

ユーザの目的達成を支援する知識をもった操作インタフェースの開発

Development of an Operation Interface equipped with Knowledge that Supports to Achieve User's Goal

大西可奈子*1
Kanako ONISHI

小林一郎*1
Ichiro KOBAYASHI

*1 お茶の水女子大学理学部
Ochanomizu University Faculty of Science

This paper proposes a new interface, equipped with knowledge that supports to achieve user's goal, that can operate plural application software simultaneously with natural language. In order to decide which application software should be operated by given natural language instruction, we have proposed dictionaries to identify the concrete value of how much a certain vocabulary is related to each application software. By this, it has become able to operate plural application software with one interface by referring to this value.

1. 研究背景と目的

近年、コンピュータの価格低下やインターネットの普及により、コンピュータを使用する機会が格段に増加している。それに伴い、さまざまな機能を備えたアプリケーションソフトウェアが提供されるようになってきた。これは一見すべての人に利便性を与えているように見えるが、実際は多くの人々がソフトウェアを使いこなせていない。

近い将来、情報家電が家庭に入ってきた時に、それぞれの複雑な操作をひとつの情報集約端末（例えば癒しの効果を持つペット型ロボットなど）を使って自然言語で操作できるようなインタフェースの必要性が高まると言える。このようなことを踏まえ、本研究では、言語指示で複数のアプリケーションソフトウェア（本研究においては MS Word, MS Excel, MS Outlook を対象にする）を操作するためのインタフェースの開発を目的とする。

2. システム概要

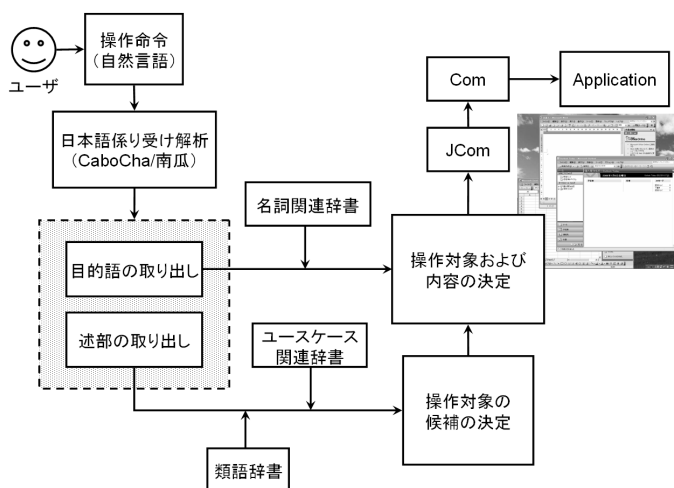


図 1: システム概要

送りたい」という入力文に対して、述部として「送る」を、目的語として「メール」を抽出する。

2.2 類語辞書とユースケース抽出

本研究では、類語辞書を用いて述部からユースケースを抽出する。類語辞書とは、同義語や類義語、さらに本研究で使ったアプリケーションソフトウェアを使用する際に、同等の意味で使用されると思われる述部をひとつの述部に集約するための辞書で、その集約先の述部がアプリケーションソフトウェアの操作のユースケースとなっている。ユースケースはアプリケーションソフトウェアのコマンドに直接結びついているため、ユースケースを決定することにより各アプリケーションソフトウェアに対して実行候補であるコマンドが決定する。最終的に、これらのコマンドのいずれかが実行される。

例えば、「ワードを動かして」という命令文に対しては、述部「動かす」が「始」に集約され、この命令文に対するユースケースは「始」に決定する（図 2 参照）。このユースケース「始」はアプリケーションソフトウェア毎にひとつのコマンドに結びついている。

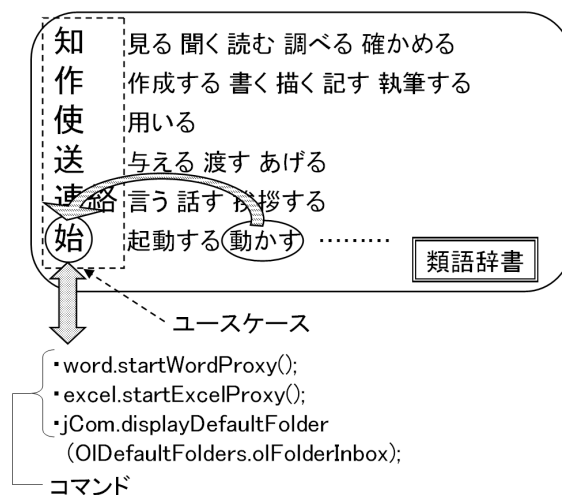


図 2: 類語辞書とユースケース

2.1 入力文解析

奈良先端科学技術大学院大学で開発された日本語係り受け解析器 CaboCha/南瓜 [1] を使って、システムは、「メールを連絡先: 大西可奈子, お茶の水女子大学人間文化研究科数理・情報科学専攻, 東京都文京区大塚 2-1-1, 03-5978-5709, 03-5978-5709, onishi@koba.is.ocha.ac.jp

