特　集「ロボットは東大に入れるか？」

統合研究基盤：質問応答システムの互換コンポーネント化による再利用性向上と開発自動化支援

A Research Platform Using Question-Answering System for Reusability and Automation in Developments

狩野　芳伸　科学技術振興機構さきがけ，国立情報学研究所
Yoshinobu Kano　PRESTO, Japan Science and Technology Agency (JST). / National Institute of Informatics.
kano@nii.ac.jp or kano@kachako.org., http://kachako.org/kano/

Keywords: interoperability, compatibility, reusability, platform, question-answering system.

1. はじめに

大学入試問題を解けるような質問応答システムの構築は複雑かつ多様な技術の集大成であり，専門的な知識を有する研究者ですら単独での研究開発はもはや不可能である．また，試行錯誤の過程で蓄積されるソフトウェア群は本プロジェクトで期待される重要な成果であり，誰もが利用しやすい形での社会還元が欠かせない．本稿では，再利用性と自動化という点に着目して，ツールやデータの共有・組合せ・実行を容易にするための統合研究基盤を紹介する．

入試問題を解くということは質問に対して答える作業の一種といえるので，既存の質問応答システムの仕組みを応用できる可能性が高い．ただし，既存の質問応答システムが想定する質問と答えは入試問題とは異なる．そのため入試問題に対応できるよう多かれ少なかれ改良が必要であり，科目によっては質問応答システムのごく一部だけ再利用したいという状況も想定される．そうなると，単に質問応答システム全体をプログラムとして提供しても再利用が難しいため，コンポーネントに分割して整理する必要がある．

また，入試問題を解くプログラムを作成する作業は，実際にはさまざまな試行錯誤が必要となり，プログラムに解答タスクを繰り返し実行させることになる．そのうち，入試問題を入力する部分と，解答を出力し採点する部分は共通化が可能である．プロジェクトにおける当面の目標は，大学入試センター試験を解けるようにすることであるので，まずはセンター試験の問題と解答について共通の研究基盤を提供することを狙う．

質問応答システムにせよ，センター試験問題の解答や採点にせよ，再利用性のためにコンポーネントに分割して共有し，研究者が興味ある部分に集中できるようなるべく作業を自動化・省力化するという目的に変わりはない．共有可能なコンポーネントは研究の進展とともに増加が見込まれるため，コンポーネントの形式を標準化したうえで互換性を担保することが重要である．

我々はUIMA（Unstructured Information Management Architecture）[Ferrucci 06]を標準化の枠組みとして用いることにした．UIMAはまさに上記のようなコンポーネントの共有を意図して提供されている国際標準であり，Apache UIMA[*1]としてオープンソースで公開されている．UIMAはすでにさまざまな研究開発で用いられている枠組みで，実装が安定している．標準化の利点は，いったんUIMAに準拠させてしまえばUIMAが担保する範囲での互換性ができ，その利用方法はUIMA一般のドキュメントで学べるということにある．さらに，UIMAに準拠した実行環境であれば好みの環境で実行できるし，UIMAのAPIなどを用いて自前の環境に組み込むこともできる．また，さまざまな研究グループからUIMAコンポーネントが公開されているため，それらを同時に利用することも容易である．GATE[Cunningham 02]などほかの枠組みとの相互変換によりさらに多くのツールを使うこともできる．必要に応じて枠組み間の変換を行えばよいから，UIMA準拠であることは制約にならない．

コンポーネントの可搬性や意味的な互換性などを考慮すれば，UIMA準拠という以上に開発者の自動化・省力化をサポートすることができる．UIMA準拠の実行環境としては，Kachako[Kano 12a, 狩野 12c]が最も自動化を考慮して設計されている．本稿で述べるコンポーネントは，単にUIMA準拠というだけでなく，Kachakoの要求する自動化に耐え得る実装になっている．また，

*1 http://uima.apache.org/

Kachakoの提供するほかのコンポーネントとの意味的な互換性も考慮されている．質問応答システムのコンポーネント分割にあたっては，NTCIRのACLIA[Mitamura 08, Mitamura 10]によるタスク分割を基準とした．

