10 サーバ連携に基づく継続的な音声認識応用システム開発

小林 哲則　　中野 鐵兵
早稲田大学 理工学術院 情報理工学科

従来音声認識応用システムの開発は，エンジン開発者から提供を受けた音声認識エンジンをアプリ開発者がシステムに組み込みユーザに渡すという，一方向型開発パラダイムに沿って行われてきた．そこでは，ユーザの手元にシステムが渡った段階で，システムの性能の成長は止まってしまう．本稿では，エンジン開発者とアプリ開発者さらにはユーザも含めた密なる連携に基づいて開発を行う双方向型開発パラダイムを提唱し，サーバ連携によってそれを支える仕組みを実現した．半自動的，継続的に音声応用システムの性能を改善する仕組みを，アプリケーションとエンジンの双方に非依存な形で実現できることとなり，多くの利便性の高いシステムが開発可能になることが期待される．

音声認識応用システム開発の諸問題

本稿では，2006 年度からの 3 年間，経済産業省のプロジェクト「音声認識基盤技術の開発」[1] のもとで開発された，サーバ連携に基づいて良質の音声認識応用システム開発を可能にする新たな音声認識アーキテクチャについて述べる．

プロジェクト開発当時から，音声認識エンジンの性能は非常に高いものがあった．NHK のニュース字幕変換器はすでにその数年前からほぼ完璧な書き起こしを実現していたし，京大の国会議事録作成器のデモを見ても，誤り個所を見つけるのに苦労するほどであった．しかし，一方で，我々が音声認識応用システムのユーザとしてその恩恵にあずかれる場面はいまだに少ない．

良質な音声認識応用システムができるためには，音声認識エンジンを使いこなして良いアプリケーションにつなげるためのノウハウなり技術が必要である．音声認識エンジンの特性，および音声利用時のユーザの特性に関する深い知識を持った上で，音声認識エンジンをカスタマイズし，応用システムを設計することが望まれている．音声技術をもって，有用なアプリケーションをどのように実現するかについての解を与える役割である．ここでは，この音声認識エンジンとアプリケーションとをつなぐ役割を音声ソリューションと呼ぶことにする（図 -1 参照）．実際，音声認識のビジネスをうまく軌道に乗せているいくつかの企業は，この音声ソリューションに多くの人材とコストをかけている．しかし，その効果をより顕著なものとするためには，今一歩スケールメリットを出せるための組織なり技術なりが必要とされている．

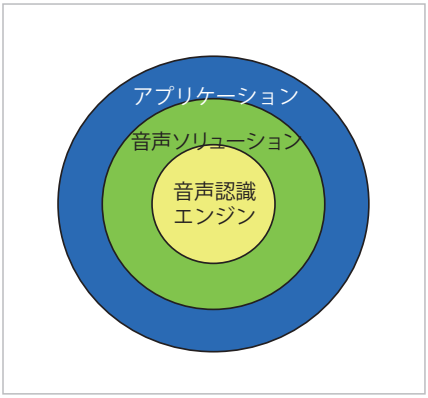

図 -1　音声認識応用システムを作る技術階層

では，この役割はどのように果たされるべきであろうか．それをエンジン開発側に求めることは，日本のエンジンメーカが置かれた状況からすると，現実的ではない．膨大な数のアプリケーションに万能な音声認識エンジンを開発することは困難であって，どこかに特化して強みを発揮するエンジンを開発している状況を考えると，エンジンメーカが音声ソリューションをビジネスの軸として展開することが難しいことは容易に想像できる．日本のエンジンメー

カが一般に大企業であって，細かなビジネスを相手にできないことも，音声ソリューションを手掛けることを難しくしているようにも思われる．

また，アプリ開発者がこの役割を担うために，音声認識エンジンと音声利用時のユーザファクタの専門家であらねばならないというならば，これも筋の良い話ではない．音声の専門家であることを求められるなら，音声アプリケーションの開発者は増えそうもない．そうなると，音声アプリケーションは決して広がることはない．良いアプリケーションが増えるためには音声アプリケーションの作り手が増えることが絶対的な条件なのである(図 -2 参照)．

