

自然言語を利用したアプリケーションサービスの統合化

Natural Language-based Application Service Integration

岩爪道昭^{*1} 小林 一郎^{*2} 高橋 裕介^{*1} 杉本 徹^{*1}
 Michiaki Iwazume Ichiro Kobayashi Yusuke Takahashi Toru Sugimoto

岩下 志乃^{*1, *3} 伊藤 紀子^{*1, *4} 菅野 道夫^{*1, *4}
 Shino Iwashita Noriko Itoh 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} 御茶ノ水女子大学 理学部
 Ochanomizu University, Faculty of Science

Everyday language computing (ELC) is a new computational paradigm that all people, from small children to aged persons, can access and use computing systems with his/her own everyday language. As a way to realize ELC, we proposed a framework of language-based operating system (LOS), and we are now working intensively to develop the fundamental part of it. In this paper, we report our status of research on LOS. In LOS, all kinds of information processing are done through everyday language. Language-based application is a kind of agent software that provides services through language. In this paper, we discuss the application service integration in LOS from the experience of implementing prototype systems.

1. はじめに

日常言語コンピューティングは、人が普段使っている言葉を媒介してコンピュータの知的化を目指す新しいパラダイムである。日常言語コンピューティングでは、既存のアプリケーションは、自然言語で操作・管理・実行するための LAPI (言語プログラミングインタフェース)による「言語化」を通じてサービスを提供する。本研究では、これまでのプロトタイプINGの成果を踏まえ、日常言語コンピューティング環境におけるアプリケーションの統合的利用法について考察し、今後の展望について論じる。

2. 日常言語コンピューティング

日常言語コンピューティングシステムの概要を図 1に示す。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会話する。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(詳細は、[岩爪 02, 岩爪 03a]を参照)。

2.1 言語アプリケーション

日常言語コンピューティング環境では、エンドユーザへのサービスは、クライアント秘書を仲介役として、言語アプリケーションによって提供される。

LOS における全てのサービスは、クライアント秘書を介して、LOS 上に存在する言語アプリケーションによって提供される。言語アプリケーションは、スクラッチから設計・開発することも可能であるが、既存のソフトウェア資源やアプリケーションサービスを

連絡先 岩爪道昭, 理化学研究所脳科学総合研究センター,
 埼玉県和光市広沢 2-1, TEL: 048-462-1111 ext. 7192,
 FAX: 0480467-9493, lwazume@brain.riken.jp

^{*3} 現在, 九州大学九州大学 大学院芸術工学研究院 ^{*4} 現在,
 同志社大学文化情報学部

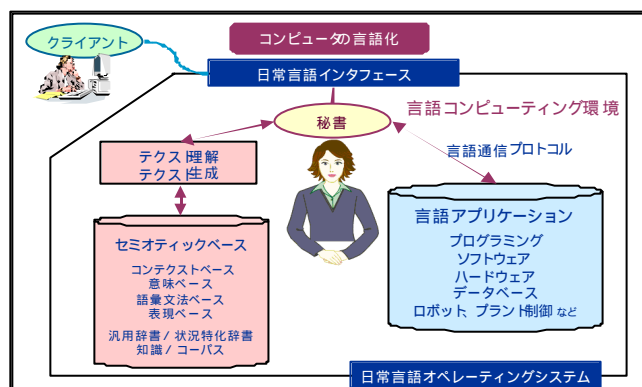


図 1 日常言語コンピューティングの概要

有効に利用することも不可欠である。そこで、我々は、既存のアプリケーションを言語化するための方法論について取り組んできた。ここで、言語化とは、コンピュータ内部の全ての資源 (ソフトウェア、ハードウェア) に対して、言語 (の意味) に基づくアノテーションを施し、そのアノテーション情報に基づいて、コンピュータ環境のサービスを実行・管理すること。

2.2 レガシーアプリケーションの言語化

我々は、これまでに、レガシーアプリケーションの言語化を実現するための方式として、言語プログラミングインタフェース (以下、LAPI: Language Application Programming Interface) と呼ばれる API を提案し、その構成法および自然言語媒介したソフトウェアの制御方式について検討を進めてきた [岩爪 -3b]。特に、LAPI 内部におけるユーザからの要求 (入力文) とソフトウェアコマンドの自然言語表現の間の意味マッチング (アクティブマッチング) の方式について取り組んできた [岩爪 04]。