本稿では，2章でまずUIMA，Kachako，ACLIAについて簡単に紹介する．3章で研究基盤について記述し，4章で本稿を締めくくる．

2. 背　　景

2・1　UIMA

UIMAの実行単位はコンポーネントと呼ばれ，コンポーネントを組み合わせて実行可能なUIMAワークフローを作成する．UIMA自体は基本的にコンポーネントそのものの提供はしないため，UIMAの利用者は通常，自前のコンポーネントを作成するか，第三者のコンポーネントを使うことになる*2．UIMAコンポーネントは，メタデータを記述するdescriptorと呼ばれるXMLファイルと，対応する実行ファイルに分離されている．UIMA APIの実装はJavaとC++の双方で提供されており，コンポーネントの実装には通常これらのAPIを用いる．

UIMAワークフロー実行時のデータ構造はCASと呼ばれる汎用構造に統一されている．一般的なUIMAコンポーネントはCASを一つ受け取り処理結果を加えてから同じCASを返す．CASは生テキストを保持する部分と，テキストへの付加情報を保持する部分に分かれている．テキストへの付加情報についてはデータの型付けが必須であり，開発者はtype systemと呼ばれる型階層をtype system descriptorとしてXMLファイルで定義する．UIMA標準のデータ型は数値や文字列，配列など基本的な型のみのため，type systemの互換性を考慮する必要がある．UIMAコンポーネント同士であれば形式的な互換性があるのに，実際には組み合わせても意味のある実行結果が得られないという事態が起こるからである．ただし，これはソフトウェアを組み合わせて実行する際に生じる一般的な問題であり，UIMAが問題なのではない．本稿ではこれを意味的な互換性と呼ぶ．

作成したUIMAコンポーネントは，プログラムを修正することなくWebサービスとしても展開できる．展開したWebサービスはUIMAコンポーネントとしてWebサービスでないコンポーネントと自由に混合してワークフローを構成できる．Webサービスでの展開は，実行に特殊な環境が必要なツールや，実行ファイルを公開できないツールに適している．

UIMAコンポーネントにはaggregateと呼ばれるほかのコンポーネント群を子としてもつ「箱」もある．「箱」は子コンポーネントの実行順序をプログラマブルに制御できる．「箱」は入れ子にできるため，理論的にはほとんどあらゆる実行順序を実現できるが，多くの利用では単に順次実行するパイプラインであることが多い．

2・2　Kachako

KachakoはUIMA準拠の統合プラットフォームと互換UIMAコンポーネント群を提供している．Kachakoの目的は自然言語処理におけるユーザタスクを徹底した自動化によりサポートすることであり，プラットフォームおよびコンポーネントのインストール，ワークフロー生成，Webサービス展開，大規模処理，結果の視覚化，汎用比較評価などを全自動でサポートする機能を提供している．比較評価機能の応用として混合器の作成もできる．

自動化を実現するためにはコンポーネントの可搬性と互換性が必須である．3章で述べるコンポーネントはそのような可搬性や互換性を満たした形で実装されている．

2・3　ACLIA

ACLIA（Advanced Cross-Lingual Information Access）はNTCIR-7およびNTCIR-8で行われた質問応答タスクである．ACLIAでは，参加者が質問応答タスク全体の中の興味ある部分のみでも参加できるよう，中間結果を定義してタスクを切り分けており，参加者はそのフォーマットに従って結果を提出し評価される仕組みになっている．

3. 研 究 基 盤

本研究基盤はUIMA準拠とすることで，ユーザの必要に応じた多様な方法での利用を可能とした．図1に研究基盤の概念的な構成を示す．UIMAコンポーネントを実行したい場合は，UIMA準拠の任意の実行プラットフォームを用いてUIMAワークフローとして実行すればよい．実行プラットフォームにはグラフィカルユーザインタフェースからプログラマ向けのAPIアクセスを提