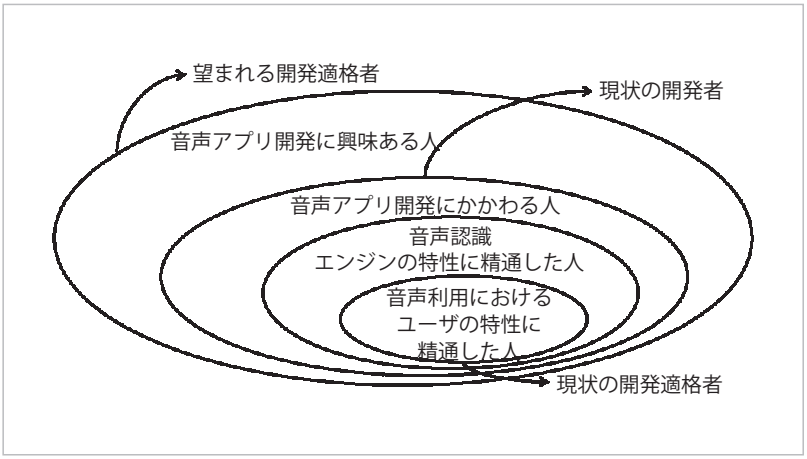

図 -2　音声認識応用システムの開発者の階層

プロジェクトでは，現状日本の音声開発が抱える音声ソリューションにかかわる問題を，技術的支援によって解決することを試みた．音声の専門家でもなく音声利用時のヒューマンファクタにも疎い人が，質の高い音声アプリケーションを容易に開発できるようにするためには，どのような仕組みが必要かを考え，それを支える技術を開発した．本稿では，現状抱える問題を若干詳細に述べた後，提案した Proxy-Agent を核としたサーバ連携の形と，それを用いた音声応用システムを育てる仕組みについて紹介する．

一方向型開発から双方向型開発へ [2)]

現在の段階で，音声認識応用システムは，音声認識エンジンの性能を十分に引き出しているとは言い難い．音声ソリューションに失敗しているのである．このような現状には，日本における音声認識応用システムの開発体制が少なからず影響しているようだ．日本において音声認識応用システムの開発は，極端な分業体制の中で行われることが多い．エンジン技術者が開発したエンジンをアプリ開発者が受けとってこれにアプリケーションをかぶせて音声認識応用システムを作る．情報の流れは，システムの流れと同様に，エンジン開発者からアプリ開発者，ユーザへとおおむね一方通行であって，逆の流れは稀である [3)] (図 -3 参照)．ここで驚くべきことであるが，アプリ開発者は，エンジンについての詳しい特性も，音声認識応用システムを利用したときのユーザの振舞いに関する知見も持っていないことが圧倒的に多いという．

性能が上がったとは言え，音声認識エンジンはどのように使っても 100% の性能を発揮できるほどの完成度を持った部品にはなりきっていない．アプリ開発者は，エンジンの性能を引き出すための術を知る必要があるのだが，それを知る構造がないようだ．エンジンメーカ側も，アプリ開発に必要な性能情報を提示する必要があるのだが，それが十分でない．このため，どういう条件で使うとどういうことになるのか，どの程度の性能になるのかがアプリ開発側に見えていない．結果，適切なエンジンの使い方ができない，良いアプリケーションができない，音声認識が広まらない，ユーザが慣れない，性能が上がらない（性能が出るためにはある程度の慣れが必要である），とネガティブスパイラルができあがる．

また，アプリ開発者は，真の意味で問題を掴んでいないという問題もある．現状で，売ったシステムの動作解析はできない．現場で何が起こっているか分からない．結果としてユーザの声が開発にフィードバックされない，よって使いやすくならない．こ

こでもネガティブなスパイラルが形成されてしまう．そもそも問題がなにかは，机上の検討では困難なのである．

そこで，プロジェクトにおいては，ユーザ・アプリ開発者・エンジン開発者間で互いに情報を共有しながら双方向の密なる連携を実現し，継続的に育てることができる音声認識応用システムの開発環境を実現することを目指した．ここで提案する，双方向型音声認識アプリケーション開発パラダイム（図 -4）では，ランタイム（アプリケーション利用時）のユーザの振舞いに関するデータを収集し，これを開発サイドに対してフィードバックする．また，フィードバックされたランタイムデータの解析に基づいて部品を改良した後，これを随時再配信できるようにすることで，ユーザは常に最適な状態で音声認識システムを利用できるようになる．また，開発者間では，開発にかかわるさまざまな知見を共有できるようにする．このことによって，音声アプリケーションが日々「育つ」仕組みができあがるとともに，アプリ開発者がエンジンやヒューマンファクタに関する深い知識がなくとも，一定水準の音声認識応用システムを開発することが可能となる．

