

要求者支援のための記述解析システムASREDの開発

滝沢陽三 上田賀一

茨城大学

本研究では、日本語による要求仕様導出システムARDES/Jの記述解析サブシステムであるASREDの考察および実現について述べている。ASREDの目的は、要求者が要求を自然言語で仕様化できることである。要求者が要求を自然言語で定義し確認できれば、要求者の意図が開発者に伝え易くなる。しかし、現在の自然言語処理技術は極めて制限された処理しか行えない。本研究では、記述解析において構文解析による処理と単語による辞書検索を一緒に用いることで、記述から多くの情報が得られることを示す。この考えに基づいて実現したASREDのプロトタイプを示し、システム全体についての検討事項を報告する。

ASRED : the description analysis system for supporting requirer

Yozo Takizawa Yoshikazu Ueda

Ibaraki University

In this paper, we propose the requirement description analysis system, ASRED, which is a subsystem of the system for derivating specification, ARDES/J. The goal of ASRED is that a requirer can specify some requirements using natural language. The requirer will convey the idea more easely if he/she can define and confirm requirements by natural language. But current processing techniques of natural language can process only the restricted sentences. We describe that many informations about the specification are acquired using both the syntax analysis and the search for words in dictionaries. Under this consideration, ASRED is implemented as a prototype system, then we report some matters about a whole system.

1. はじめに

ソフトウェア開発過程のうち、初期の作業である仕様情報を抽出する要求定義段階はまだあいまいである。その作業のほとんどは開発者による手作業で行われ、要求者の要求が形式的かつ安定して定義されずに設計が進められることも多い。要求が十分に把握されずに開発が進めば、実装に時間がかかるだけでなく間違いも起こりやすく、開発者が要求者の意図しない製品を開発してしまう恐れもある。これらを解決するためには、要求者側である程度要求を仕様化できる手法と、要求者が適切に仕様を作るためのコンピュータによる支援システムが必要である。本研究では、この目的に沿ったシステムについて使われる手法と共に検討・考察を行い、日本語による要求仕様導出システムARDES/J(the system for Acquiring Requirement and DERivating Specification from descriptions written in Japanese)としてまとめた。

開発者ではない要求者が仕様化作業を行うためには、設計段階で用いられる記法ではなく、自然言語によって要求の形式化が行えるようにすべきである。しかし、コンピュータによって自然言語処理を完全にサポートすることは不可能であり、現状において、語彙的にも文法的にも極めて制限された処理しか行えない。また、自然言語処理には膨大な辞書情報が必要であり、より完全な解析を利用しようとするほど解析速度等の面で実用性に欠ける。更に日本語を扱う場合は日本語独特の表現に対する処理も加わり、実用的な支援システムへの適用を更に困難なものとしている。

ARDES/Jの記述解析サブシステムであるASRED(the Acquirement System of REquirement from Descriptions)は、この自然言語処理の限界を想定して構成されている。入力される記述に対しては字句・構文解析を行うが、解析が不可能な記述に対しては単語レベルで関連情報を検索し、記述に追加することで解析を行う。これにより、入力される記述が形式化されていなくても、ある程度の領域固有の情報(ドメイン知識)を付加できる。このためには、字句・構文解析のための情報も含んだドメイン知識を、記述分析とは別に獲得する手段、ドメイン分析手法・システムも必要になる。

自然言語を用いた要求定義手法・技術に関する

研究はこれまでも数多くあるが、本研究のように要求者側に焦点を絞ったものはほとんどない。

もちろん、自然言語を利用する目的の多くは要求者と開発者、あるいはグループ開発における開発者間の意志疎通を図るものであるため、参考になるものは多い。例えば、英語記述から手作業でオブジェクト構造を認識し設計するもの[1]、仕様書の記述・変換を統合して行うもの[2][3]、用語知識を再利用可能な辞書情報として蓄積する手法・ツール[4][5]、自然言語処理によって得られた単語情報からオブジェクト要素を認識するもの[6]などがある。これらは、安定した仕様や領域固有の情報を自然言語記述から作成、あるいは自然言語の形で蓄積する手法・システムとして注目できる。しかしこれらは、自動化が行えるほど手続き化されてなかったり、プログラミングや文書化の手間を省くといった目的に沿ったもの、手作業の負担の軽減に留まっているものがほとんどである。そして、これらはあくまで設計者の負担を軽くするためのものであり、要求者自身が分析・定義を行い、結果を確認することを前提としていない。

