

セミオティックベースを使った日常言語アプリケーションシステム (第2報)

Everyday Language Application Systems with the Semiotic Base (Part 2)

伊藤 紀子^{*1} 杉本 徹^{*1} 岩下 志乃^{*1} 岩爪 道昭^{*1} 高橋 祐介^{*1} 小林 一郎^{*1,*2} 菅野 道夫^{*1}
 Noriko Ito Toru Sugimoto Shino Iwashita Michiaki Iwazume Yusuke Takahashi Ichiro Kobayashi Michio Sugeno

^{*1} 理化学研究所 脳科学総合研究センター
 Brain Science Institute, RIKEN

^{*2} お茶の水女子大学 理学部
 Faculty of Science, Ochanomizu University

Under the name of Everyday Language Computing paradigm, we implemented three prototypes of language-based application systems as flagship components of a language-based operating system. One enables users to operate a word processor through everyday language. Another allows users to write computer programs through everyday language. The other plays a role as Question-Answer system, which refers to help texts attached with software and provides an answer that varies according to user's level of expertise and preferred words. All the systems use the Semiotic Base, which is a linguistic resource that deals with the relationship among context, meaning and wording of texts in their specific tasks as well as in text understanding and generation.

1. はじめに

我々は、日常的な言語を使ってさまざまな情報処理を行う日常言語コンピューティング (ELC: Everyday Language Computing) と呼ばれる新しいパラダイムを提案し、その実現を目指して研究を進めている [岩爪 03]。日常言語コンピューティングシステムでは、ユーザは日常言語を使ってクライアント秘書エージェントと話し、クライアント秘書は、ユーザの指示や希望を日常言語の形式に基づく通信プロトコルである言語プロトコルの形でアプリケーションソフトに送信し、それを受け取った言語アプリケーションは、指示の意味を理解して処理を実行する。これら一連の流れを実現する枠組を我々は言語オペレーティングシステム (LOS: Language-based Operating System) と呼んでいる。LOS における言語の意味の理解と生成の処理はすべて、選択体系機能言語学 (systemic functional linguistics) で提唱されている言語モデルに基づいて設計されたセミオティックベース (Semiotic Base) という言語資源を利用して行われる。言語アプリケーションによる処理の途中結果や用いられる知識はすべて、語彙文法特徴、意味特徴、概念構造と結びつけられた日常言語テキストとして表現される。

すでに我々は、セミオティックベースのプロトタイプとそれを使ったテキスト理解・生成アルゴリズムの設計・実装を行い、さらにその応用として、日常言語によってワープロ操作やメール管理を行うことができる言語アプリケーションシステムの試作システムを実現し、[杉本 03] でこれらのシステムの構成と処理の内容について説明した。

今回我々が実現したシステムは、スマートヘルプシステム、日常言語プログラミングシステム、言語ワープロシステムの3つで、これらのうち日常言語プログラミングシステムと言語ワープロシステムについては、[杉本 03] の試作システムを発展させたものである。本論文では、これらのシステムの構成と処理の内容について、追加・拡張した機能やコンテンツを中心に説明を行い、残された課題について述べる。

2. 日常言語コンピューティング

以下に、日常言語コンピューティングの基本原則別に、今回我々が実現した要素技術をまとめる。

(1) セミオティックベースを用いたテキスト理解と生成

プロトタイプでは十分に扱われていなかった、ユーザからの入力文やマニュアルテキストの修辭構造解析と、その結果をユーザへの応答文や実行可能な日常言語プログラムの生成に利用する方法の実装に重点をおいた。

また、テキスト生成時に、コンテキストベースの資源を利用することにより、発話のモチベーションを導出し、それに基づいてテキスト生成を行う仕組みを追加した。

(2) 言語知識・コーパスの利用

LOS やアプリケーション内部の知識、手続き、機能、動作などを言語で記述しておき、その意味を理解して、処理に活用するための手法や、ヘルプファイルやマニュアルなど既存の「言葉で書かれた知識」の意味を理解し、ユーザに適切なヘルプを提供するための手法を実装した。