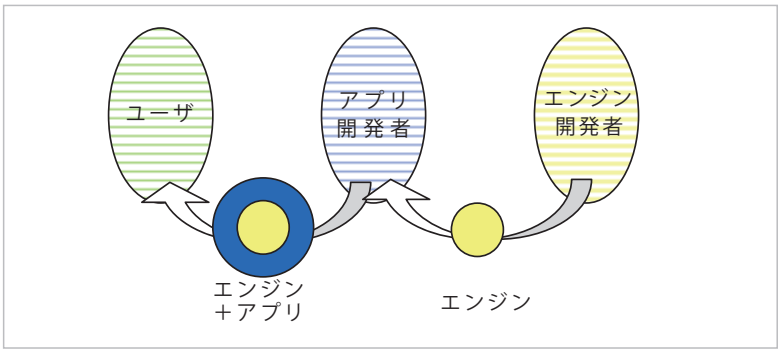

図 -3　一方向型開発パラダイム

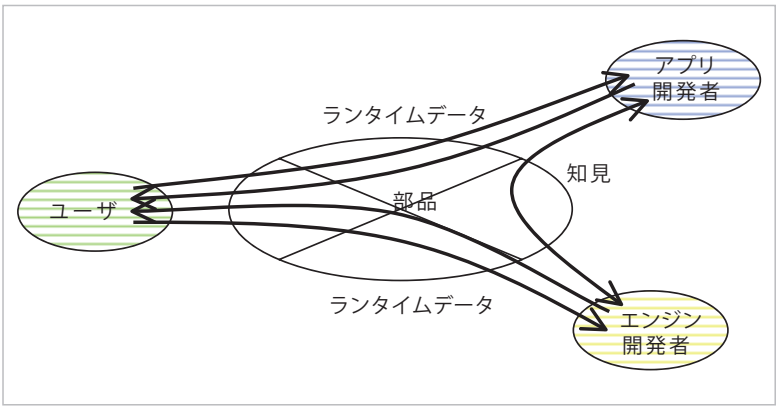

図 -4　双方向型開発パラダイム：ユーザから開発者へはランタイムデータのフィードバック機能を，開発者からユーザへは部品の再配布機能を，開発者間には知見の相互共有機能を与える．

Proxy-Agent アーキテクチャ

提案する双方向型開発の中核を担うのは Proxy-Agent と呼ぶ新たな音声認識応用システムの構成要素である．Proxy-Agent は，音声認識エンジンの開発者およびアプリ開発者の負担を抑えた上で，さまざまなサーバ連携を可能にする [4]．このサーバ連携機能をエンジンおよびアプリケーションの双方に依存することなく行うことによって，情報共有の可能性を広げ，スケールメリットを実現することを目指している．

Proxy-Agent アーキテクチャは，Proxy-Agent，Application，Engine-Adapter，Device-Adapter の 4 つの要素，およびサーバサービスから構成される（図 -5）．Proxy-Agent とは，アプリケーションと音声認識エンジンの間に入ってその連携を担当するソフトウェアであり，アプリケーションから音声認識エンジンに対する制御信号と音声認識エンジンの入出力に関する情報の収集を行う．Engine-Adapter とは，1 つ以上のプラグイン群から構成される仮想音声認識エンジンオブジェクトを表し，音声認識機能の実装が含まれる．認識対象となる入力データは Device-Adapter から取得する．Device-Adapter とは実際の入力デバイスからのデータ取得ロジックを包含したデータ提供オブジェクトであり，Proxy-Agent はデバイスからエンジンへのデータの流れを中継することで，実際に対象となるデータを収集する．Engine-Adapter も Device-Adapter も Proxy-Agent に対するプラグインとして用意される．アプ

リケーションはProxy-Agentとメッセージの送受信を行い，Engine-Adapterの機能を呼び出す．Proxy-Agentアーキテクチャでのシステムの動作のイメージを以下に示す．

