能動的情報資源の概念を用いた知識検索支援システム

羽鳥 秀明† 阿部 亨†† 木下 哲男††

† 東北大学大学院 情報科学研究科
†† 東北大学 情報シナジーセンター

概要 近年，組織内に遍在する知識を電子化し，メンバー間で効率的に共有・活用することで，高度な知的活動を促進しようとする考えが広まってきており，これを支援するための知識検索システムが多数提案されている．しかし，従来のシステムは，トピックの異なる複数の知識を組み合わせ利用する場合への考慮が十分でなく，そのような場合に対しては，既存システムを用いたとしても，ユーザが多くの作業を行う必要があり，ユーザに課される負担は依然として大きい．これに対し，本稿では，ユーザが複数の知識を組み合わせ利用するために行う一連の検索作業を包括的に支援するシステムを提案する．提案システムでは，能動的情報資源の概念を用い各知識を構造化し，知識自体に能動性・自律性を与えている．これにより，従来ユーザに委ねられていた作業の一部を知識側に代行させ，検索に要する作業の効果的支援を図っている．

Knowledge Retrieval Support System Based on Active Information Resource

Hideaki HATORI†, Toru ABE††, and Tetsuo KINOSHITA††

† Graduate School of Information Sciences, Tohoku University
†† Information Synergy Center, Tohoku University

Abstract To support the knowledge sharing and utilization in a group, various knowledge retrieval systems have been proposed. However, the existing systems assume that a user retrieves and uses a set of knowledge about a single topic at once. Therefore, even if the user could utilize those systems, many tasks remain for the user to retrieve knowledge about different topics, organize them and get a combined knowledge to be used. To overcome this problem, we propose a novel knowledge retrieval system. In the proposed system, employing the concept of active information resource, each knowledge is structured and managed as an agent to make it autonomous, and then knowledge itself can assist the user actively with respect to his/her retrieval tasks.

1. まえがき

近年，組織内に遍在する知識を電子化し，メンバー間で効率的に共有・活用することによって，高度な知的活動を促進しようとする考えが広まってきている．これら電子化された知識の共用・活用を支援するために，メタデータ [1]，N-gram インデキシング [2]，あるいは，形態素解析 [3] 等を利用した知識検索システムが数多く提案されている．しかし，それらの多くは，単一のトピックに関する知識を検索・利用する状況を想定しており，例えば，システム構築のために複数の製品を連携させる場合のように，トピックの異なる複数の知識を組み合わせ利用する状況は十分考慮されていない．そのため，従来の知識検索システムを用いた場合，トピックの異なる複数の知識を組み合わせ利用するためには，異なるクエリによる検索の反復や，取得した知識の中での適切な組み合わせの選択など，ユーザが多くの作業を行う必要があり，ユーザに課される負担は依然として大きいままである．

これに対し，本稿では，ユーザが複数の知識を組み合わせ利用するために行う一連の検索作業を包括的に支援するシステムを提案する．提案するシステムでは，能動的情報資源（Active Information Resource: AIR）[5] の

概念を用い各知識を構造化することで，知識自体に能動性・自律性を与えている．これにより，従来ユーザに委ねられていた作業の一部（特に，知識の組み合わせの選択）を知識側に代行させ，検索に要する作業の効果的支援を図っている．

本稿の構成は以下のとおりである．まず第 2 章で複数の知識を組み合わせ利用する場合の従来の検索システムの問題点について指摘する．次に第 3 章で複数の知識の組み合わせ手法と能動的情報資源の適用について述べ，第 4 章で提案システムの設計と動作について説明する．最後に第 5 章でまとめる．

2. 知識の検索

本章では，トピックの異なる複数の知識を組み合わせ利用する場合に必要な作業について説明し，従来の検索システムの問題について述べる．

2.1 複数の知識を組み合わせ利用する場合の作業

電子化されたドキュメントとして知識を蓄積した場合，更新，検索，提示など，知識を再利用・共有するための各処理はドキュメント単位で行われる．このため，1 つのドキュメントに多数のトピックに関する知識が含まれると，トピック単位で各処理を行うことができず，知識の再利用・共有の観点から種々の問題が生じる [4]．従って，ドキュメントとして電子化された知識を再利用・共有するためには，1 つのドキュメントに含まれるトピック数が少ない方が効果的であり，理想的には，1 つのドキュメントに単一のトピックに関する知識を記述した形式（以降，single-topic ドキュメントと呼ぶ）が望ましいと考えられる．

しかし，実際に知識を利用する際には，異なる複数のトピックに亘る知識が要求される場面も多い．このような状況に対応するためには，異なる複数のトピックに関する知識が記述されたドキュメント（以降，multi-topic ドキュメントと呼ぶ）を介し，トピックの異なるドキュメント同士を組み合わせ利用する必要がある．例えば，複数の製品を連携させ 1 つのシステムを構築する場合，ユーザは，図 1 に示すように，製品を連携させるためのノウハウ（multi-topic ドキュメントに相当）と各製品のマニュアル（single-topic ドキュメントに相当）を組み合わせたものを用いることになる．

