際、被験者から「このドメイン知識は、必要な情報を見つけにくく、また、理解しづらい。さらには、思考を混乱させる」との指摘を受けた。よって、我々はユーザーとドメイン知識とのインターフェースとして、表現形式が重要であることを認識した。

本論文では、要求分析プロセスにおける表現形式の効果性を調査した。1 つのドメイン知識を 2 種類の表現形式で被験者に

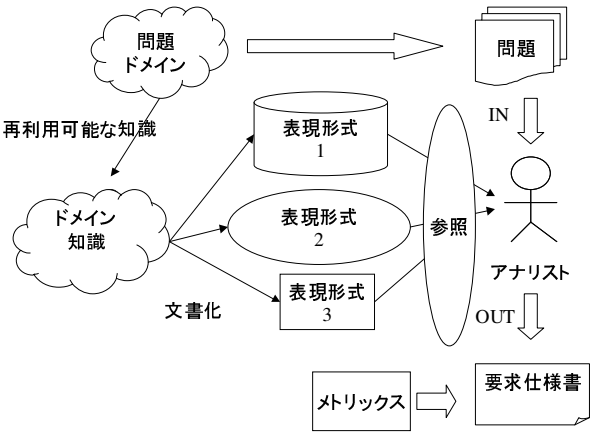

図 1　問題ドメインと表現形式との関係

提供して要求分析を行わせ、その分析結果を評価することでドメイン知識表現の効果を見る。その評価のために 4 つのメトリックスを定義している。

本論文の構成は以下のとおりである。2 章では、ドメインとドメイン知識表現との関連について説明し、3 章では、問題ドメインのカテゴリについて言及する。4 章ではドメイン知識を整理するために、ドメイン知識の特徴を定義し、5 章では、要求獲得プロセスにおけるドメイン知識の効果を測るためのメトリックスを定義する。6 章で、効果性の確認実験の詳細とメトリックスによる測定結果を示す。最後に、本実験の結果と今後の予定について述べる。

2. 問題ドメインと表現形式の関係

ドメイン知識のデータ化をするためには、表現形式について考慮しなければならないが、本章で、ドメイン知識と表現形式との関係について明らかにする。図 1 は要求分析における問題ドメインと表現形式の関係およびメトリックスについての説明である。熟練したアナリストであれば非形式的な形で持っているような、現実の問題領域に対して再利用可能な知識が、「ドメイン知識」である。ドメインに詳しくないアナリストはそれらを何らかの「表現形式」によって具体的な形とすることで、「参照」することができる。対して、要求分析プロセスとは、「問題ドメイン」から発生した「問題」を分析し、「要求仕様書」を記述する作業である。さらには、要求仕様書の出来を確認する作業も含むこともある。それゆえに、問題ドメインを認識する助けとなる、ドメイン知識の表現形式の選択には、注意を払うべきであると考えられる。

ドメイン知識の明示化が容易であったり、表現形式の間で変換が容易であれば、表現形式に注意を払う必要はない。しかし実際には、ドメイン知識の明示化は手間がかかり、一旦明示化するとその表現形式に依存した情報となってしまうため、他の表現形式に変換することも難しい。全く同じ情報を別の表現形式で表すことはできるが、逆に表現形式のもつ特色を弱めることにもなる。そこで我々は、始めから表現形式に依存した形でデータ化する手法を想定し、その準備として表現形式の効果を測ることとした。ある表現形式の効果を評価するためには、問題ドメインに対して実際に要求分析の実験を行い、その結果を測定することが必要である。この評価は、それぞれの問題ドメインごとに実験が必要となり、コストが掛かる。しかし、問題ドメインをカテゴライズする手法はいくつかあるため [8] [3]、同じカテゴリに属する問題ドメインには、同様の分析結果が適用できると期待する。

3. 問題ドメイン

あるドメインについてのある表現形式の効果を測ることは可能であるが、ドメインは千差万別であるため、ある特殊なケースのみの結果となってしまう。しかしながら、問題ドメインを分類する研究は既にあり、類似した問題ドメインにおいては、ある表現形式の効果に類似した結果が期待できるのではないかと考えた。研究の例を挙げれば、[3] は 3 つの大きなドメインに分けており、[8] はオブジェクトシステムモデルを使用して問題ドメインを分類している。それは例えば、資源の返還 (車のレンタルなど)、資源の供給 (注文購入など)、資源の使用 (注文販売など)、などである。我々はごく自然に、車のレンタル業務は図書館業務と類似していることを理解できる。そして、2 つの業務は、2 人のアクターがオブジェクトの所有権を譲渡しあうというほとんど同じものになるため、それを表現する形式として同じ形式 (例えばアクティビティ図) を選択した場合に、同じ効果が現れるのは、自然なことであると思われる。