2.3 操作対象の候補決定

先行研究 [2] においては、ユースケースを表現する特定の動詞を含まないユーザからの様々な発話によっても、アプリケーションソフトウェアを操作可能にするため、述部に対する類語辞書を導入し、ユーザの発話をユースケースに結び付ける特定の述部に柔軟に結び付ける工夫を行った。

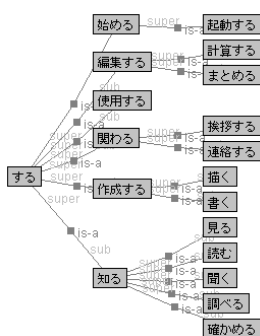


図 3: ユーザの発話内容オントロジー

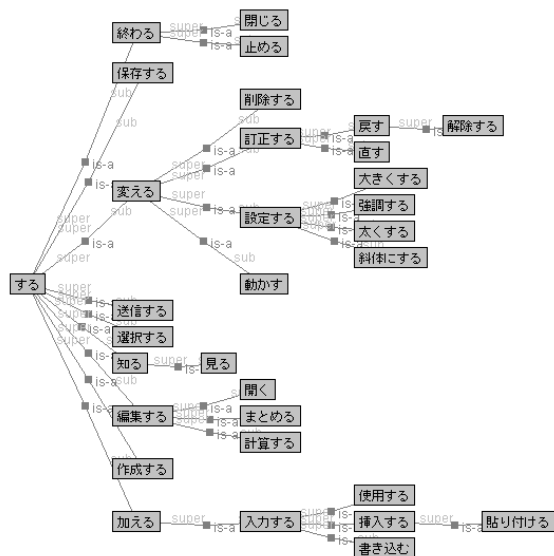


図 4: 操作対象のユースケースオントロジー

本研究では、複数のアプリケーションソフトウェアを同時に操作することを目的とするため、ユースケースオントロジーをシステムに持たせる（図 4 参照）。オントロジーの抽出には、溝口らによって開発された AFM (Activity First Method) [3] を考慮して行い、オントロジーの作成には法造 [4] を用いた。このユースケースオントロジーには、各ユースケースがどの程度、あるアプリケーションソフトウェアに関係している状態であるかを示すユースケース関連辞書が備わっている（図 5 参照）。

また、ユースケース関連辞書は状況に応じて二つ用意されている。ひとつはアプリケーションソフトウェアが起動状態ではない、すなわち行為を及ぼす対象がない場合に使用されるユーザの発話内容オントロジーを使用したユースケース関連辞書で、もうひとつは、すでになんらかのアプリケーションソフトウェアが起動状態である場合に使用されるユースケースオントロジーを使用したユースケース関連辞書である。これは、ユーザの命令文から抽出した述部の意味合いが状況に応じて変わるため導入した。例えば、以下に示す二つの文章、

- ワードを動かして。
- さっき挿入した画像を動かして。

は、同じ述部「動かす」であるが、一方がアプリケーションソフトウェアの起動を期待するものである反面、他方はすでに起動したアプリケーションソフトウェアを使っての操作を期待するものである。このような、状況による意味合いの違いを二

つのユースケース関連辞書が解決している。これにより、選択されたユースケースとあるアプリケーションソフトウェアがどの程度関連しているのかの状況に応じた度合いがわかり、操作対象となるアプリケーションソフトウェアをある程度絞り込むことができる。