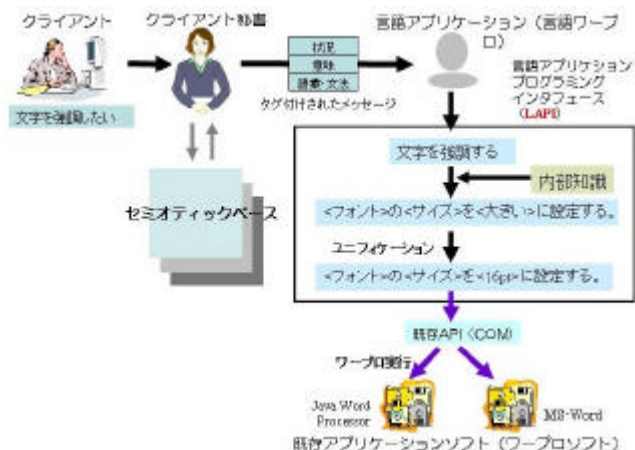


図 2 言語アプリケーション概要

2.3 日常言語アプリケーション

また、具体的なドメイン、アプリケーションを設定し実証システムの開発を進めてきた[杉本 03, 岩爪 03a, 伊藤 04].

言語ワープロシステムは、既存のワードプロセッサを LAPI によって言語化した日常言語アプリケーションの一種である[杉本 03, 岩爪 03b]. 言語ワープロシステムに関する研究の狙いは、ワードプロセッサにとどまらず、レガシーソフトウェア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。

スマートヘルプシステムは、ユーザとクライアント秘書エージェントが対話しながら、ユーザの言語使用や知識レベルに応じて、既存のコンピュータ操作に関するヘルプデータを個人化して出力する、質問応答システムである[岩下 04].

言語プログラミングシステムは、複雑な操作手順や実行条件を含む複数の文からなるユーザからの要求仕様を、一貫した入力テキスト(日常言語プログラム)として処理して、応用ソフトウェアの操作を実行することができるシステムである[杉本 04].

3. インタフェースの統合化(共通化)

LOS としてユーザにシームレスなサービスを提供するためには、第一段階として、共通のユーザインタフェースおよびアプリケーションインタフェースを介して、各アプリケーションを呼び出し、実行処理結果を返す必要がある(図 2 参照)

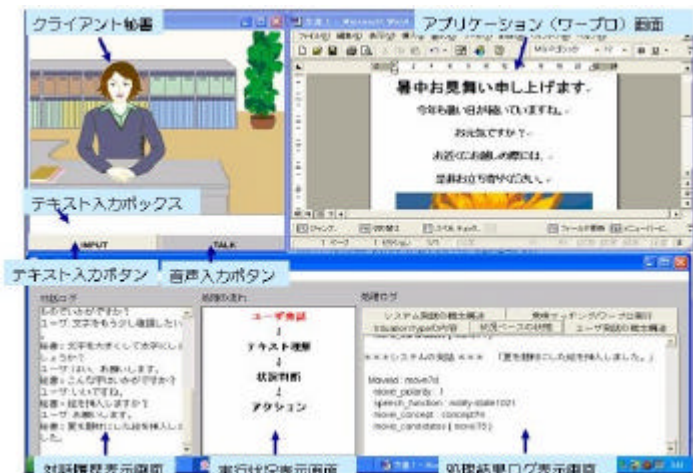


図 3: 言語アプリケーション(ワープロ)プロトタイプ

2003 年度までのプロトタイプでは、ユーザはテキスト入力によってクライアント秘書に指示を与えていたが、2004 年度に試作・拡張した言語ワープロシステム、言語プログラミングシステム、スマートヘルプシステムは、音声入力によって(図 12 参照)、共通のインタフェースから呼び出し、サービスを利用することが可能である(図 3 参照)

音声認識には、Julius[Julius]を使用し、言語モデルは、過去の3年間の対話実験[岩爪 03]によって収集された理研模擬対話音声コーパスを利用して構築されている。