本論文では、問題ドメインの分類については扱わない。既存の研究において、分類は出来ているものとする。

4. ドメイン知識

ドメイン知識を表現するモデルは既に多くある。例えば、PAORE [6] は機能を階層構造で示すことによりドメインを表現しており、[10] では、単語間を共起関係で結んだマトリックスで表現している。また、それぞれにおいて、ドメイン知識とは質や量も違うものとなる。本章ではドメイン知識をデータ化するために、ドメイン知識のパラメータとなる各々の特徴について改めて考察する。なお、要求獲得作業が必要とするドメイン知識のなかでも、一般常識と技術知識以外の、顧客のビジネスに関する固有の知識について注目する。

4.1 ドメイン知識の特徴

ドメイン知識は以下によって、特徴づけることができる。

- 量

ドメイン知識として扱うデータの量について考えるのは重要である。獲得すべき要求について十分な情報があることが必須条件であるが、どの程度が十分であるかについては、アナリストの事前に持っている知識などによって千差万別である。また、多すぎる情報は、アナリストを混乱に落としいれる。

- 質

ここでいう質とは、情報が適切であるかどうかである。逆にいえば、ドメイン知識として不適切な情報がないこと、と言い換えることもできる。一般に、正当性や、正確性などといわれるものである。

- 視点

ステークホルダーが違えば、必要とする情報も違う。例えば、ユーザーマニュアルはユーザー視点で使い方を説明しているが、

仕様書は SE の視点で書かれている。アナリストは場合に応じて視点を切り替えなければならないため、ドメイン知識の特徴として、視点は重要な特徴である。

- 表現形式

我々は、ある情報を表すときに、文章で自在に表現したり、フォーマルな図式言語を用いたりと、様々な表現形式を用いることが出来る。同じドメイン知識でさえ、別の表現形式で表すと全く別の印象を受けるため、アナリストはしばしば違う要求を獲得することになる。よって、ドメイン知識を文書化する際には、表現形式を考慮することは重要である。

- 抽象化

問題点を効果的に発見するためには、適切な抽象化が重要である。例えば、具体的に過ぎると、例外フローや詳細なプロパティなどに気を取られ、全体像を把握するのに時間がかかる。しかし、詳細な情報がなければ、システムは多くの欠陥を抱えることになる。要求獲得を補うためのドメイン知識は、適切な抽象度を持つべきである。

本研究では、上記のドメイン知識の特徴のうちの 1 つである、表現形式に注目する。以前の研究 [10] において、ドメイン知識を要求獲得に用いるときに、まず表現形式が重要であることを認識したからである。

5. メトリックス

図 1 に見られるように、ドメイン知識自体は、基本的に直接計測できるものではなく、ある具体的な形で表現されることで明示化されることとなる。以降、ドメイン知識をある表現形式で明示化したものをドメイン資料と呼ぶ。我々は、そのような表現形式の効果性をどのように測るかという指針がないため、実際に要求獲得作業を行い、その作業結果を調べることにした。作業結果を測るために、4 つのメトリックス (再現率、精度、依存度、汎用率) を定義した。なお、ドメイン資料の情報を表す単位として、“概念” という単語を使用している。一般的な情報の単位として用いられる “単語” を、基準としても良いが、同じ単語が抽象化レベルの違う内容を表していることが多々あるため、あえて “概念” という言葉を用いている。基本的には、単語の集合のことである。

表現形式を測るためには、直接的に表現形式を測る測度を用意するべきだが、それは非常に難しく手間が掛かり、さらにはあまり有効ではなさそうである。それは図の線の数と、文章の単語の数を数えて比較しても、あまり意味がないと思われるからである。よって、表現形式を含めてすべての特徴を測る測度を用意した。そして実験の際には、表現形式以外の特徴を出来る限り同じにすることにより、得られた結果に表現形式の効果が現れるようにする手法をとった。

5.1 再 現 率

再現率はドメイン資料が要求獲得にとって、十分なものかどうかを測る。この再現率が低いと、ドメイン知識が不十分なものであったといえる。再現率は以下の式で測る。

$$\text{再現率} = \frac{\text{獲得された概念}}{\text{ドメイン資料内の全概念}}$$

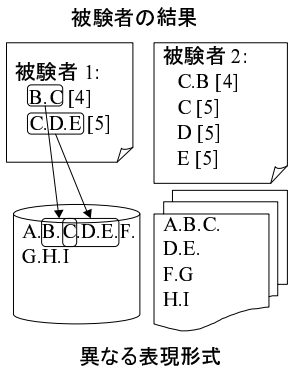

図 2 依存度の一例

5.2 精 度

精度はドメイン資料に正しい情報が含まれているかを測る。この精度が低いとドメイン資料としてふさわしくない情報が多く含まれていることになる。精度は以下の式で測る。

$$\text{精度} = \frac{\text{獲得された中で正しい概念}}{\text{獲得された全ての概念}}$$

5.3 依 存 度

依存度は獲得結果がドメイン資料に依存している割合を測る。依存度が高いと、ドメイン資料をそのまま使用していることになる。逆に依存度が低いと、ドメイン資料をアナリストが改変して使用していることになる。これは、結果のフォーマットとドメイン資料との相関として捕らえることもできる。依存度は以下の式で測る。