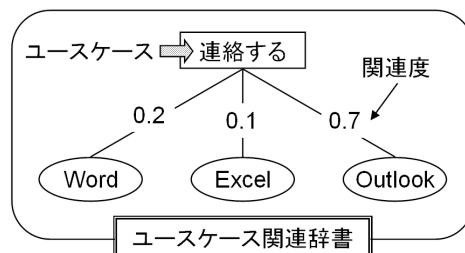


図 5: ユースケース関連辞書

2.4 操作対象および内容の決定

動詞によって挙げられた複数のユースケースの候補をひとつに絞り込むため、本研究では更に名詞関連辞書を導入した（図 6 参照）。これは、ある名詞が各アプリケーションソフトウェアにどの程度関連しているかを数値化した値を保持した辞書で、この数値が高いほどその名詞は各アプリケーションソフトウェアとの関係が強いと判別する。名詞関連辞書を使って、候補であるユースケースに名詞関連度を付与し、この値が最大となるものを最終的に操作対象となるユースケースとする。これにより複数のアプリケーションソフトウェアの動作の競合は防止され、操作の代替も可能となる。

例えば「メールを書きたい」という命令を受け取った場合、システムは「書く」と「メール」に最も関連するメールエディタ、すなわち Outlook を起動しようとするが、Outlook はなんらかの理由で起動できない状態とする。そこでシステムは代替するアプリケーションソフトウェアを探すため、メールの代替となるものを推論し、「図表入りテキスト」を扱えるエディタを代わりに探す。そして上位概念であるエディタからサブクラスを探し、「図表入りテキスト」を扱えるテキストエディタを発見する（図 7 参照）。それによりテキストエディタとしての機能を持つものとして、3つのアプリケーションソフトウェアが存在することがわかる。ここで Outlook は使用不可であり、Word と Excel のどちらを起動するかが問題となるが、同様に 2 つにもユースケース関連辞書と名詞関連辞書を基に関連度を付与してやると関連度の大小関係より Outlook の代替として使用するものは Word に決定される（図 8 参照）。

2.5 関連度の数値化

本システムにおいては、語彙ごとの各アプリケーションソフトウェアに対する関連度合いを考慮し、関連度を筆者自らが付与した。その数値を利用することにより、ユーザの命令とアプリケーションソフトウェアを結びつけ、複数のアプリケーションソフトウェアを操作することができるシステムとなっている。

また本システムにおいては、判別されたユースケースを実際にアプリケーションソフトウェアのコマンドに結び付けるために JCom から Com を利用している。JCom とは、Java から Windows の COM (Component Object Model) を利用できるミドルウェア [5] で、Word, Excel など既存の COM の機能を Java から利用できる。また COM とは、Microsoft 社が提唱する部品化されたプログラムを作成・利用するための基盤となる技術仕様である。

3. 処理概要

下記のシナリオの流れを例にして、システムの動きを具体的に説明する。

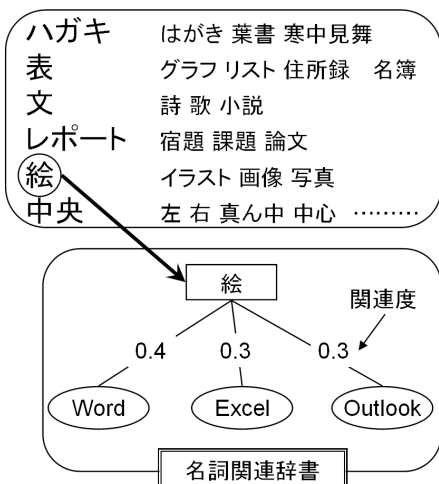


図 6: 名詞関連辞書

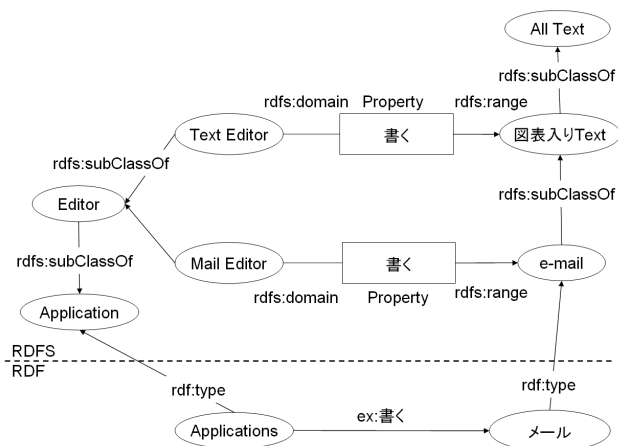


図 7: ドメインオントロジー 1

	ユーザの命令	操作対象	期待する操作
1	メールを書きたい	Outlook	起動
2	写真を入れて招待状を作りたい	Word	起動
3	「お知らせ」と書いて	Word	文字を書き込む
4	件名「こんにちは」で送りたい	Outlook	件名を入れて新規メール作成
5	真ん中に動かして	Word	中央寄せ
6	エクセルを動かして	Excel	起動