(3) 意味のマッチング

より柔軟なソフトウェアの制御を実現するために、テキスト理解の結果をアプリケーション操作に結びつける際に、セミオティックベース内の特にコンテキストベースの資源を利用したマッチングの実装を行った。

今回実装したスマートヘルプシステムと日常言語プログラミングシステムでは主に(1)、(2)を、言語ワープロシステムでは主に(1)、(3)に焦点を当ててシステムの実装を行った。次節以降で、その詳細について説明する。

3. セミオティックベースとテキスト処理

セミオティックベースは4つの主要コンポーネント(コンテキストベース、意味ベース、語彙文法ベース、表現ベース)と2つの補助コンポーネント(電子化辞書、コーパスベース)からなる。セミオティックベースとそれを用いたテキスト理解・生成の処理の詳細については [伊藤 03] を参照されたい。

今回、より状況に即した処理を実現するために、コンテキストベースのコンテンツの拡充を図った。コンテキストベースは、言葉がやりとりされる状況のタイプのタクソミーとその特徴を記述するための活動領域・役割関係・伝達様式の選択体系網からなる状況ベースと、対話のやり取りのパターンを納めたステージベース、そして状況依存の概念をフレーム形式で定義した概念辞書の3つの部分から構成される。

例えば、表 1のユーザと秘書の対話において、秘書が2つ目の問いを発するためには、ユーザの発話から得られた二箇所の送付先に関して、ユーザとの社会的関係、特にユーザと送付先との社会的距離と階層関係とが異なり、同一の文面の暑中見舞いを送るのが不自然である、という判断を下さなくてはならない。

表 1:ユーザと秘書の対話例 1

秘書	暑中見舞いをどなたに送りますか？
ユーザ	友人と会社の同僚です。
秘書	暑中見舞いの文面を変えますか？

このような判断を行うための、コンテキストベースの資源とコーパススペース内にある言語で書かれた知識である状況記述知識と状況内知識などを用いたテキスト生成プランニングの処理の詳細については、[高橋 04]を参照されたい。

修辞構造解析については、[Mann 88]の修辞構造理論に則って人手でマニュアルテキストの分析を行い、その結果から、接続詞や接続助詞以外にも、修辞関係ごとに述部に現れる用言のリストや段落番号や改行などの文書構造に基づいたルールを抽出した。修辞構造解析の結果を利用する方法については、この後で簡単に述べる。

4. 言語アプリケーション

4.1 スマートヘルプシステム

スマートヘルプシステムは、図 1に示すように、ユーザとクライアント秘書エージェントの対話型になっており、ユーザと対話しながらコンピュータ操作に関するヘルプを提供する。

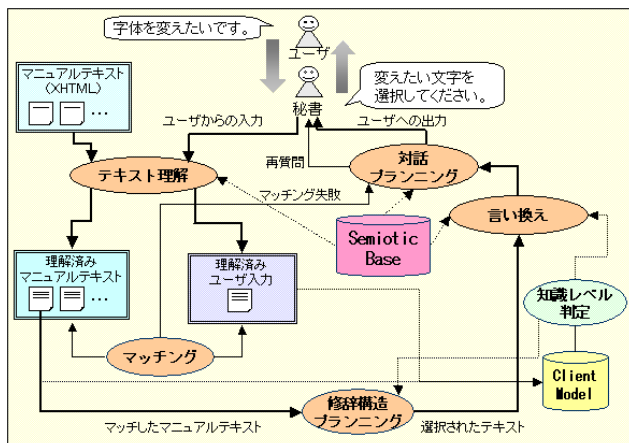


図 1: ヘルプシステムの処理過程

本システムは理解、マッチング、プランニング、言い換えの 4つの処理過程から成る。マニュアルテキストは予めセmiotickベースによって理解され、利用可能な形で保持してある。まず、ユーザが入力を行うと、セmiotickベースはその入力テキストの理解を行う。次に、ユーザ入力とマニュアルテキストの理解結果を用いて、質問テキストに合ったマニュアルテキストが検索される。そして、マッチングで複数のマニュアルが検索されたとき