1. アプリケーションからProxy-Agentに対して音声認識開始メッセージを送信
2. Proxy-Agentは Device-Adapter経由で認識対象となる音声データを取得
3. Proxy-Agentは Engine-Adapter経由で音声認識エンジンに対して音声データを入力
4. Proxy-Agentは Engine-Adapterから認識結果を取得
5. Proxy-Agentはこのとき取得した音声データと認識結果をログとして蓄積(モニタリング機能)
6. アプリケーションは，Proxy-Agentから認識結果を取得

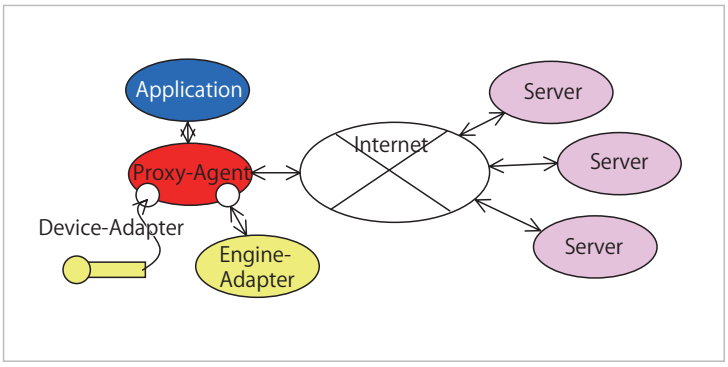

図-5　Proxy-Agentアーキテクチャ

Proxy-Agentはその名前の通り“プロキシ”として振る舞い，アプリケーションと音声認識エンジンのメッセージの送受信を中継する役割を担う．そのために，アプリケーションからは音声認識エンジンが提供するすべての機能に対するメッセージの送信による呼び出しが可能となる．また，音声認識エンジンが提供していない機能をProxy-Agentが提供する枠組みを備えることにより，音声認識エンジンの種類に依存しない，独自の機能の提供も可能となる．たとえば，音声認識エンジンが出力する結果をアプリケーションが処理しやすいフォーマット(JSON形式等)へ変換する機能や，アプリケーションの状態や認識対象の特徴，前後のユーザの操作等のランタイム情報をモニタリングのログに対して付与する機能の提供が可能となる．さらにProxy-Agentは，ネットワーク経由で外部のサービスとの連携機能を提供することで，双方向型開発パラダイムの実現に有効な機能の実現を可能にする．

1. モニタリング機能によって蓄積されたログデータを分析用サーバへ送信(フィードバック機能)
2. 開発者が分析用のサーバにて収集されたデータを分析
3. 開発者が作成した新モデルや機能追加用部品をProxy-Agentに対して配信(配信情報の取得機能)

この枠組みをさまざまなアプリケーションに対して導入することで，フィードバック情報が1カ所に蓄積され，多数のユーザを対象とした振舞いの分析と統計データの算出が可能となる．最終的には，実使用環境に適した語彙や類義語の作成，モデルの構築が可能となり，双方向型の開発パラダイムの実現が見込まれる．現状では，Sphinx 4，VORERO，Juliusに加え，プロジェクトで東工大が開発したT^3の4つのデコーダがProxy-Agent対応になっている．

連携サーバ群

Proxy-Agentの周囲にはさまざまなサービス機能を持つサーバ群を配し，開発者間，あるいは開発者とユーザとの間での知見や資源の共有を実現した(図-6)．

利用ログ蓄積サーバは，前章で述べたようにProxy-Agent経由で音声応用システム利用時におけるユーザの生の振舞いをアップして蓄える．また，モニタリングによって得られたデータの視覚化・解析を行うツール群も用意して利用ログ解析を支援する．アプリ開発者は，これらの道具立てを利用して，システムをユーザに提供した後も，システムの改良を継続的に行うことができる．またその改良結果をProxy-Agent連携による配信機能によって，ユーザに届けることができる．