$$\text{依存度} = \frac{\text{ドメイン資料をそのまま使用}}{\text{獲得された全ての概念}}$$

図 2 は典型的な依存度の例である。被験者 1 はドメイン資料の 2 つの部分集合を結果に含んでいるが、被験者 2 は部分集合を持っていない。

5.4 汎 用 率

他の測度は被験者個々人について測定するが、汎用率は同一のドメイン資料を使用したグループごとに測定する。この測度は、ドメイン資料と被験者の集合との親和性を測る。つまり、異なるアナリストが同じドメイン資料を使用して獲得した結果の共通性と相違性を測るものである。汎用率が高いと、誰が獲得作業を行ったかに関わらず、類似した結果が出ることになる。汎用率は以下の式で測る。

$$\text{汎用率} = \frac{\text{グループの概念の積集合}}{\text{グループの概念の和集合}}$$

6. ケーススタディ

このケーススタディの目的は要求分析プロセスにおけるドメイン知識表現の効果性を確認することである。ドメイン知識とドメイン知識表現の 2 つの関係性を考慮するためには、1 つのドメイン知識を 2 つの表現形式で表すことと、2 つのドメイン知識を 1 つの表現形式で表す 2 種類のパターンが考えられるが、本実験では、1 つのドメイン知識を典型的な 2 つの表現形式で表し要求分析実験を行った。その結果を定義した測度で測ることとする。確認のために、以下の仮説を置く。

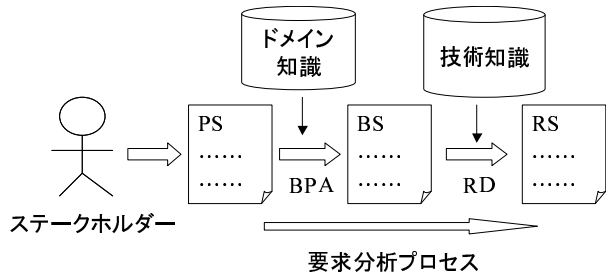

図 3　要求分析プロセス

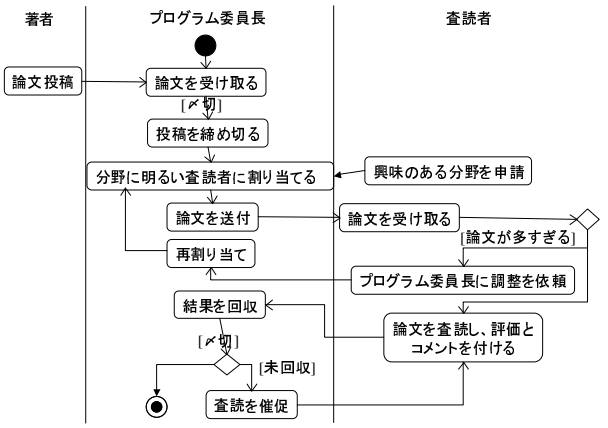

図 4　UML による表現形式の一例

- 測定結果は、異なる表現形式の元では、違う結果となる。
- 測定結果は、分析者の直感に沿うような、合理性がある。

この仮説が正しければ、少なくともこの実験においては、メトリックスを用いて表現形式の効果性を確認したと言える。さらに、この結果は類似した問題に対して、同様の効果性を見込む助けになると思われる。

6.1　ケーススタディ デザイン

ケーススタディのデザインについて説明する。

6.1.1　アウトライン

一般に、要求獲得プロセスの出力は要求仕様書であるが、仕様書には技術的知識 (自動返信メールやセッション管理など) が含まれる。そこで、ドメイン知識のみに注目するため、要求獲得プロセスを、要求分析プロセス (“BPA”) と仕様定義プロセス (“RD”) の 2 つに分けた。要求分析プロセスはドメイン知識を用いて、現行の業務を分析するプロセスである。仕様定義プロセスは技術情報も含めて、仕様を定義していくプロセスである。図 3 はケーススタディのアウトラインである。PS、BS、RS はそれぞれ、問題陳述書、業務分析書、要求仕様書の略称であり、6.1.2 節にて詳細を説明する。本実験では、“注文購入形式” ドメイン [8] の一部として “国際会議マネージメント” をドメインとして使用した。

そのドメインを 2 つの典型的な表現形式で表し、被験者に要求分析作業を行わせ、定義したメトリックスで測定した結果を比較する。表現形式の 1 つは UML 図であり、もう 1 つは文章で書かれた辞書形式である。現行の業務に含まれる様々な情報を、それぞれの表現形式で表した。図 4 は UML 図の一例で、“査読業務” をアクティビティ図で表したものである。図 5 はテキストの辞書表現の一例である。“プログラム委員長” と “査読

プログラム委員長:

プログラム委員会の長であり、会議のプログラムの決定に関して責任をもつ。主な仕事は、投稿された論文の管理とプログラム委員の選定と依頼、査読の依頼と査読結果の回収、プログラムの採択、不採択に関する最終的な決定である。

査読:

投稿された論文を査読者が評価し、プログラムに掲載するか否かの採否判定資料を作成する作業.