に一つに決定したり、選ばれたマニュアルテキストからユーザが必要としている一部のテキストを抽出したりといった、プランニングの処理を行う。最後に、決定されたテキストがセmiotickベースとクライアントモデルの情報を用いて言い換えられ、ユーザに出力される。

表 2:ユーザと秘書の対話例2

	対話	言い換え操作
ユーザ	字体を変えたいです。	
秘書	変えたい文字を選択してください。	ムード変化, 類義語
	ツールバーに MS 明朝と書いてあるところがあります。	分割, 類義語
	そこでフォント名をクリックして、字体を変えてください。	補完, 類義語, ムード変化

表 3:マニュアル「フォントを変更する」

手順	テキスト
1	変更する文字列を選択します。
2	[書式設定]ツールバーの[フォント]で、フォント名をクリックします。

このシステムの特徴は、知識ベース内のテキストをそのまま出力するのではなく、クライアントモデルを利用して個々のユーザに特化した形に言い換えて出力すること、知識ベースとして、ソフトウェアに付属のマニュアルテキストを利用することである。例えば、表 2にある秘書の発話は、表 3にあるマニュアルテキスト部分に言い換えルールを適用することによって生成されたものである。

言い換え操作は、被験者実験の結果と、マニュアルテキストの分析結果から、分割、補完、ムード変化、類義語選択の 4つを設定した。分割とは、マニュアルテキスト内に場所を示す名詞句や連体修飾節がある場合に、それをひとつの文として独立させて、まず着目すべきアイコンの場所を説明してから、残りの部分で操作の説明をする場合である。補完とは、マニュアルテキストの手順を最後まで説明した後に、タイトル文を付け足すことによって、説明された手順によってもたらされる結果をユーザに明示し、操作が成功したかどうかの確認を促す場合である。ムード変化とは、書き言葉で独白的であるマニュアルテキストを、より話し言葉的で対話的なものにするために、文末表現を「ーします」から「ーしてください」などに変える場合である。類義語選択とは、マニュアルテキストに含まれている語彙を、ユーザ本人がよく使う語彙や専門度の低い語彙に置き換える場合である。

他の特徴としては、マッチングによってユーザ質問に対する答えとなるマニュアルが検索された後、マニュアルテキストの修辞構造を利用して、その中からユーザが必要としているテキストを抽出している点である。テキストを抽出するためのルールはユーザの知識レベルごとに決められており、例えば初心者には分かりやすい説明にするために付加的な情報をなるべく減らすようにしている。表 4にルールの一部を示す。

表 4:修辞構造に基づいたテキスト決定ルール

修辞構造	ルール
solutionhood	Nuclues のテキストのみを対象とする。
sequence	全てのテキストを順番に対象とする。
conjunction	条件節とユーザ入力のマッチングを行い、最も点数が高いテキストを対象とする。

試作システムでは、対象ドメインとしてワープロ操作の中でも特に初心者ユーザからの質問が多い文字列の操作を取り上げ、

文書作成しているユーザからの文字列操作に関する質問を理解して、あらかじめ言語的な分析を施しておいた Microsoft Word 付属のヘルプを元に、適切なヘルプテキストを提供する。詳細については、[岩下 04]を参照されたい。

4.2 日常言語プログラミングシステム

日常言語プログラムシステムは、表 5にあるような、複雑な操作手順や実行条件を含む要求仕様を表している複数の文からなる一貫した入力テキスト(日常言語プログラム)を処理して、応用ソフトウェアの操作を実行することができる。