3.1 シナリオ文 1 : 「メールを書きたい」

アプリケーションソフトウェアが立ち上がっていない状態で、ユーザから「メールを書きたい」と命令された場合を例に、本システムの処理概要を説明する。

1. 状況判断と解析

最初に、どのアプリケーションソフトウェアが起動して、どのような状態であるかを把握する。現在は行為を及ぼす対象が存在していないので、最適と思われるアプリケーションソフトウェアを判断し、起動実行すればよい。形態素解析により、述部として「書く」を、また名詞として「メール」を抽出する。ユーザの命令と各アプリケーションソフトウェアとの関連度は X_{word} , X_{excel} , $X_{outlook}$ という変数で監視する。 X_{word} は Word に、 X_{excel} は Excel に、 $X_{outlook}$ は Outlook にそれぞれ対応しており、変数

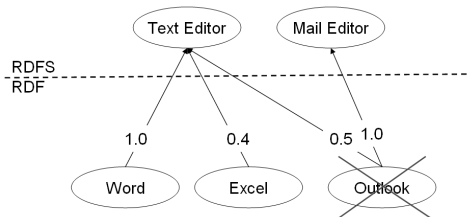


図 8: ドメインオントロジー 2

が保持する数値の高い方が、ユーザの命令と変数が対応するアプリケーションソフトウェアとの関連は強いと判別する。

2. 述部の処理



図 9: ユーザの発話内容オントロジー

ユースケース関連辞書を使って、述部の関連度を各変数に加える。述部「書く」は関連度を保持していないので、関連度を保持している別の述部を探しに行く。現在は行為を及ぼす対象が存在していないので、ユーザの発話内容オントロジーを参照し、「書く」は「作成する」のサブカテゴリであると推論し、「作成する」の関連度を監視変数にそれぞれ加える。

- $X_{word}=1.0$
- $X_{excel}=1.0$
- $X_{outlook}=1.0$

「作成する」はどのアプリケーションソフトウェアにも等しく関連するので、監視変数はどれも 1.0 となる。

3. 名詞の処理

名詞の関連度を各監視変数に加える。関連度は以下のようにになる。

名詞の関連度	Word	Excel	Outlook
メール	0.2	0.0	1.0

これより、名詞「メール」は Outlook に強く関連し、Word にも多少関連しているということがわかる。

これらを各監視変数の述部の関連度の上に加えた結果、

- $X_{word}=1.0+0.2=1.2$
- $X_{excel}=1.0+0.0=1.0$
- $X_{outlook}=1.0+1.0=2.0$

となり、関連度の大小関係より、システムは Outlook の「作成する」ユースケースから Outlook を起動する。

3.2 シナリオ文 2 : 「写真を入れて招待状を作りたい」
続けてユーザから「写真を入れて招待状を作りたい」と命令を受け取った場合を考える。

1. 状況判断と解析

「メールを書きたい」という命令に対して実行を行った後なので、Outlook が起動した状態であることから、システムは Outlook で作業をするか、あるいは他のアプリケーションソフトウェアを起動すればよいのだと考え、同様に

3つの変数 X_{word} , X_{excel} , $X_{outlook}$ で関連度を監視する。 X_{word} は Word で作業をするということと、ユーザの命令「写真を入れて招待状を作りたい」がどの程度関連しているかに対応しており、 X_{excel} と $X_{outlook}$ は、それぞれに対応するアプリケーションを起動するという作業とユーザの命令がどの程度関連しているかに対応している。

2. 述部の処理

前回同様、述部「作る」の関連度を各監視変数に加える。したがって関連度はそれぞれ、

- $X_{word}=1.0$
- $X_{excel}=1.0$
- $X_{outlook}=1.0$

となる。

3. 名詞の処理

次に名詞の関連度を加える。名詞の関連度は以下のようになっているので、

名詞の関連度	Word	Excel	Outlook
写真	0.4	0.3	0.3
招待状	0.6	0.0	0.4

- $X_{word}=1.0+0.4+0.6=2.0$
- $X_{excel}=1.0+0.3+0.0=1.3$
- $X_{outlook}=1.0+0.3+0.4=1.7$

となる。

4. 前回起動した操作対象に対応する変数への加算

初期状態がアプリケーションソフトウェア起動状態でない場合は以上の作業で終わりであるが、前回ユーザからの命令を受け取り、かつ、それに対応するアプリケーションソフトウェアのコマンドが実行された場合は、前回起動した、あるいは使用したアプリケーションソフトウェアに対応する変数には 0.1 が加算されることとした。