投稿された論文は、(ほとんどが複数の) 査読者よって、採点がつけられる。採点は, 合格か不合格かの結果と, その理由も査読者によって付記される.

公正かつ適切な査読を行うため, 当該論文の論じている分野に明るい研究者に 査読者になってもらい, 十分な時間を費やして査読することが望ましい. 査読者は投稿〆切後にすみやかに決定し, 査読はプログラム委員会の開催日前には終了し, その結果をプログラム委員長に報告しなければならない.

図 5　テキストの辞書表現の一例

表 1　問題陳述書

PS1	他の類似分野の会議との日程の衝突を避けるための情報が不足している.
PS2	プログラム委員に担当論文を割り振る作業が煩雑である.
PS3	既に投稿している論文の更新作業に手間がかかる .
PS4	投稿された論文を受け取り管理するのが大変 .
PS5	論文の投稿者に受け取った旨を連絡するのを忘れてしまうことがある .
PS6	論文〆切延長のプレッシャーが強くて困る.
PS7	査読結果を返さない査読者がいるためプログラムの決定が延びてしまう.
PS8	論文の採否を決定する際 , 全部の論文を議論する時間的余裕がない.

業務” の内容である。下線は他のタイトルへのリンクである。

6.1.2　使用する文書

本実験において使用する文書について説明する。

- 問題陳述書 (“PS”)

PS には顧客のビジネスに関する問題点と、システムによって解決したい要望が書かれている。表 1 は “ある国際会議のプログラム委員長” によって書かれた PS の例である。

- 業務分析書 (“BS”)

BS は PS に関連した現行の業務を明示したものである。当然ながら、全ての業務を明示できるわけではないので、PS に関連した部分のみに限定している。図 6 は BS の一部である。BS はアナリストが PS に書かれた問題点を明らかにする作業、つまり “業務分析プロセス (BPA)” の出力であり、ドメイン資料はその際に役立つ。

なお、BS 内には、PS との同期を取るために PS の番号をタグとして付けることとする。例えば、BS 内に “[3]” というタグが付いていれば、PS の 3 番目の問題についての分析であることを意味する。

- 要求仕様書 (“RS”)

本実験において、RS は BS を改訂して作られるものであり、機能、性能、外部インターフェースと制約について書かれている。図 7 は RS の例である。

本実験は図 3 に見られるように、BS から RS へ発展していく

BS C: (論文受け取り)
1.プログラム委員長は論文を受け取る.[4]
<>幾つかの形式で受け取り、保管する.
|> 郵便で.
|> PDFやMS-Wordなどの電子メールで.
2.新規投稿か論文の更新かをチェックする.[3]
|> 投稿者の申請.
|> タイトルと著者から既投稿論文を照会する.
3. 受け取った論文を整理する.
|> もし新規投稿であれば、保管する.
<> タイトルと著者を記録.
<> 連絡先を記録. [5]
<> 投稿日時を記録.
|> もし更新であれば、既投稿論文を更新する.
<> 更新日時は必ず記録する.

図 6　業務分析書の一部

機能 1: 投稿された論文にIDを振る.
機能 2: IDにより論文を検索する.
インターフェース 1: 論文を管理するデータベース.
品質 1: IDは一意であること.

図 7　要求仕様書: 業務分析書に追加

ものである。

6.1.3　設　　定

幾つかの設定について説明する。

- 問題ドメイン

ドメインは被験者にとっては未知の分野であるべきである。また、規模が小さすぎるドメインはふわさしくない。

- ドメイン資料

ドメイン資料は、該当ドメインのドメイン知識を文書化したものである。実験に使用する場合には、ドメイン知識の特徴の内で、質と量及び視点については出来る限り同じにするべきであり、表現形式のみ異なるべきである。

- 被験者

被験者は C 言語のような手続き型言語と JAVA などのオブジェクト指向言語、および、UML 図についての知識を持っているべきである。また、問題ドメインについて熟知していてはならない。

- 手順

– 事前学習と演習

要求獲得に関する全般的な知識について学習させる。さらに、被験者に良く知られているドメインを用いて、本実験の業務分析の手順と、分析書の書き方を学習させるために、演習を行う。

– BPA(業務分析プロセス)

用意した 2 つのドメイン資料に対して、被験者を 2 グループに分ける。それぞれの被験者は用意された PS を用いて、業務分析を行う。本実験は異なる表現で表されたドメイン資料の効果を確認するものであるため、問題ドメインは被験者にとって未知のものであるべきである。BPA の結果として得られる BS は、本実験の主要な結果として扱う。

– RD(要求定義プロセス)

被験者は要求仕様の定義を行う。このプロセスの主たる意味は、被験者のモチベーションを保つものである。BPA はしばしば被験者にとって面白いものではないため、BS を書くことを諦めてしまう可能性がある。RD の入力として BS があることを明示して、より詳細な BS を書かせることができる。