表 5: 日常言語プログラムの例

- (1) 理研の人から来たメールは「理研」フォルダに入れる。
- (2) 理研の人のメールアドレスは、末尾に「.riken.jp」が付く。
- (3) 研究室の人からのメールは、「研究室」フォルダに入れて自宅に転送する。
- (4) ただし、伊藤さん、高橋さん、小林さんからのメールでセミオティックベースに関するものは、「SB」という名前のフォルダに入れる。
- (5) この時、添付ファイルがあったらマドキュメントの「SB」という名前のフォルダに保存する。
- (6) また、添付ファイルが文書ファイルの場合は、それを表示する。

入力されたプログラムは一般に、実行可能な形、すなわち言語アプリケーションのコマンドと明示的な制御構造の組み合わせの形にはなっていない。そこで、コマンドに関する知識を参照して各文の実行可能性を検査し、必要に応じて、換言規則と事実に知識を用いた換言の処理を行う。このプログラム正規化の過程で、文中の暗黙的なループ構造の検出も行われる。その後、入力プログラムの修辞構造を考慮しながら各文の正規化の結果を統合し、実行可能な形をしたプログラムテキストを生成する。図 4は、表 5に挙げた入力プログラムから生成された実行可能プログラムである。最後に、言語アプリケーションのコマンド実行機能を使って、応用ソフトウェアの操作を実行する。

```

受信メールボックスの中すべてのメールに対して、次の処理を繰り返す。
もし、そのメールの差出人が伊藤さん等しい、または、かつそのメールがセミオティックベースに関するならば、次の処理を行う。
そのメールを「SB」フォルダに入れる。
もし、添付ファイルがあるならば、その添付ファイルをマドキュメントの「SB」フォルダに保存する。
もし、添付ファイルが文書ファイルであるならば、その添付ファイルを表示する。
以上。
そうでなければ、もし、そのメールの差出人が菅野先生と等しい、または、ならば、次の処理を行う。
そのメールを「研究室」フォルダに入れる。
そのメールを自宅に転送する。
以上。
そうでなければ、もし、そのメールの差出人アドレスの末尾に「.riken.jp」が付くならば、そのメールを「理研」フォルダに入れる。
以上。
    
```

図 4: 生成された実行可能プログラム

日常言語プログラムの処理モデルの概要を図 2に示す。この処理モデルは、ELC の枠組みにおいて言葉を使って応用ソフトウェアを操作する言語アプリケーションの一機能と位置付けられている。

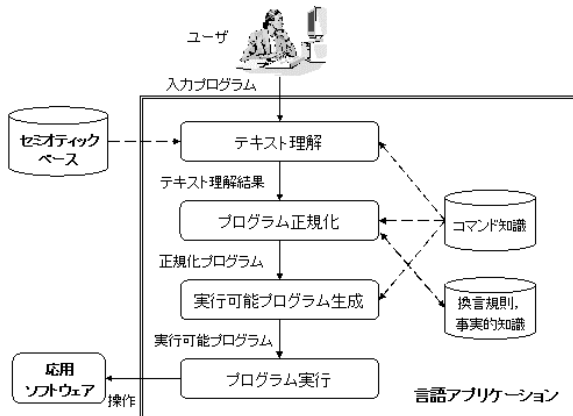


図 2: 日常言語プログラムの処理モデル

ユーザが入力した日常言語プログラムに対して、まずセミアティックベースを使ってテキスト理解処理を行い、各文の語彙文法的、意味的、概念的構造と、プログラム全体の修辞構造を同定する。図 3は、表 5に挙げた日常言語プログラムの修辞構造である。

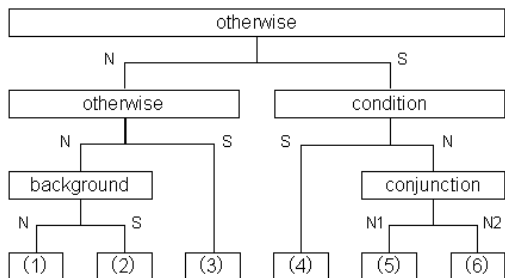


図 3: 日常言語プログラムの修辞構造

試作システムでは、対象ドメインとして電子メール管理を取り上げ、電子メール管理に関するユーザの指示を理解して Microsoft Outlook の操作を実行する。[杉本 03]と比較すると、修辞構造解析で扱える修辞関係の種類や木構造のパターンを増やし、プログラムの正規化をする際に参照するコマンド知識や換言規則・事実に知識の整備と拡充を行ったことにより、より複雑な日常言語プログラムを処理できるようになった。また、対話機能を追加することにより、以前に作成したプログラムの内容に関する質問や、内容変更の要求を受け付けることができるようになった。詳細については、[杉本 04]を参照されたい。