本研究の場合、入力を自然言語記述とするのは要求者の要求を開発に反映させるためである。すなわち、手法・システムは要求者支援が目的であり、設計者支援ではない。したがって、オブジェクト情報の獲得手法以上に、ソフトウェア開発という目的に沿った、要求者へのインタフェースとしての自然言語処理システムを確立することが重要となる。ここで紹介する記述処理システムASREDはそのインタフェースを提供するのに必要な記述解析・辞書管理・ドメイン知識獲得を実現するプログラムと、それらを統合して手法に沿って利用できるツールとして要求者に提供するユーザインタフェースから構成されている。

なお、ARDES/JはASREDと、モデル情報抽出サブシステムDESPER(the DERivation System of SPEcification from Requirement)の2つから構成される。DESPERは、ASREDの記述解析結果を元にモデル図を作成する機構を備え、後の設計段階のための、形式的手法で使われる図式的な仕様を導出する。現在はこの形式的手法として、オブジェクトモデル化技法(OMT)[7]を元に検討を行っているが、ASRED自体は記法に依存しない形で情報を出力するので、他の手法で用いられる記法に沿った仕様も導出できるようにすることを考えている。

またASREDも、方針としては辞書情報を用いた記述解析であるので、日本語を用いるシステムの開発を通して基本的な構築手法が得られれば、他の自然言語のための要求仕様導出システムの実現に活用できるものと考えている。

2. 日本語記述からの要求獲得・仕様導出システムARDES/J

ARDES/Jは上記で述べたように、大きく分けて2つのサブシステム(ASRED, DESPER)から構成される。また、各サブシステムもそれぞれ2つの部分に分かれる。両方のサブシステムとも、本来の解析・抽出作業部と、それらの作業のための辞書情報作成部から成り立っている。ARDES/Jの概観を図1に示す。

記述処理部ASREDは、実際に記述を変換する記述解析部と、その変換のための辞書を作成するドメイン辞書定義部から構成される。この変換は、記述に文章として読める形を残しつつ、コンピュータで簡単に図式表現できるよう簡略化するためのものである。ASREDの詳細は3章で述べる。

モデル情報抽出部DESPERは、簡略化された記述から仕様作成に必要な要素を導出する概念抽出部と、その概念抽出に必要な辞書を作成する概念定義部から構成される。辞書はここでは「文構造辞書」と呼び、特定の文構造あるいは特定の単文と、仕様構成要素を対応させたものが格納される。

例えば、「AはBである」という記述をOMTのオブジェクトモデルに沿って捉えたと(この例で捉えられる概念は特にOMTに限ったものではな

い)、「A: スーパークラス, B: サブクラス」という継承関係の他に、「A: クラス, B: インスタンス」というクラス-インスタンス関係が考えられる。後者のBはある特定のオブジェクトであり、単語としては固有名詞である場合であると思われる。そこで概念定義部では、「[一般名詞A]は「[一般名詞B]である」は「A: スーパークラス, B: サブクラス」を表す継承関係, 「[一般名詞A]は「[固有名詞B]である」は「A: クラス, B: インスタンス」というクラス-インスタンス関係を表すものである」と文構造辞書に登録する。概念抽出ではその辞書を利用して記述から各種の仕様構成要素を得、仕様を表す情報としてまとめて出力する。