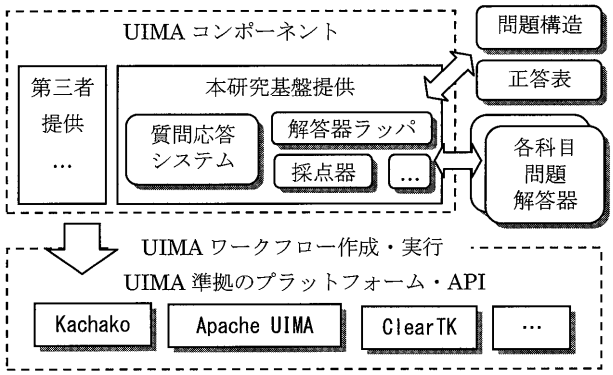

図1　研究基盤とその利用方法の概念図．
コンポーネントを組み合わせてワークフローを生成・実行する

*2 Apache UIMAはオープンソースでありどのような改変も可能であるが，本稿では互換性のためUIMAの提供するAPIは変更せず，ワークフローの実行エンジンもUIMA提供のものを使うことを前提とする．

供するものまでさまざまである．特に，2·2節で触れたKachakoは，テキストや画像などの非構造化データにおける相互運用性と自動化を実現するもので，本プロジェクトにおいて必要な相互運用性に関わる一般的な議論はその中ですでに尽くされているといえる．本研究基盤の課題は，そうした議論のうえに，入試問題を解くために利用可能な実際のコンポーネントを実装することにある．各コンポーネントはオープンソースで公開の予定である．また，自動実行可能なものはKachakoからも配信する．

3·1 質問応答システムのコンポーネント化

質問応答システムのコンポーネント化にあたり，既存システムとして横浜国立大学で開発されたMinerVAとカーネギーメロン大学で開発されたJavelin IVを提供いただき，これらをUIMAコンポーネント化した．

MinerVAにはFactoid [Mori 05]とNon-factoid [石下09]の二種の日本語質問応答システムが含まれている．MinerVAの実装はPerlであるが，可搬性のためJavaで再実装したうえでUIMAコンポーネント化を行った．

Javelin IV [Shima 08]はJavaで実装された質問応答システムで，そのうち日本語のベースラインシステムを提供いただいた．MinerVAから作成されたコンポーネントと互換になるようにコンポーネント化を行った．

§ コンポーネント化の設計

まず，2·3節で述べたACLIAのタスク分割を基準としてコンポーネント化を行った．具体的には，質問解析·文書検索·回答抽出·回答選択の各コンポーネントである．また，いずれの質問応答システムも内部的に形態素解析や検索エンジンといった外部ツールを呼び出して利用している．こうした外部ツールも極力UIMAコンポーネント化して切り出すようにし，細かな粒度でコンポーネントの置換や共有が行えるように設計した．

コンポーネントの互換化設計における一般的な指針は，機械的に読取り可能で明示的な入出力メタデータ記述を行える形に実装することである [狩野 12]．すなわち，コンポーネントの処理はなるべく自己完結するようにして，外部から見てそのコンポーネントがどのような役割を果たすのかを明確にする必要がある．言い換えると，コンポーネントのソースコードを解読しなくとも，メタデータ記述で役割が判断できるよう，適切な入出力を設定する必要がある．そうすれば，タスクに応じてプログラムを修正することなくより多くの場合で再利用できるはずである．さらに，入出力情報からワークフローの自動生成なども可能で，理想的な相互運用の形といえる．

形態素解析などの文章を受け取ってそこに追加情報を加えるようなツールは比較的単純であり，適切なデータ型を定義したうえで入出力を記述すればよい．しかし，質問応答システムの場合は，同じ質問に対して解答を多数出力する，あるいは検索エンジンを呼び出す際にクエリを多種生成したうえ各クエリについて上位100件の検索結果を用いる，といったことがしばしば行われる．この場合，新たに生成されるのはテキストへの追加情報だけではなく，テキストそのものということになる．質問文·クエリ·検索結果·解答文などを一つのCASに混在させてしまうと，コンポーネントの作成時に特定の構造を前提として実装しなければならず，汎用性が失われてしまう．