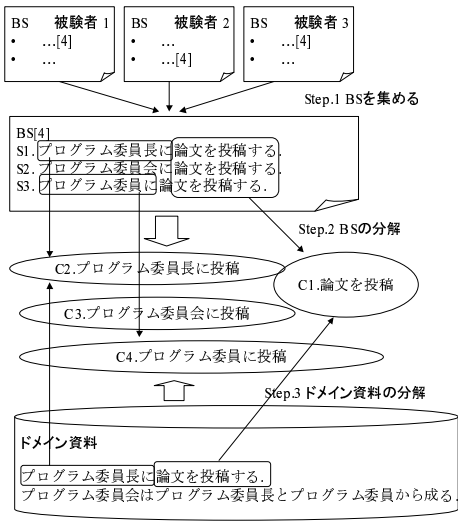

図 8　BS を概念に分解

6.1.4　分析の仕方

ドメイン知識とその表現について注目するため、BPA の出力である “業務分析書” のみを分析対象とする。BPA において、被験者は PS を理解するためにドメイン資料を必要とするため、BS は分析対象としてふさわしいと考えられる。

ドメイン資料を測るために、メトリックスを用いるが、その単位として “概念” を導入する。より一般的には、“単語” を用いることが出来るが、同じ単語が抽象化のレベルなどによって違う意味を持つことが多々あるため、動詞と名詞を組み合わせることで、意味を固定化することとする。動詞と名詞の対のことを本実験では “概念” と呼ぶこととする。

BS はこの概念という単位で分解し、測定される。図 8 を用いて、BS を概念に分解する手順について説明する。

（1） 各被験者から BS を収集し、タグの付いた文のみを抽出する。このタグは、被験者が問題を分析した箇所を意図的に指し示していることを意味しているからである。これはつまり、ドメイン資料を使用したとみなすことができる。図 8 で言えば、3 人の被験者から、3 つの文章 (S1, S2, S3) を抽出している。

S1　プログラム委員長に論文を投稿する。
S2　プログラム委員会に論文を投稿する。
S3　プログラム委員に論文を投稿する。

（2） 文章を概念に分解する。S1-S3 を C1-C4 の概念に分解している。

C1　論文を投稿
C2　プログラム委員長に投稿
C3　プログラム委員会に投稿
C4　プログラム委員に投稿

（3） ドメイン資料を同様に概念に分解する。基本的に文章を概念に分解するため、図式表現から概念を抽出することが多少の困難を伴うが、表現形式以外を同じにするという制約から、文章形式のドメイン資料の概念が図式表現の概念と同じとみなすことができると考えられる。そのため、本実験では、文章形式から概念を抽出したものを全概念として設定する。

（4） 分析結果とドメイン資料から抽出した概念をメトリックスを用いて測定する。例においては、C1-C4 を測定する。

- 再現率

ドメイン資料は C1 と C2 を持っている。

$$\text{再現率}: \frac{2(\text{ドメイン資料})}{4(BS \text{ の全概念})} * 100 = 50\%$$

- 精度

C1-C3 が正しく、C4 は間違いである。

$$\text{精度}: \frac{3(\text{正しい概念})}{4(BS \text{ の全概念})} * 100 = 75\%$$

- 依存度

依存度はドメイン資料がそのままに使用された割合を測るため、元のままの情報として、概念ではなく、文章を使用する。つまり、文章とドメイン資料の部分集合との適合を測る。S1 はドメイン資料そのままに使用されている。

$$\text{依存度}: \frac{1(\text{資料と一致})}{3(\text{使用された概念})} * 100 = 33\%$$

- 汎用率

C1 は 3 人の被験者に使用され、C2-C4 はそれぞれに使用されている。

汎用率 $(C1)$: $\frac{3(3 \text{ 人が使用})}{3(\text{全被験者数})} * 100 = 100\%$

汎用率 $(C2 - C4)$: $\frac{1(1 \text{ 人が使用})}{3(\text{全被験者数})} * 100 = 33\%$

$$\text{汎用率 (平均)}: \frac{(100 + 33 + 33 + 33)}{4} = 49.8\%$$

6.2 結　　果

本学において、4 回の講義を使って、実験を行った。2 つのドメイン資料 (A,B) を用意し、応じて被験者を 2 つに分けた。

- 問題ドメイン

“国際会議マネージメント” をドメインとして選択した。被験者の学生には知名度が低く、多くのワークフローを持つ。よって、このドメインは適切である。

- 用意したドメイン資料
  - 資料 A: UML 図

クラス図とアクティビティ図、ユースケース図をドメイン知識表現として使用した。クラス図は、国際会議のオブジェクト (論文、差読者、プログラム委員会など) の関連を表現している。アクティビティ図は国際会議の管理業務について表現している。ユースケース図は、国際会議と各アクターとの関連を表現している。図 4 はアクティビティ図の一例である。それぞれの図に関する数は以下である。

クラス図:19. アクティビティ図:70. ユースケース図:15.