なお、DESPERは記述から後の開発段階で必要な仕様を作成するシステムであるので、その利用は(要求者側ではなく)開発者側が行う。導出された仕様は開発者によってチェックされ、仕様として不完全な点やあいまいな部分が指摘される。入力される(簡略化された)記述には存在するが仕様として導出できなかった場合は、記述があいまいであるか辞書情報に登録されていない文構造をもつことが考えられる。また、構成要素として導出されたが他の導出された構成要素と矛盾する場合は、記述内容に問題があることになる。記述自体に問題がある場合はその再検討を要求者側に促し、辞書情報の欠落はその問題の記述を基に新たな概念定義を辞書に対して行うことになる。

3. 記述処理システムASRED

上述のように、ARDES/Jの記述処理サブシステムASREDは記述解析部とドメイン辞書定義部から

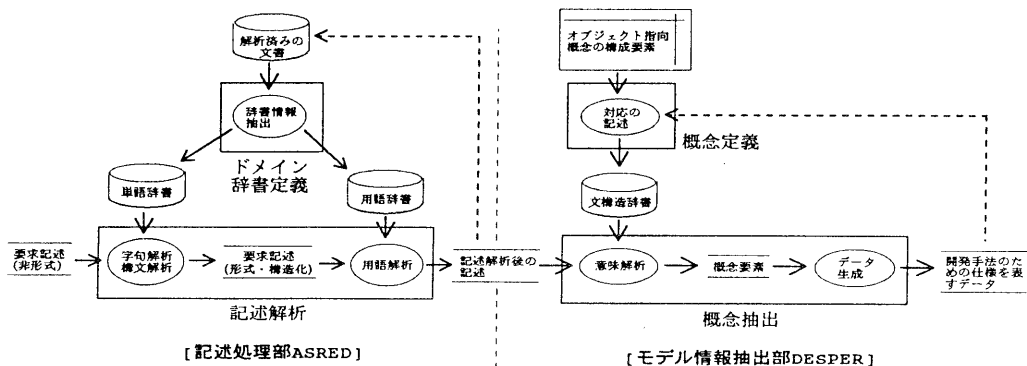


図1 日本語による要求仕様導出システムARDES/J

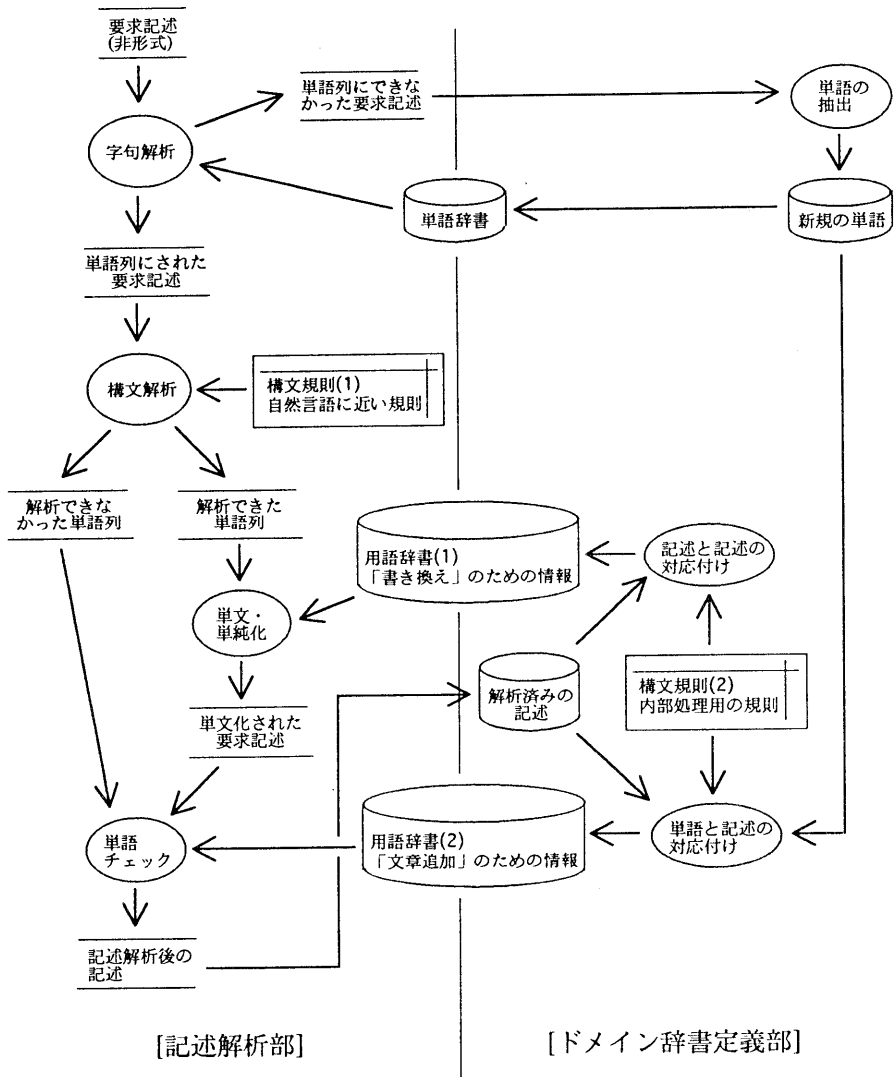


図2 ARDES/J記述処理システムASRED

構成される。ASREDの処理手順概観を図2に示す。ASREDでは3つの辞書が用いられ、記述解析部は辞書の利用によって解析を行い、ドメイン辞書定義部はその辞書の作成・更新等を行う。以降、図2に沿ってASREDにおける処理の流れを説明する。

3.1 記述解析部

記述解析部では、あらかじめ用意された各種辞書情報を元に、字句解析・構文解析による記述の構造化と、単文・単純化と単語チェックによる実

際の記述解析を行う。