配信機能は，音声認識アプリケーションを構成す

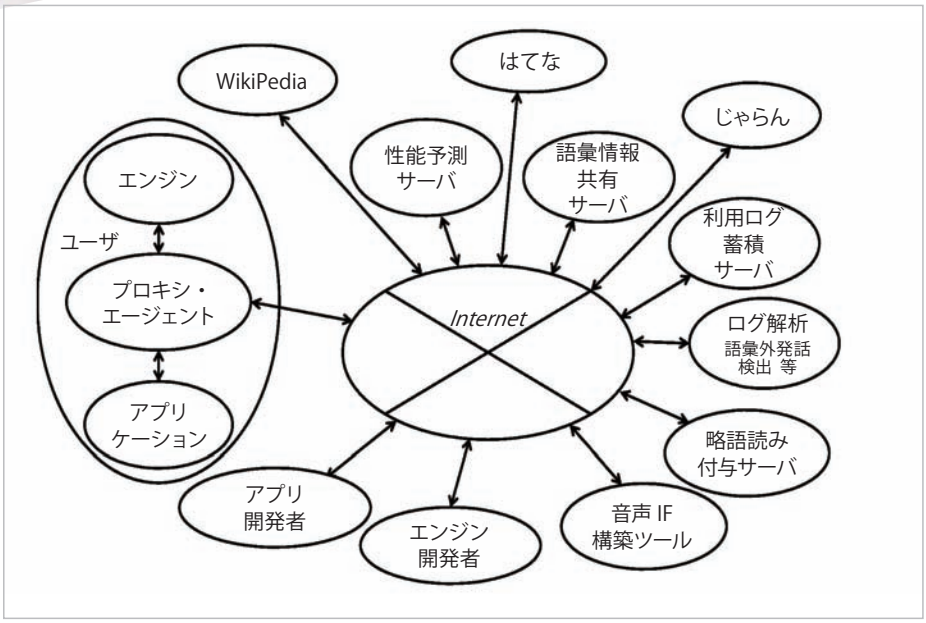

図 -6　ユーザ・アプリ開発者・エンジン開発者間の連携に基づく双方向的開発パラダイム：Proxy-Agent を核とするサーバ連携の形

るコンポーネントや言語資源の部品を共有する仕組みとして，Proxy-Agent のプラットフォームである Eclipse RCP の枠組みに従って提供されている．この枠組みは更新部品の配信としてだけでなく，音声認識応用システムに必要なさまざまな機能部品の共有に利用可能である．アプリケーションやエンジンの構成単位を Eclipse プラグインとして用意すれば，それらプラグインは共有可能となり，アプリケーション開発者は，Eclipse 上の GUI を用いて，ここから必要なプラグインを選択することができる．オープンな枠組みを採用していることで，共有可能な部品の作成とその共有を，エンジン開発者やサイト運営者だけでなく，すべての開発者に対して可能とし，より広い範囲での部品の共有を実現する．

語彙情報共有サーバは，Web 上の言語資源から語彙情報を定期的に収集することで，音声認識システムに必要となる語彙を効率的に生成・管理する [5]．単語に付与したタグの集合によって語彙を表現する機能を持ち，これによってアプリケーション用語彙の新規作成から，その継続的な更新まで包括的な解法を提供する．このことによって，これまで各々の開発者がアプリケーションごとに用意していた語彙定義プロセスは一元化され大幅に効率化されることが期待できる．また，Proxy-Agent の配信機能によって，Web に現れる新出単語についても利用者に届けることができ，自動的に更新されることとなる．ここでは，NEC が提供する略語読み付与サーバと連携して，略語の扱いも可能となる．

旭化成が開発した性能予測サーバは，標準評価データを用いてデコーダの評価を行うことができる．近い将来，少数の評価サンプルを与えることでデコーダの性能予測分布を与えることを検討している．この機能により，アプリ開発者は，ターゲットとなる環境において，どの程度の認識性能が見込まれるのかを知った上でシステムの設計が可能となる．

そのほか，開発の知見をパターン・ランゲージの形で表現し，これを開発者間で共有する仕組みも実現されている．音声認識応用システムの開発において直面する代表的な問題とその解決方法を記した手引書を，多くの技術者が共同して作ることになる．これにより，経験の少ない技術者でも，先人の失敗を繰り返すことなく効率的にシステムを開発することができる．

もちろん，開発支援に必要な機能はプロジェクトで開発できたもののほかにも数多くあって，それらについては今後地道に開発を続ける必要がある．Proxy-Agent は，機能拡張を容易に行えることを特徴としているため，新たなサーバが開発されたと