  - 資料 B: テキストベース

これは、単語辞書であり、タイトルの語とその説明で構成されている。さらに、説明文中に他のタイトル語へのリンクをつけた。図 5 はテキストベース資料の例である。テキストベース資料の規模は以下である。

タイトル語の数:32. 1 タイトル当りの説明文の単語数:73.

- 被験者

本学の学部 3 年生 32 人を対象とした。C 言語と JAVA、および UML 図については学習済みである。“国際会議運営” については、不慣れである。25 人がデータの使用に賛同したため、25 人分のデータを使用している。UML 図を使用したグループは 12 人であり、テキストの辞書を使用したグループは 13 人である。

- 手順

図 3 に沿って、以下の 4 回の手順で実験した。

（1） 事前学習と演習

本実験における分析作業について解説した。被験者は、図書館業務を例として、実際にビジネスプロセス分析を行い、BS の書き方を練習した。図書館業務は被験者になじみ深いため、ドメイン資料は用意しなかった。1.5 時間ほどの時間を費やした。

（2） BPA (ビジネスプロセス分析)

被験者は、国際会議運営を問題ドメインに使用し、BPA を行った。2 つのドメイン資料 (UML 図と用語辞書) を用意し、それぞれを使う 2 つのグループに分けた。問題陳述書は表 1 の 8 つの記述から構成されていた。この作業には 2 時間を掛けた。この作業結果 (BS) が本実験の主たる分析対象である。

（3） RD (要求定義プロセス)

被験者は引き続いて要求定義を行った。BS に技術情報を付加し、RS にする作業であり、2 時間を掛けた。

（4） 要求分析 (BPA + RD)

被験者は、ドメイン資料を入れ替えて新しい 5 つの問題に対して要求分析を行った。2 時間を掛けた。

BS をメトリックスを用いて測定した。表 2 はグループごとに測定した結果である。表 3 は PS の各々の問題ごとに測定した結果である。

6.3 考　　察

仮定を確認し、それぞれの結果について考察する。

- 再現率

テキスト資料の再現率が高い。この結果はテキスト資料が理解しやすく、使用が容易であったことを意味していると考えられる。これは、テキスト資料は自然言語で書かれており、被験者はテキスト形式の資料になれていることを考えると、合理的な結果であるといえる。

- 精度

2 つの測定値はほとんど同じである。しかし、標準偏差について鑑みると、UML 図の方が高いことがわかった。この結果は、UML 図のグループにおいて、かなり高い精度の集団とかなり低い精度の集団に分かれたこと、しかしながら、テキストのグループは、一様に類似した精度を示したことを意味していると考えられる。これは、テキスト形式が知識を逐一追加していくこと、UML 形式は理解に時間が掛かるが一旦理解すると全体像をつかみ易いことを考えると、合理的な結果であるといえる。

- 依存度

UML 資料の依存度が非常に高い。この結果は、テキスト資料が改変されて使用されるのに対し、UML 資料はそのままに使用されていることを意味していると考えられる。これは、テキスト資料が 1 箇所に様々な情報を含み、使用するためには情報のピックアップが必要なのに対し、UML 形式はそのまま再利用が可能であることを考えると、合理的な結果であるといえる。

個人ごと	再現率		精度		依存度		汎用率	
	TEXT	UML	TEXT	UML	TEXT	UML	TEXT	UML
平均	23.5	18.7	90.6	93.3	62.1	85.3	14.5	20.3
標準偏差	7.4	3.8	5.9	10.3	11.5	16.9	12.7	16.2

表 2　測定結果の被験者ごとの平均

問題ごと	再現率		精度		依存度		汎用率	
	text	UML	text	UML	text	UML	text	UML
PS1. 他の会議との日程の衝突を避ける	84.6	60.0	91.0	88.8	44.0	37.5	17.5	24.5
PS2. 論文を割り振る作業が煩雑	76.9	64.3	93.8	86.1	66.7	81.8	16.7	18.6
PS3. 論文の更新作業に手間がかかる	72.7	69.2	98.3	100.0	57.1	81.8	17.6	31.1
PS4. 論文を受け取り管理するのが大変	80.0	64.0	74.2	99.4	40.0	88.9	12.1	27.8
PS5. 受け取った旨を連絡するのを忘れてしまう	100.0	75.0	91.2	100.0	66.7	66.7	19.5	19.4
PS6. 文〆切延長のプレッシャーが強く	70.0	55.6	71.5	79.9	26.9	58.8	9.2	11.3
PS7. 査読結果を返さない査読者がいる	61.5	75.0	82.7	97.6	47.1	64.3	10.9	13.7
PS8. 全部の論文を議論する時間的余裕がない	58.3	73.9	93.8	92.1	33.3	89.5	16.1	18.9
平均	75.5	67.1	87.1	93.0	47.7	71.2	14.9	20.6
標準偏差	13.3	7.3	9.8	7.5	14.7	17.8	3.7	6.8