字句解析では通常のコンパイラの字句解析同様、単語を認識して記述をトークン列に変換する。ただし扱う記述は自然言語であるので、トークンは品詞あるいは品詞に近い分類である必要がある。更に、本研究では日本語を扱うので、ベタ書きの記述を認識できなければならない。単語の認識には「単語辞書」によって行うが、あらかじめ用意された辞書情報のみでは認識できないかもしれない。その場合は、認識できない単語を含んだ文章は別に保存し、ドメイン辞書定義部に送って単語辞書情報の更新を促す。

構文解析では、構文規則に従って字句解析後の記述から構文を認識して文章を構造化する。文章には単文の他に、構文的に複数の単文要素を含む複文・重文もある。後の処理で記述を単文化するために、構文解析ではこれら複文・重文も認識する必要がある。なお、上述のように、自然言語処理にはまだかなりの制約があり、できるだけ自然言語に近い構文規則を用いたつもりでも、認識できない構文あるいは単語列が必ず存在するはずである。構文解析では解析できた記述と解析できなかった記述に分け、前者には単文化や単純化を行い、後者には単語レベルでの情報追加を行う。なお、単語レベルの情報追加は、前者の解析できた記述にも行われる。

単文・単純化では、構文解析できた記述全てに対し、単文要素をただ一つしかもたないようにすると共に、内部処理しやすくするためより簡単な構造をもつようにする。簡単な構造とはここでは「名詞・助詞・動詞」しか単語としてもたない文章のことで、「形容詞・副詞」等の修飾語は、名詞や動詞に関する何らかの「属性」を表していると思われる。したがってそれを表す記述を追加することになるが、このためには、その修飾語と修飾語に修飾されている単語から適切な記述を生成できなければならない。この生成のため、専用の用語辞書が必要である。また、後の段階で処理しやすいように、文の構造そのものを交換しておいた方がいい場合もある。ドメイン固有の表現が含まれる場合、それは必須である。この交換のための辞書情報も必要である。本研究では、この辞書情報を「書き換えのための用語辞書」として一つにまとめている。「接続詞」の類は単文化で処理される。