これは直前に使用したアプリケーションソフトウェアと、その他の候補ユースケースであるアプリケーションソフトウェアとに差異をつけるためである。今回は、前回起動したアプリケーションソフトウェアが Outlook であることから、それに対応する変数 $X_{outlook}$ には 0.1 が加算される。

よって関連度は最終的に、

- $X_{word}=1.0+0.4+0.6=2.0$
- $X_{excel}=1.0+0.3+0.0=1.3$
- $X_{outlook}=1.0+0.3+0.4+0.1=1.8$

となる。

以上より、この大小関係からシステムは Word で作業するのが最適であると判断し、Word のコマンドを実行する。

3.3 シナリオ文 3 以降

以降のシナリオも同様の計算によって関連度を計算する。

- 「お知らせ」と書いて

述部「書く」から「作成する」の関連度を加えて、 $X_{word}=X_{excel}=X_{outlook}=1.0$
 前回起動したのが Word であることから、 $X_{word}=1.0+0.1=1.1$
 したがってシステムは、対応する関連度が最大となった Word で作業する。

- 件名「こんにちは」で送りたい

述部「送る」と名詞「件名」の関連度を加えて、 $X_{word}=0.1+0.1=0.2$

$$X_{excel}=0.1+0.1=0.2$$

$$X_{outlook}=0.8+0.8=1.6$$

前回起動したのが Word であることから、

$$X_{word}=0.1+0.1+0.1=0.3$$

したがってシステムは、対応する関連度が最大となった Outlook で作業する。

- 真ん中に動かして

述部「動かす」と名詞「真ん中」の関連度を加えて、 $X_{word}=1.0+0.5=1.5$
 $X_{excel}=1.0+0.4=1.4$
 $X_{outlook}=1.0+0.1=1.1$
 前回起動したのが Outlook であることから、 $X_{outlook}=1.0+0.1+0.1=1.2$
 したがってシステムは、対応する関連度が最大となった Excel で作業する。

- エクセルを動かして

述部「動かす」と名詞「エクセル」の関連度を加えて、 $X_{word}=1.0+0.0=1.0$
 $X_{excel}=1.0+1.0=2.0$
 $X_{outlook}=1.0+0.0=1.0$
 前回起動したのが Word であることから、 $X_{word}=1.0+0.0+0.1=1.1$
 したがってシステムは、対応する関連度が最大となった Word で作業する。

4. まとめと今後の課題

本研究が提案する手法を利用することにより、ユーザの命令に対して柔軟な対応が可能になり、ひとつのインタフェースで複数のアプリケーションソフトウェアが動かせるようになった。

今後の課題として、ユーザとシステム間のコミュニケーションを対話形式にし文脈を理解することにより、一語発話 [7] のような命令についてもユーザの希望する動作を推論できるようなシステムにすることを考えている。それにより、現在は前回使用したアプリケーションソフトウェアについて 0.1 を加算するという単純な機構が、より複雑な文脈にまで対応できるシステムになり、より高度な文脈に柔軟に対応できるようになる。また、対象ドメインを拡張し、ドメインオンロジーの拡充とともに、状況判断機構を持ち、より複数のアプリケーションとの連携を図ることができる柔軟な操作インタフェースとして開発を進めていくつもりである。

参考文献

- [1] 奈良先端科学技術大学大松本研究室：CaboCha/南瓜
<http://chasen.org/taku/software/cabocha/>
- [2] 言葉によるアプリケーションソフトウェアの操作, 菊池友里子, お茶の水女子大学理学部情報科学科卒業論文, 3月, 2005.
- [3] タスク・ドメインロールに基づくオントロジー構築ガイドシステムの設計と開発 - 石油精製プラントを例として -, 石川誠一, 久保成毅, 古崎晃司, 來村徳信, 溝口理一郎, 人工知能学会論文誌, Vol17, No.5, pp585-597, 2002.
- [4] オントロジーエディタ法造, <http://www.hozo.jp/>
- [5] JCom(Java-COM Bridge)
<http://sourceforge.net/projects/jcom>
- [6] 知の科学「オントロジー工学」, 溝口理一郎, 人工知能学会編集, 2005.
- [7] Hitoshi IIDA: One-word Utterance Pragmatics and Emotional Speech Intention, Language Understanding and Action Control, Annual Project Report Grant-in-Aid for Creative Basic Research 13NP0301, March, 2006, Tokyo Institute of Technology, Tanaka, Hozumi (105-114)