表 3　問題ごとの測定結果

無論、結果としての BS のフォーマットが業務のフローをメインに書く形式になっていることが大きな要因である。

- 汎用率

UML 資料の汎用率が高い。この結果は、UML グループの多くの被験者が類似した分析を行ったことを意味していると考えられる。これは UML 図が具体的で、誰に対しても限定した情報として伝達されることを考えると、合理的な結果であるといえる。

全体の傾向は上記のものであるが、PS 内のそれぞれの問題記述の中には、逆の傾向が見られるものがある。続いて特徴的な結果について説明する。

PS1 は UML の中で依存度が最も低い。UML は本実験における出力形式と非常に親和性が高いため、驚く結果である。この結果は、多くの改変を受けたか、資料をあまり参照されなかったことを意味していると考えられる。これは、PS1 に必要な情報が多くの項目の情報であり、UML 図表現では詳細な情報が埋没することが多いことを鑑みれば、おそらく納得できるものであると思われる。

PS7 と PS8 の再現率は、全体とは逆にテキスト資料の方が低くなっていた。この結果は、テキスト資料から使用された概念が少なかったことを意味していると考えられる。これは、PS7 と PS8 が後段の情報を必要としており、UML 図は業務をシーケンスとして提供しているが、テキスト資料は後段の作業を直接的に明示してはいないことを考えると、合理的な結果であるといえる。

PS4 の精度には、ほぼ同値の全体とは逆に大きな隔たりが存在する。PS4 に関して詳細に分析した結果、テキストグループは “プログラム委員会” を “運営委員会” と誤使用するケースが見られた。当然ながら、テキスト資料には “論文はプログラム委員長に送付される” と “査読業務” の説明文に書かれており、かつ “プログラム委員長” の説明文に、“論文を受け取る” ことが仕事の一部である記載がある。しかし、慣れないドメインに対する分析作業中であれば、語感を優先して勘違いを犯すことは十分考えられるため、おそらく納得できるものであると思われる。

PS6 は両グループ共に精度が最も低く、他の測度に関しても、一様に低い結果が見られた。この結果は、PS6 が NFR に関わる問題であることを鑑みれば、合理的な結果であるといえる。特に UML などの形式では、NFR の問題を表現することは難しいものである。

全体的に、汎用率は依存度と相関が高いが、PS8 に関しては、依存度に隔たりがあるにも関わらず、汎用率にはそれほどの差がない。これは、UML 形式が多く再利用されたにもかかわらず、同じ結果を出した被験者が少なかったことを意味する。つまり、PS8 に関して再利用できる部分が多くあり、かつ、被験者は別々の部分を再利用していたことになる。残念ながら、この点に関して、合理的な説明ができていない。

結果のまとめとして、テキスト資料は多くのアナリストに馴染み易いが、時々誤解を誘発し、ほとんどの場合に改変されて使用され、全体に渡る情報を得ることには不向きであるという傾向にあることがいえる。また、UML 資料は、多くが再利用可能であるが、アナリストに理解力を要求し、詳細な情報は埋没しやすく、誰が使用してもある程度同じ内容として受け取られる傾向にあるといえる。

7. 関 連 研 究

ドメイン知識は要求獲得において明確に重要であるが、ドメイン知識の表現の仕方については、あまり研究がなされていない。代わりに、要求仕様書自身の表現の仕方についての研究は多い [11]。[5] に、5 つの分類 (“ユーザーマニュアル”、“視覚化”、“フォーマルメソッド”、“プロトタイプ”、“テストケース”) についての言及がある。4.1 節で使用したドメイン知識の特徴は、この分類に影響を受けている。例えば、特徴の “視点” はこの分類の “ユーザーマニュアル” をベースにしているしかしながら、幾つかの分類については考慮していない。本実験においてドメイン知識は単体で存在するため、“プロトタイプ” な

どは不要である。また、推論を行わないため “フォーマルメソッド” も不要である。情報の集合を扱うものとして、将来的には推論機構は必要であるが、今回は扱っていない。

UML はドメイン知識を表現する典型的な表現形式の 1 つである [2]。UML は組織やビジネスなどの、構造、機能、振る舞いのプロパティを表現することができる。[12] において、それらの類似の分類についても言及されていた。以上のことから、ケーススタディで扱う表現形式として UML を用いた。

よく知られている表現形式としてクラス図に似た、LEL [1] がある。PAORE [6] と呼ばれる手法では、ドメイン知識は機能分解され、シンプルな木構造として表現されている。

現実世界とと要求との関係はドメイン知識を考慮する際に重要である。Problem Frams [3] はソフトウェア開発における問題の構造を表している。

最近、オントロジーを、知識として使用するものが増えてきた。オントロジーには多くの種類があるが、[4] は単純に、“シソーラスと推論機構の組み合わせ” でオントロジーを作っている。この定義をベースに、類語辞典の形式での表現の一部として、今回の実験において、表現形式に辞書形式を採用した。インターネット技術、特に WIKI がこの用語辞書を作成するのに役立った。

我々は、ドメイン知識の表現についての研究を見つけることが出来なかったが、[5] が要求獲得に際して複数の表現形式を用いて書くことで、様々な角度から検討を重ねる手法について言及していた。