単語チェックでは、入力として上記の単文・単純化された記述と、構文解析で解析できなかった記述を受け付け、記述の構文とは無関係に、特定の単語に対し特定の文章を付加する。これは、特に不完全な構文をもつ記述や、固有名詞に対し有効である。この処理を行うためには、特定の単語と特定の文章をあらかじめ対応付けた情報が必要であり、ここでは「文章追加のための用語辞書」として用意されるものとする。単語チェックを経た記述は、ASREDの処理結果として出力される。

3. 2 ドメイン辞書定義部

ドメイン辞書定義部では、記述解析部であらかじめ必要とされる各種辞書情報を、登録すべきものとして認識された単語や記述を元に作成する部分である。ただし、「ドメイン」という言葉をここでは使っているが、一般に言われる「ドメイン分析」の手法とは違い、辞書情報はあくまで記述解析部での処理に必要なものでしかない。したがって、ある分野での知識を認識するため記述として表現するが、得られた辞書情報からドメイン領域を特定したりモデル化したりすることはない。

ドメイン辞書定義部は主に、単語の抽出、記述と記述の対応付け、単語と記述の対応付けの3つの処理から構成される。なお、これらの処理はエディタあるいはそれに類似した登録用ツールを用いて実際の処理を行うことはできるが、処理の自動化は想定されていない。これらの処理で抽出される情報は記述解析部の自動化を実現するためのものでもあり、領域固有の知識として手作業で作成する方が望ましいからである。

単語の抽出では、単語辞書に必要とされる単語をさまざまな記述から取り出し、新規の単語として登録したり、後の記述との対応付けに利用したりできるように整理する。単語を取り出す記述としては、記述解析部での処理をし易くするため、記述解析部の字句解析で認識できなかった文章を利用するべきであるが、システムを使い始めた頃はある分野に必要な単語情報がほとんどないことが予想される。したがって、基礎となる単語辞書構築のためには、字句解析部で必要とするかどうかの確認をせずに、ある程度の単語登録を(要求記述ではない)一般的な記述から得る必要がある。

記述と記述の対応付けでは、記述解析部の単文・単純化に必要な情報を作成する。上述のように、文章構造を単純化したりドメイン固有の表現を交換したりするための、「書き換えのための用語辞書」として作成を行う。内容は記述と記述の対応表であり、以下のようなものが考えられる。

- (1)ある構造をもった文章と、それと同じ意味を表すより簡単な構造をもつ文章群
- (2)ある構造の文章とそれに出現する修飾語とその被修飾語の組み合わせと、それと同じ意味を表

すより簡単な構造の文章群およびその修飾・被修飾語のない文章構造をもつ文章

- (3) (1)と似ているが、簡単な構造をもってはいるがドメイン固有の表現である文章と、同じ意味を表すドメイン固有の表現ではない文章

それぞれの文章のうち、後者の文章は一般的な表現でより簡単な構造をもつ必要があり、他の、似通った領域の解析済みの記述を参考に作成されるべきである。しかし単語辞書作成と同様、システムを使い始めた頃は辞書情報がほとんどなく、したがって最初は解析済みの記述ではない文書を参考に作成する必要がある。

単語と記述の対応付けでは、記述解析部の単語チェックに必要な、単語と記述の対応表を「文章追加のための用語辞書」として作成する。ここで登録される情報は1つの単語のみから追加の判断ができるドメイン固有の知識であり、要求記述を書いた要求者が暗黙にもつ知識である場合が多い。よって、作成対象の単語は「単語の抽出」で得た新規の単語の他に、単語辞書作成や記述変換用辞書作成と同様一般的な文書にある単語、更には要求者と開発者との質疑等を元に基礎的な辞書を作成する必要があるかもしれない。