4.3 言語ワープロシステム

言語ワープロシステムに関する研究の狙いは、既存アプリケーションソフトを日常言語によって制御実行するための方法を確立することにある。言語ワープロシステムは、既存のワードプロセッサに、言語アプリケーションインタフェース(LAPI)と呼ばれる API によって言語化された日常言語アプリケーションの一種である([岩爪 03])。

(1) 言語アプリケーションインタフェース(LAPI)

LAPI は、言語プロトコルによるクライアント秘書や他のエージェントからのメッセージを受信・理解するための外部インタフェース、日常言語で表現された内容を実行可能な形式に展開する意味マッチングモジュール、展開された自然言語表現系列を対象アプリケーションの実装レベル(コマンド)に変換する内部インタフェースによって構成される(図 5 参照)。マッチングモジュールには、LAPI 内部知識と呼ばれるアプリケーションに関する知識が格納されており、外部からの要求を実行可能な実装レベルの形式に展開したり、アプリケーションの実行結果を外部へ報告する際に、形式表現から言語表現へ変換するために用いられる。

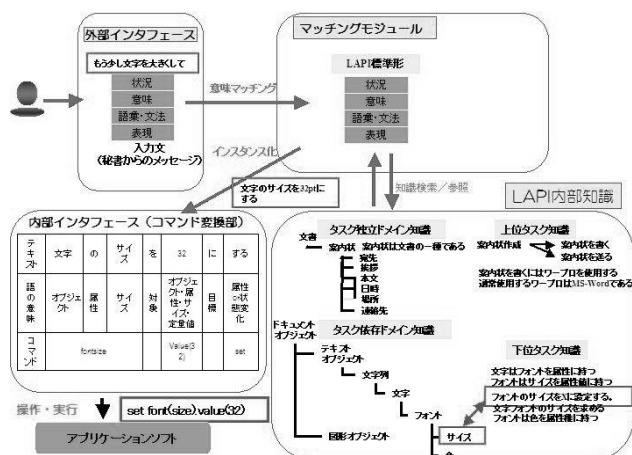


図5 言語ワープロシステムの構成

(2) LAPI 標準形

LAPI 標準形は、自然言語によって表現された対象アプリケーションの操作・制御に関するモデルである。図 6 に LAPI 標準形の例を示す。ここで、< > で囲まれた部分は、操作対象オブジェクトのクラス名、サブクラス名、属性名、値などを指し示している。上から1行目は、最も汎化された LAPI 標準形の1つである。2行目は、具体的なオブジェクトクラスが指定されたクラス表現になっている。一方、3行目は、< 値 > のスロットに、定性値として「大きい」という言語表現が入っているインスタンス表現である。これは、クライアント秘書を介して言語アプリケーションに送られてくる指示内容が、「文字も少し大きくして欲しい」といった定性的かつ曖昧な表現になるケースが少なくないからである。このような言語による定性値の表現を媒介して、最終的に4行目のインスタンス表現に具体化することになる。このインスタンス表現まで具体化されると、内部インタフェースに受け渡され、コマンド変換表に基づいて、実際にアプリケーションが操作・実行される。