[7] はどのような表現形式が学生に好まれるかどうかについて調査していた。表現形式自体は直接要求工学に関わらないが、好悪を探索する手法として、利用可能であるとおもわれた。

8. 結　　論

我々は要求獲得におけるドメイン知識表現の役割を調査している。ドメイン知識の効率的な理解に、表現形式が重要であると思われるため、表現形式とドメインの関係に注目した。今回の研究は、その始まりとして、要求分析における表現形式の効果性を確認することを目的とした。効果性の確認のため、1 つのドメイン知識を 2 つの表現形式 (UML 図形式と用語辞書形式) で表して要求獲得実験を行い、その結果を測定した。実験の目的は以下の仮定を確認することである。

- 測定結果は、異なる表現形式の元では、違う結果となる。
- 測定結果は、分析者の直感に沿うような、合理性がある。

もしこの仮定が正しければ、得られた結果は表現形式の効果を確認できるものだと思われる。結果として、ほとんどの測定結果が表現形式において違いが見られ、その多くに合理的な説明ができた。例えば、UML 形式はオブジェクト間の通信や、後段の作業に関わる情報を必要とする問題の解決に向いていると思われたし、辞書形式は、情報の欠落を避けることに向いていると思われた。

今後の予定として、ドメインと表現形式との関係を明らかにすることに必要な、本論文では確認できなかった幾つかの問題がある。本論文では、1 つのドメインを複数の表現形式で表して獲得実験を行ったが、さらに、1 つの表現形式を使用して複数のドメインを表して実験を行うことが必要である。他に、表現形式だけではなく、他のドメイン知識の特徴についても同様な実験を行うべきである。同様のドメインカテゴリに属するドメインが同じ測定結果を持つかどうかについても確認する予定である。

謝　　辞

この研究は柏森情報科学振興財団の助成を受けて遂行された。

文　　献

[1] J. C. S. do Prado Leite and A. P. M. Franco. A Strategy for Conceptual Model Acquisition. In *Proceedings of First IEEE International Symposium on Requirements Engineering*, pages 243–246, 1993.

[2] H.-E. Eriksson and M. Penker. *Business Modeling with UML, Business Patterns at Work*. John Wiley & Sons, 2000.

[3] M. Jackson. *Problem Frames, Analyzing and structuring software development problems*. Addison-Wesley, 2000.

[4] H. Kaiya and M. Saeki. Ontology Based Requirements Analysis: Lightweight Semantic Processing Approach. In K.-Y. Cai, A. Ohnishi, and M. F. Lau, editors, *QSIC 2005, Proceedings of The 5th International Conference on Quality Software*, pages 223–230, Melbourne, Australia, Sep. 2005. IEEE Computer Society.

[5] A. Katasonov and M. Sakkinen. Requirements quality control: a unifying framework. *Requirements Engineering*, 11(1):42 – 57, March 2006. Springer-Verlag London Ltd.

[6] J. Kato, M. Saeki, A. Ohnishi, M. Nagata, H. Kaiya, S. Komiya, S. Yamamoto, H. Horai, and K. Watahiki. PAORE: Package Oriented Requirements Elicitation. In *Proceedings of 10th Asia-Pacific Software Engineering Conference (APSEC 2003)*, pages 17–26, Chiang Mai, Thailand, Dec. 2003. IEEE Computer Society Press.

[7] A. Mabbott and S. Bull. Alternative Views on Knowledge: Presentation of Open Learner Models. In *Intelligent Tutoring Systems: 7th International Conference, ITS 2004*, pages 689 – 698, Aug. 2004. LNCS 3220.

[8] N. A. M. MAIDEN and M. HARE. Problem domain categories in requirements engineering. *Int. J.Human-Computer Studies*, 49(hc980206):281–304, 1998.

[9] M. J. Muller, S. Kuhn, D. M. Wildman, and E. A. White. Participatory Design: Introduction. *Commun. ACM*, 36(6):24–28, Jun. 1993.

[10] A. Osada, D. Ozawa, H. Kaiya, and K. Kaijiri. Modeling Software Characteristics and Their Correlations in A Specific Domain by Comparing Existing Similar Systems. *IEICE Technical Note*, 105(128):1–6, Jun. 2005. SS2005-12 (In Japanese).

[11] H. Solheim, F. Lillehagen, S. A. Petersen, H. Jorgensen, and M. Anastasiou. Model-Driven Visual Requirements Engineering . In *13th IEEE International Conference on Requirements Engineering (RE'05)*, pages 421–428, 2005.

[12] W. Tracz. Domain analysis working group report: first international workshop on software reusability. *ACM SIGSOFT Software Engineering Notes*, 17(3):27 – 34, Jul. 1992.

[13] S. Viller and I. Sommerville. Social Analysis in the Requirements Engineering Process: From Ethnography to Method. In *Fourth IEEE International Symposium on Requirements Engineering (RE'99)*, pages 6–13, 1999.