4. ASREDの実現

実現は、既存の自然言語処理システム[8]等を用いず、広く知られている基礎的な自然言語処理技術[9][10][11]を基に行った。なぜなら今回の目的にはプロトタイプとして、実現において手法がどんな情報と処理を必要とするかを明確にすることがあったからである。また、既存のシステムを用いるとそのシステムで利用されている概念や出力データに合わせて目的のシステムを構築しなければならず、「手法の有効性」も確認しにくい。もちろん、プロトタイプを通してその有効性が確認できれば、実用化に向けて既存の優れたシステムを利用することになると考える。

プロトタイプは記述解析部とドメイン辞書定義部を処理する2つのツールとして実現されている。図3が記述解析部、図4がドメイン辞書定義部でのツールの利用例である。なお、字句解析や記述変換の実現にはSchemeを用い、構文解析は便宜上 yaccを用いた。使用した構文規則は、構文解析に

```

<文> ::= <単文> . | <単文> <文>
      | <単文> 接続助詞 <文>
      | 接続詞 <文> ;
<単文> ::= <修飾部> <述部> ;
<述部> ::= 動詞 ;
<修飾部> ::= <修飾> | <修飾> <修飾部> ;
<修飾> ::= 副詞 | <体言> 助詞 ;
<体言> ::= 名詞 | <連体部> 名詞
          | 名詞 並列助詞 <体言>
          | <連体部> 名詞 並列助詞 <体言> ;
<連体部> ::= 連体詞 | 形容詞 | 形容動詞 ;

```

図5 構文解析に用いる構文規則

```

<記述> ::= ε | <記述> <単文> . ;
<単文> ::= <名詞句列> 動詞 ;
<名詞句列> ::= <名詞句>
             | <名詞句列> <名詞句> ;
<名詞句> ::= 名詞 助詞 ;

```

図6 解析結果の記述を規定する構文規則

用いるものを図5に、ドメイン辞書定義に用いて解析結果の記述を規定するものを図6に示す。GUI部はSTk、エディタにはMule、ビューワにはTcl/Tkで実現したものを利用した。

図3の(1)は記述解析部のメインウィンドウであり、最初に入力される要求記述および最終的に出力される解析済み記述のファイル指定と、実際の解析処理を行うサブウィンドウ起動ボタンから成る。ファイル指定ではそれぞれ修正・表示のためにエディタやビューワが起動できる。(2)が解析処理のサブウィンドウであり、各処理段階の指定と処理の途中結果が表示される。処理段階の指定後に処理ボタンを押すと、指定済みのファイル名を入出力として各種プログラムが起動する。(3)は要求記述、(4)は単文化処理の例である。

図4の(1)はドメイン辞書定義部のメインウィンドウであり、参照する記述および定義すべき辞書のファイル名の指定と、それぞれに対する処理プログラムを起動するメニューから成る。辞書情報追加においては登録のための「枠組み」がエディタによって呼び出され、必要な情報が必要な項目を埋めることによって登録される。