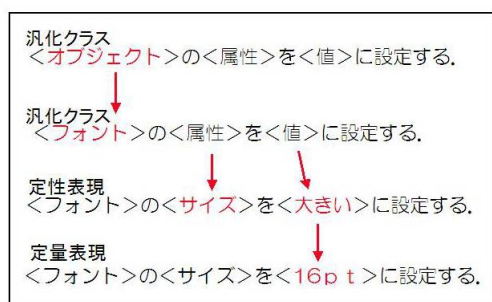


図6 LAPI 標準形

(3) 意味マッチング

本研究では、意味マッチングをユーザ入力文から LAPI 標準形のインスタンスを生成するプロセスとして従えている。昨年度のシステムでは、このプロセスにおいて主に LAPI 内部知識を積極的に利用したマッチング方式を検討したが([岩爪 03])、今年度は、状況ベースに格納されている様々な知識源([高橋 04])を利用する方法について検討した。具体的には、状況タイプ・ジャンル構造を利用することでマッチングの対象となる LAPI 文を絞り込みを行ったり、状況層に含まれるテナー情報とドメイン知識の概念を予め関連付けておくことで、より状況に即した柔

軟な文書コンテンツの操作が可能になった(詳細は[岩爪 04]参照)。

5. まとめ

日常言語コンピューティングの中核要素である言語アプリケーションの具体例として、スマートヘルプシステム、言語プログラミングシステム、言語ワープロシステムの3つを実装した。[杉本 03]の試作システムで実現されていた要素技術と比較すると、コンテキストベースのコンテンツを構築し、それを利用することによって、テキスト生成や意味マッチングを拡張し、より発話の状況に即した処理が出来るようになった。また、[杉本 03]では扱われていなかった、言語知識・コーパスの利用を実現し、スマートヘルプシステムと言語プログラミングシステムに組み込んだ。

今後の課題として、言語アプリケーションの対象ドメインを広げることと、そのための資源構築環境の整備があげられる。また、LOS のミドルウェア層の整備により、セミオティックベースおよび言語プロトコルとの連携を強化したい。

参考文献

- [岩爪 03] 岩爪道昭, 小林一郎, 杉本徹, 岩下志乃, 高橋祐介, 伊藤紀子, 菅野道夫: 日常言語コンピューティング(第2報)—日常言語に基づく計算機資源の管理・実行環境を目指して—, 人工知能学会論文誌, Vol. 18, No. 1, pp. 45-56 (2003)
- [杉本 03] 杉本徹, 岩爪道昭, 小林一郎, 伊藤紀子, 高橋祐介, 岩下志乃, 菅野道夫: セミオティックベースを使った日常言語アプリケーションシステム, 第 17 回人工知能学会全国大会, 新潟(2003), 3G1-06
- [伊藤 03] 伊藤紀子, 杉本徹, 高橋祐介, 小林一郎: セミオティックベースを使ったテキスト処理アルゴリズム, 第 17 回人工知能学会全国大会, 新潟(2003), 3B1-01
- [高橋 04] 高橋祐介, 小林一郎, 岩爪道昭, 菅野道夫: コンテキストベースのコンテンツと構造—セミオティックベースにおけるコンテキスト層の再検討—, 第 18 回人工知能学会全国大会, 金沢(2004), 3E2-01
- [Mann 88] Mann, W. C. and Thompson, S. A.: Rhetorical Structure Theory: Toward a Functional Theory of Text Organization, *Text*, Vol. 8, No. 3, pp. 243-281 (1988)
- [岩下 04] 岩下志乃, 伊藤紀子, 小林一郎, 杉本徹, 菅野道夫: マニュアルテキストを用いた個人化ヘルプシステム, 第 18 回人工知能学会全国大会, 金沢(2004), 3E2-04
- [杉本 04] 杉本徹, 伊藤紀子, 岩下志乃, 菅野道夫: 日常言語によるプログラミング, 第 18 回人工知能学会全国大会, 金沢(2004), 3E2-03
- [岩爪 04] 岩爪道昭, 小林一郎, 高橋祐介, 菅野道夫: ソフトウェア制御のための状況を考慮した動的意味マッチング, 第 18 回人工知能学会全国大会, 金沢(2004), 3E2-02