そこで，UIMAのsofaという仕組みを用いて汎用性をなるべく保持するようにした．sofa（Subject OF Analysis）とは多面的なデータを保持するための仕組みで，単一のCASの中にsofaと呼ばれる仮想的なCASを複数保持できる．必要に応じて仮想的なCASを別のCASにコピーすれば，特定の用途を前提につくられたコンポーネントと形態素解析のような一般的なコンポーネントを混在させて利用できる [Kano 12b]．質問応答システムの場合は，新たなテキストが増えるごとにsofaを生成するようにした．

3·2 大学入試センター試験の解答と採点

本プロジェクトの最初の課題は，大学入試センター試験の解答器を作成することにある．試験問題の解答器は，解答する科目によって仕組みが大きく異なることが予想される．また，同じ科目であっても，問題によってその性質が異なるため，一人の研究者で一科目分の問題をすべて解答できるようなシステムを構築するのは難しいと思われる．そのため，別個につくられた解答器を総合して実行できるような仕組みが理想的である．

解答器の仕組みにかかわらず，問題とその解答は基本的に共通といってよい．本プロジェクトでは，大学入試センター試験問題アノテーション済みデータ（問題構造·問題分類）を作成しており，国立情報学研究所との間でデータ利用許諾に関する覚書を締結すれば研究目的で利用可能である．研究基盤ではこれをUIMAを直接用いない解答器との共通入出力フォーマットとして利用する．

§1 問題構造と正答表のリーダおよびライタ

アノテーション済みデータには試験問題そのものに問題構造を追加記述したXMLファイル群と，解答を記述できる正答表XMLファイル群がある．本基盤ではすべてUIMA準拠としているため，問題構造や正答表のXMLデータをUIMAの形式に変換する問題構造·正答表リーダおよびライタコンポーネントを提供する．

§2 解答器の作成とラッパ

解答器は問題構造XMLを入力，正答表XMLを出力とした上で，どの科目のどの問題分類を解けるかを宣言するようにする．基盤側ではそのように作成された解答器を前提に，入出力を受け渡してUIMAとの相互変換を行うラッパコンポーネントを提供する．この場合，解答器の作成でUIMAの知識が必要とされることはない．

解答器を最初からUIMAコンポーネントとして作成

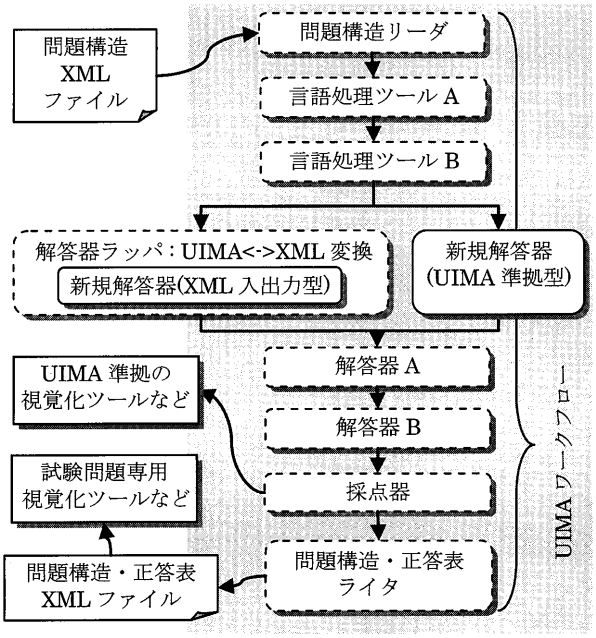

図2　解答器の研究開発で想定される典型的な UIMA ワークフローと利用方法.
解答器を UIMA 準拠で作成する場合（右）と XML 入出力の場合（左）を示す

することもできる．解答器の内部構造も適切に分割してコンポーネント化すれば，前節の質問応答システムのように共有と再利用が容易になる．図2に典型的な例を示す．

§3 採点器

最後に，UIMA の形式に変換された正解と解答器の出力を比較する採点コンポーネントを提供する．また，ベースラインとして，簡易なランダム解答器を提供する．

これらのコンポーネントの組合せにより，UIMA ワークフローとして問題の読込み・解答・採点までを行える．また，問題構造 XML 形式で書き出すライタコンポーネントを用いれば，問題構造 XML 形式を前提とした視覚化ツールなどもそのまま利用できる．