今回は上述の通り簡単な実現に留めたため、処理プログラム・構文規則・各種枠組み等は記述解析のためのものとして、内容的にも性能的にも不

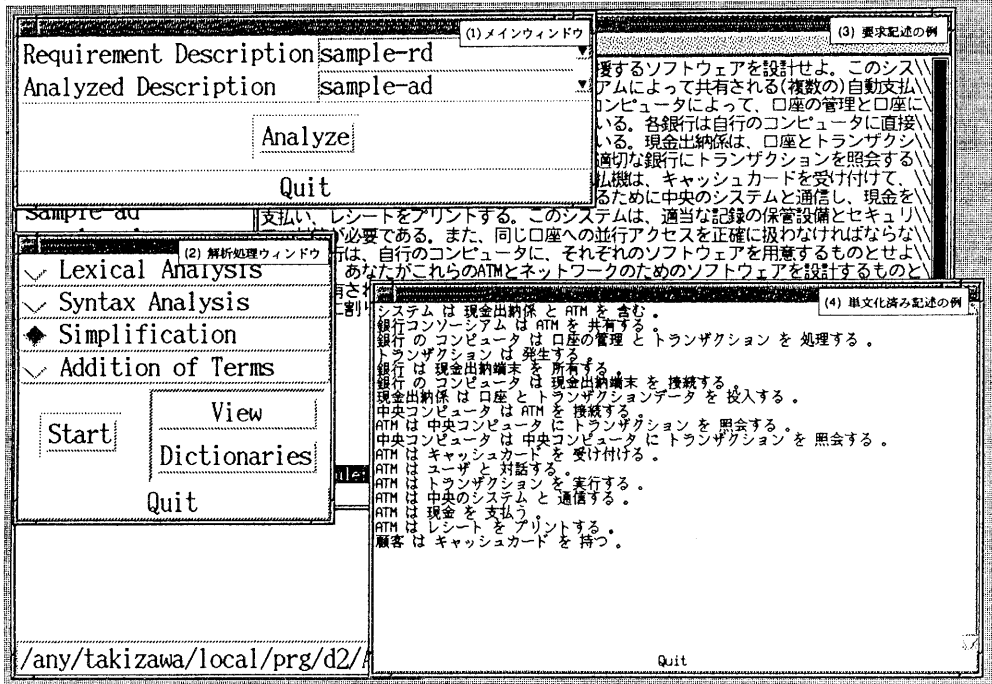


図3 ASRED記述解析部実現例

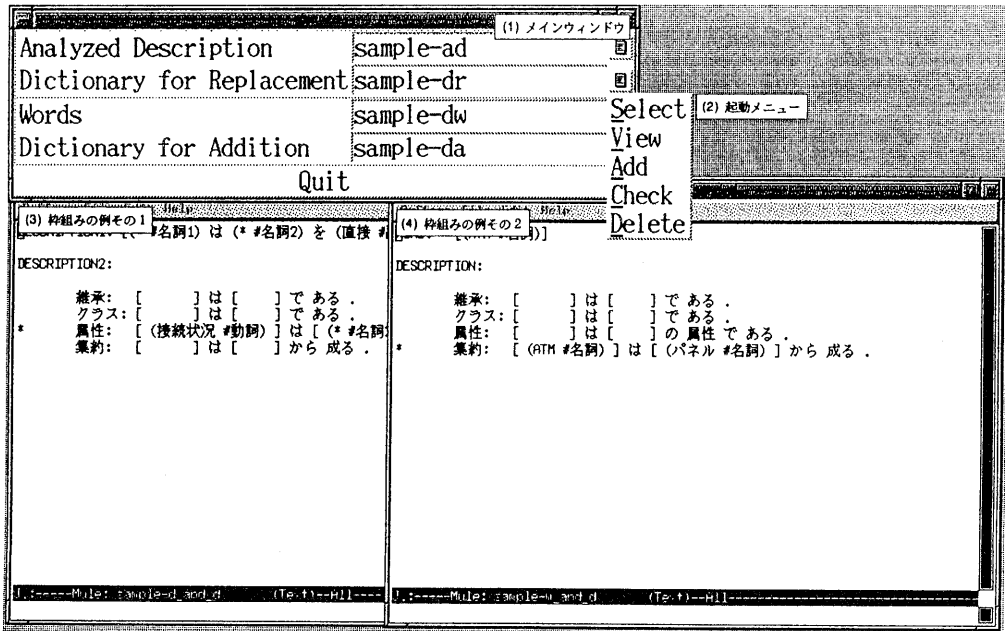


図4 ASREDドメイン定義部実現例

十分なものが多い。しかし、実現モジュールとして何が必要か、途中経過を示すべき情報は何か、どのように作業を進めていくことになるかが具体的に確認できた。

5. まとめ

本研究では、仕様導出システムARDES/Jの概観を紹介し、記述解析システムASREDの詳細な手順と簡単な実現例を示した。これにより、手法の体系付けとASREDの各種実現上の確認を行うことができた。特に実現部分では、自動化を含めて、どの部分でどのような処理が必要かがはっきりした。記述解析部は処理の確認のためにいくつか対話的な機構が必要ではあるが、処理自身は辞書情報の充実と共に、要求記述の読み込みから解析結果出力までを一通り自動的に行う必要がある。ドメイン辞書定義部では、エディタやビューワによる手作業が中心となるが、矛盾のない辞書情報を大量に登録するためには、「枠組み」と枠組みに埋め込まれた情報の整合性チェックの処理が重要となる。

課題もいくつか明確にされた。特に、適用例は実用性を確認する上で不可欠である。今回のプロトタイプが通常の要求記述に対しどこまでの処理を可能にしているかは、大量の辞書情報登録を含め、ある程度の規模で試用しなければ確認できないだろう。更に、一般的に言われている「開発分野の違い」が、辞書情報や処理内容にどの程度影響を与えるかを調べる必要がある。それを踏まえ、システムの各構成要素にどれだけ実用的なシステムを実現あるいは既存のものを利用するかを考察しなければならない。

もう一つの課題は、モデル情報の抽出を行うDESPERの作成とARDES/JとしてのASREDとの統合である。ASRED単体のみでは、要求者は自身の要求を整理あるいは詳細化できるが、その出力結果だけでは開発者側が開発上の整合性等を確認できない。ASREDとDESPERを組み合わせ、ARDES/Jとして手法的にもツールとしても統合して初めて、要求者と開発者の間の溝を埋めるため