きも，その効果を簡単にアプリケーション側に伝えることが可能になっている．

以上のようなサーバ群との連携に基づいて，音声認識応用システムを開発・運用することで，開発の弱点であった音声ソリューションがカバーされ，良質の音声認識応用システムが実現されることが期待される．

展望

本稿では，ユーザ・開発者の連携に基づいて，自動的・継続的なシステム開発と修正システムの再配信を可能にする音声アーキテクチャについて述べた．このアーキテクチャでは，音声ソリューションにコストをかけられない体制においても，良質なシステムを開発できる可能性がある．最近では Google のボイスサーチなど，魅力を感じさせるシステムも出てきているが，これらの成功しているシステムは，すべてこのような「システムが成長していく仕組み」を持っている．本稿で述べたアーキテクチャは，これをアプリケーションとエンジンの双方に非依存に行い，スケールメリットを生じさせるための仕組みでもある．

この継続的な開発と再配信によって商品の質を高める方法論は，ソフトウェアの世界ではごく普通のことである．しかしながら，ソフトウェア以外の世界ではなかなかこれを受け入れる素地がないようだ．「我が社がユーザの手を借りなければ完成に至らない未熟なシステムを市場に出すなどまかりならん」という発想である．未熟なシステムでも，成熟への道筋をつけた上で市場にでるなら問題は少なかろうが，そういった考えが受け入れられないのは残念である．

また，通常であれば音声のようにヒューマンファクタが使い勝手に深く影響を持つシステムの普及には，多かれ少なかれデファクトをとるシステムの存在が必要である．システムがユーザに受け入れられるためには，ユーザがある程度システムに慣れる必要があり，このためにはインタフェースにアプリケーションを超えた一貫性が求められる（どのアプリケーションを使ってもおおよそ同じ使い方ができることが望まれる）．通常，それを主導するのはデファクトをとるシステムである．しかしながら，音声認識においては，なかなかデファクトをとるシステムが実現しない．現状，エンジン，ソリューション，アプリケーションとバランス良く開発できる体制がなく，広い応用分野を対象としたデファクトをとれるほどに優良なシステムが作りにくいからと考える．一社であるいは 1 つのアプリケーションでデフォルトをとれるシステムを開発できないならば，業界全体でインタフェースに一貫性を持たせ標準を作りだしてはどうかと思うのであるが，これもまた結構綱引きがあって難しいようだ．であれば，せめて開発知見の共有を進めてほしいと願っている．主観的な主張に終始すれば，なかなか綱引きは終わらない．客観的なデータの分析は我々の進むべき方向を教えてくれるに違いない．共同で音声認識の価値を周知させ，パイを広げることこそが重要と考える．音声認識システムの使い方さえ適切であれば，我々がユーザとしてその恩恵にあずかる場面が少ないはずがない．

参考文献

1) 経済産業省 高度情報通信機器・デバイス基盤プログラム　情報家電センサー・ヒューマンインタフェースデバイス活用技術開発「音声認識基盤技術の開発」最終成果報告書．
2) 小林哲則：音声認識応用システム開発の新パラダイム，情報処理学会音声言語情報処理研究会，SIG-SLP-74，pp.109-114（Dec. 2008）．
3) 2005 年度 新エネルギー・産業技術総合開発機構音声認識技術実用化に向けた先導研究事業「音声認識技術実用化に向けた先導研究」報告書．
4) Nakano, T., Fujie, S. and Kobayashi, T. : Extensible Speech Recognition System Using Proxy-Agent, Proc. IEEE ASRU 2007（Dec. 2007）．
5) Sasaki, H., Nakano, T., Fujie, S. and Kobayashi, T. : A Collaborative Lexical Data Design System for Speech Recognition Application Developers, ACM CSCW 2010 (Feb. 2010)．

（平成 22 年 9 月 7 日受付）

小林 哲則（正会員）koba@waseda.jp
1985 年早大大学院博士課程修了．工学博士．同年法政大・工・講師，同助教授を経て，1991 年より早大勤務．現在，理工学術院情報理工学科教授．音声対話システム，ヒューマンインタフェースの研究に従事．

中野 鐵兵（正会員）teppei@gowell.org
2009 年早大大学院博士課程修了．博士（工学）．2006 年より早大 IT 研究機構客員研究員．音声認識応用システム開発支援技術，ヒューマンインタフェースの研究開発に従事．