共通の音声言語インタフェースを構築することで、これまでアプリケーションごとに提供されていたユーザインタフェースではなく、LOSのイメージをより具体化することができた

4. アプリケーションの統合化

しかし、表層のインタフェースの共通化だけでは、LOS の特徴の一つの個人化が実現されていると言えない。インタフェースだけでなく、アプリケーションの機能やサービスそのものもユーザの要求や利用形態に応じてカスタマイズし、複数のアプリケーションのサービスを統合的に利用可能にすることが我々の目標である。

アプリケーションの統合に、大別すると次の形態が考えられる。

- (a) 単一アプリケーションの統合化
- (b) 異種アプリケーションの統合化

以下では、これらの形態について考察する。

4.1 単一アプリケーションの統合化

ノービスユーザにとって、アプリケーションで提供されるサービスは過多である場合が多く、膨大な機能の中からユーザの問題解決に必要な機能だけを選別するのは困難である。したがって、ユーザに習熟レベルや利用目的に応じてサービスや機能がカスタマイズされることが望ましい。

例えば、最も簡単な例としては、

文字を強調したい

とい要求に対して、

- フォントのサイズを大きくする**
- フォントをボールド体にする**
- フォントの色を変更する**
- フォントの種類を変更する**

のような各機能の組み合わせが考えられる。今、状況 S_i における新しい機能を $new_action(S_i)$ 、利用可能な個目の既存機能を $action_j(S_i)$ とすると、

$$new_action(S_i) = (action_1(S_i), \dots, action_j(S_i))$$

とい系列で形式的に表現される。これは古典的な AI のプランニングにおけるプラン構造と見ることができ、ここでの狙いは、状況 S_i における新機能に関するユーザの言語使用(日常言語表現)とアプリケーション固有の各機能間の中でマッピングをとることにある。すなわち、ユーザの使用する日常言語に特化したアプリケーションのカスタマイズを行うことである。

例えば、ソフトウェアに予め典型的な状況を想定した機能がプランライブラリとして用意されている場合は、その機能に対して自然言語表現を割り当てておく。これを LAPI 標準形と呼ぶ[岩

爪 03b]. ユーザからの要求文とLAPI標準形の間で意味のマッピングを行い, 一定の閾値を越えた当該プランの系列を選択する(意味マッピングの詳細は[岩爪 03b, 岩爪 04]参照).

もし, ユーザの要求文に該当する標準形がない場合は, 対話的にプランニングを行い, 最終的にユーザによって受理されたプランの系列に対して, 新規に言語表現を割り当てる.

このようなアプローチは一種の事例ベースと見なすこともできるが, ソフトウェアの機能を形式化された内部表現として取り扱うのではなく, 言語表現を割り当て明示化することでユーザが自分のことばでソフトウェアの機能を利用することが可能になる点が重要である.

4.2 異種アプリケーションのサービス統合

異種アプリケーションの連携・統合に関する技術は, 主にビジネスアプリケーション分野で特に盛んであり, EAI (Enterprise Application Integration) と呼ばれている. EAI は, 企業内で業務に使用される複数のコンピュータシステムを有機的に連携させ, データやプロセスの効率的な統合をはかること, また, それを支援する一連の技術やソフトウェアの総称のことである.

また近年, Tim Berners-Lee らが提唱するセマンティック Web の構想では[TBL 02], 人間に代わりインターネット上の多種多様な情報資源から必要な情報収集し, 複数のサービスを統合・連携を実現するセマンティック Web エージェント可能性が示唆されており, DAML-S 等サービス連携のための仕様も取り決められている[Ankolokar 01]. 又, これらの仕様に基づく, より具体的な Web サービス連携に関する研究[杉本 03b]や Web サービスの連携の技術を実世界に結びつけるための研究[幸島 03]など行われてきている.

しかし, これまでのセマンティック Web におけるサービス連携の対象ドメインは, インターネット上の定型的ビジネスプロセスの合成を目的としたものが多く, 実世界におけるノービスユーザの視点にたったサービス連携に殆どなれていない. また, ソフトウェアだけでなくヒューマンリソース自体もシステム内に取り込んでシームレスにサービスを提供するためには, ビジネスロジックを取り扱うだけでは不十分である.