のシステムとして利用できる。DESPERに関しては、簡単な構造をもつ記述をモデル情報に変換するプログラムを試作しているが[12]、現段階ではさまざまな開発手法のためのモデル図に変換するだけの柔軟性は備わっていない。各種手法で使われる記法の検討が必要である。

参考文献

- [1] 佐伯元司, 蓬萊尚幸, 榎本 肇: 自然言語からモジュール構造を得る手法について, 情報処理学会論文誌, Vol.30, No.11 (1989).
- [2] 大西 淳: ソフトウェア要求定義のためのコミュニケーションモデル, 情報処理学会論文誌, Vol.33, No.8 (1992).
- [3] 磯田定宏, 黒木宏明: 統合化CASEシステムSoftDAの機能, コンピュータソフトウェア, Vol.10, No.2 (1993).
- [4] 安田 剛, 田中誠一郎, 他: 用語知識再利用による要求分析支援手法(I)(II), 情報処理学会第42回全国大会講演論文集(5), pp.177-180 (1991).
- [5] 井上秀行, 山田 淳, 山城明宏, 井上勝博: 要求分析支援ツール JOKER'91, 情報処理学会第44回全国大会講演論文集(5), pp.333-334 (1992).
- [6] 大野雅志, 原田 実: オブジェクト指向分析支援システムCAMEO - 日本語文章記述からの設計要素の自動抽出 -, 情処研報SE-99-14, pp.105-112 (1994).
- [7] J.ランボー他著, 羽生田栄一監訳: オブジェクト指向方法論OMT, トップラン (1992).
- [8] 松本裕治, 黒橋禎夫, 宇津呂武仁, 妙木裕, 長尾真: 日本語形態素解析システムJUMAN使用説明書 version 1.0, 京都大学工学部長尾研究室(1993).
- [9] 淵 一博 監修: 自然言語の基礎理論, 知識情報処理シリーズ第4巻, 共立出版(1986).
- [10] R.グリシュマン, 山梨正明・田野村忠温共訳: 計算言語学 コンピュータの自然言語理解, サイエンス社 (1989).
- [11] 草薙 裕: LISPによる自然言語処理, 工学図書 (1988).
- [12] 滝沢陽三, 上田賢一: オブジェクト指向に基づく要求記述からの形式的仕様の導出手法, 情処研報SE94-8 (1993).