4. おわりに

本稿では，大学入試問題の解答器作成のための研究基盤として，質問応答システムおよび大学入試センター試験の解答採点について，互換コンポーネント化して共有できる研究基盤を紹介した．今後の発展としては，二次試験への対応と，解答器に利用できるコンポーネントの増加を考えている．また，同種の問題に対する解答器が複数あれば，結果を適切に混合することで全体の性能向上が見込めるため，混合器の実装が課題となろう．

謝　辞

国立情報学研究所の神門典子教授には ACLIA および質問応答システム全般についてご指導とご助言をいただいた．横浜国立大学の森 辰則教授・石下円香氏およびカーネギーメロン大学の三田村照子教授・嶋 秀樹氏にはそれぞれ質問応答システムをご提供いただいた．ここに深謝申し上げたい．大学入試センター試験の問題および解答データについては，株式会社ジェイシー教育研究所が販売する「大学入試センター試験問題データベース　センター Ten 2011　通常版 全教科セット」を利用した．

◇ 参 考 文 献 ◇

[Cunningham 02] Cunningham, H., Maynard, D., Bontcheva, K. and Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications, *40th Anniversary Meeting of the Association for Computational Linguistics*, pp. 168-175, Philadelphia, USA（2002）

[Ferrucci 06] Ferrucci, D., Lally, A., Gruhl, D., Epstein, E., Schor, M., Murdock, J. W., Frenkiel, A., Brown, E. W., Hampp, T. and et al.: Towards an interoperability standard for text and multi-modal analytics, *IBM Research Report*（2006）

[石下 09] 石下円香・佐藤 充・森 辰則：Web 文書を対象とした質問の型に依らない質問応答手法，人工知能学会論文誌，Vol. 24, No. 4, pp. 339-350（2009）

[Kano 12a] Kano, Y.: Kachako: Towards a data-centric platform for full automation of service selection, composition, scalable deployment and evaluation, *19th Int. Conf. Web Services*（*IEEE ICWS 2012*），Hawaii, USA（2012）

[Kano 12b] Kano, Y.: Towards automation in using multi-modal language resources: compatibility and interoperability for multi-modal features in Kachako, *8th ed. of the Int. Conf. on Language Resources and Evaluation*（*LREC 2012*），Istanbul, Turkey（2012）

[狩野 12c] 狩野芳伸：Kachako: 誰でも使える全自動自然言語処理プラットホーム, 2012 年度人工知能学会全国大会（第 26 回）（2012）

[Mitamura 08] Mitamura, T., Nyberg, E., Shima, H., Kato, T., Mori, T., Lin, C.-Y., Song, R., Lin, C.-J. and Sakai, T., et al.:Overview of the NTCIR-7 ACLIA: Advanced cross-lingual information access, *NTCIR-7 Workshop*（2008）

[Mitamura 10] Mitamura, T., Shima, H., Sakai, T., Kando, N., Mori, T., Takeda, K., Lin, C.-Y., Song, R. and Lin, C.-J., et al.: Overview of the NTCIR-8 ACLIA Tasks: Advanced cross-lingual information access, *NTCIR-8 Workshop*（2010）

[Mori 05] Mori, T.: Japanese question-answering system using A* search and its improvement, *ACM Trans. Asian Language Information Processing*（*TALIP*），Vol. 4, No. 3, pp. 280-304. New York, NY, USA: ACM（2005）

[Shima 08] Shima, H., Lao, N., Nyberg, E. and Mitamura, T.: Complex crosslingual question answering as sequential classification and multi-document summarization task, *NTCIR-7 Workshop*（2008）

2012 年 8 月 1 日　受理

著 者 紹 介

狩野　芳伸（正会員）

2001 年東京大学理学部物理学科卒業，2007 年同大学院情報理工学系研究科博士課程単位取得退学．東京大学情報理工学系研究科特任研究員などを経て，2011 年科学技術振興機構さきがけ研究者・国立情報学研究所外来研究員．博士（情報理工学）．言語資源の互換性と相互運用性，および自然言語処理の自動化プラットフォームの研究に従事．