小林らは, 日常言語コンピューティング環境における通信規約として自然言語を用いたプロトコル通信を提案している[小林 02]. EAI の大手ベンダー Gartner 社 VPO の Schulte が,

「アプリケーション統合を実現するために重要なことは, 冗長性・不一致性を排除するのではなく, 受け入れた上で対処方法を検討することが必要になる。」

と指摘しているように[Gartner], 現実のレガシーアプリケーションを連携する場合にも, 日常言語を媒介とする通信が有効に機能するケースが多々考えられる. 例えば, 「滞在先」, 「宿」, 「ホテル」, 「旅館」といった異なる属性情報によってされるデータの相互参照から, 「隠れ家っぽいお店を探して欲しい」というユーザの要求と「個室のあり」といふ店の属性情報との意味的マッチングをとることで, 単なるレガシーシステムのブリッジ的役割だけでなく, より知的な情報提供などが可能になると考えられる. 言語プロトコル通信に関する詳細な考察については[小林 05]を参照されたい.

5. まとめ

日常言語コンピューティングは, 人が普段使っている言葉を媒介してコンピュータ的化を目指す新しいパラダイムである. 本稿では, これまでのプロトタイピングの成果を踏まえ, 日常言語コンピューティング環境におけるアプリケーションの統合的利

用法に議論した. 現在, 単一アプリケーションの統合化からプロトタイピングを進めている.

参考文献

- [伊藤 04] 伊藤紀子, 杉本徹, 岩下志乃, 岩爪道昭, 高橋祐介, 小林一郎, 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 2 報), 第 18 人工知能学会全国大会, 2C1-05, 金沢, 2004.
- [岩下 03] 岩下志乃, 小林一郎, 伊藤紀子, 杉本徹, 菅野道夫: 日常言語によるヘルプ出力の個人化について, 第 17 回人工知能学会全国大会, 新潟, 2003.
- [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 人工知能学会全国大会, 3E2-04, 金沢, 2004.
- [岩爪 02] 岩爪道昭, 小林一郎, 伊藤紀子, 高橋祐介, 菅野道夫: 日常言語コンピューティング - 言語現象の社会的, 機能的解明と日常言語オペレーティングシステムの実現に向けて -, 人工知能学会論文誌近未来チャレンジ特集, Vol.17, No.2, pp.181-185, 2002.
- [岩爪 03a] 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.
- [岩爪 03b] 岩爪道昭, 小林一郎, 高橋祐介, 菅野道夫: 自然言語を媒介としたソフトウェア制御のための意味マッチング方式の検討, 第 17 回人工知能学会全国大会, 新潟, 2003.
- [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 人工知能学会全国大会, 3E2-02, 金沢, 2004.
- [小林 02] 小林一郎, 岩爪道昭, 杉本徹, 岩下志乃, 小澤順, 菅野道夫: 自然言語をコンピュータの通信プロトコルにする研究, ファジィ学会誌, Vol.14, No.5, pp.491-502, 2002.
- [小林 05] 小林一郎, 菅野道夫: 言語プロトコル通信における基本的枠組みの再考, ファジィ学会誌, 第 19 回人工知能学会全国大会, 1B1-02, 北九州, 2005.
- [幸島 03] 幸島明男, 和泉憲明, 車谷浩一, 中島秀之: "セマンティック Web エージェントによる実世界指向のサービス連携機構", 第 2 回ユビキタスコンピューティングシステム研究会報告, pp.123-128, 情報処理学会, 2003.
- [杉本 03a] 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.
- [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 人工知能学会全国大会, 3E2-03, 金沢, 2004.
- [杉本 03b] 杉本悠樹, 村上陽平, 荒井幸代, 石田亨: "セマンティック導入によるユーザ指向 Web サービス連携支援", 第 17 回人工知能学会全国大会, 新潟, 1D4-01, 2003.
- [Ankolokar 01] A. Ankolokar et al., "DAML-S: Semantic Markup for Web Services", In Int. Semantic Web Working Symposium, pp.411-430, 2001.
- [Gartner] <http://www.gartner.com/>
- [TBL 02] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", Scientific America, 284(5), pp.34-43, 2001.
- [Julius] <http://julius.sourceforge.jp/>