このように複数の知識を組み合わせ利用するためには，ユーザは，各トピックに関する知識を検索するだけでなく，整合の取れた知識の組み合わせを選択する必要が生じる（複数の製品を連携させる前述の例ならば，組み合わせる知識での製品のバージョンを揃える等）．従って，ユーザによる知識の活用を支援するためには，次の作業

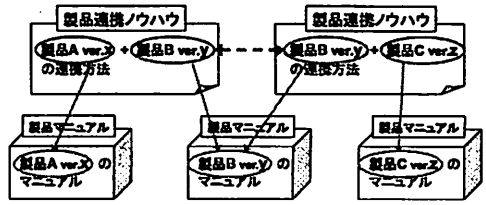

図 1 トピックの異なる複数の知識の組み合わせ (製品 A,B,C を組み合わせてシステムを構築する際に必要な知識)

を代行する機構の実現が効果的であると考えられる．

- 各トピックに関する知識を検索
- 整合の取れた知識の組み合わせを選択
  - multi-topic ドキュメント間での整合性を検証
  - multi-topic ドキュメントと single-topic ドキュメント間での整合性を検証

2.2 従来の検索システム

現在，種々の知識（ドキュメント）検索システムが提案されているが，その多くは，1 つのトピックに関する知識を検索・利用する状況を想定している．そのため，これらを用いた場合，トピックの異なる複数の知識を組み合わせ利用するためには，ユーザが，異なるクエリにより検索を繰り返す必要がある．

これに対し，異なるクエリによる検索を自動的に繰り返すシステムも提案されている [6]．このシステムでは，検索によって得られた情報を取得することでユーザの知識が変化することに着目し，その変化によって生まれるユーザの新たなニーズをシステム側が推測することで，システムが代行して新たなトピックに関する情報を取得してユーザに提示している．また，AreaView2001 [7] では，新たな分野に関する知識を得ようとするユーザに対して，その分野に関連する複数のトピックの知識の集合をシステムが代行して取得し，取得した知識を構造化した上でユーザに提示している．しかし，これらのシステムでは，検索対象となる複数のトピックをユーザが指定することができず，また，整合の取れた知識の組み合わせを選択する場合も考慮されていない．従って，トピックの異なる複数の知識を組み合わせ利用するためには，ユーザが多くの作業を行う必要がある．

3. 能動的情報資源 (AIR) を用いた知識検索支援

本章では，まず，ユーザが複数の知識を組み合わせ利用するために行う作業をシステム側で代行する手法を提案し，次に，能動的情報資源（AIR）の枠組みに基づく提案手法の実現法について説明する．

3.1 複数の知識の組み合わせ

ここでは，複数の製品を連携させ1つのシステムを構築する場合の知識を例として，提案手法について説明する．今回の例では，以下の2種類の知識を扱う．

- **製品マニュアル**
  製品を開発したベンダーやコミュニティが発行している各製品の操作説明等を記述したドキュメント
- **製品を連携させるためのノウハウ**
  複数の製品を連携して動作させる際の特徴，システム構成，動作環境，導入・運用方法，事例等を記述したドキュメント

ここで，「製品マニュアル」と「製品を連携させるためのノウハウ」は，前節で述べたsingle-topicドキュメントとmulti-topicドキュメントに各々相当する．今回の例では，マニュアルとノウハウの内容は，対象とする製品および製品のバージョンに依存し，また，ノウハウには，2つの製品を連携させるための知識が記述されているものとする．従って，2つの製品を連携させる場合は，両製品を連携させるためのノウハウと両製品のマニュアルの組み合わせを利用することになり，3つ以上の製品を連携させる場合は，複数のノウハウと各製品のマニュアルの組み合わせを利用することになる．

一般に，各製品には複数のバージョンが存在し，連携可能な他の製品のバージョンや連携させるためのノウハウはバージョン毎に異なる．従って，連携させる製品名のみクエリとして入力し，必要なノウハウやマニュアルを検索する場合，検索結果からバージョンの整合の取れたノウハウ，マニュアルの組み合わせを選択する必要が生じる．

ノウハウ（multi-topicドキュメント）とマニュアル（single-topicドキュメント）間での整合については，ノウハウやマニュアルがシステムに追加・登録された段階で，製品名とバージョンの一致する組み合わせを決定しておけば，実際に知識を検索・利用する際にも，その組み合わせを用いることができる．一方，ノウハウの組み合わせに関しては，ユーザがクエリとして製品名を指定するまで，連携させる製品が不明であるため，事前に組み合わせを決定できない．

そこで，本稿では，適切なノウハウの組み合わせを以下の2段階で決定する手法を提案する．

（1） まず，ユーザが入力したクエリに適合する（ノウハウ内に記述された2製品が両者ともクエリ内に含まれる）ノウハウを決定する．

（2） 次に，決定されたノウハウ毎に，不足する知識を持つ他のノウハウを順次探索することで，ユーザのクエリを完全に満たす（クエリに含まれる製品を全て含む）ノウハウの集合を決定する．

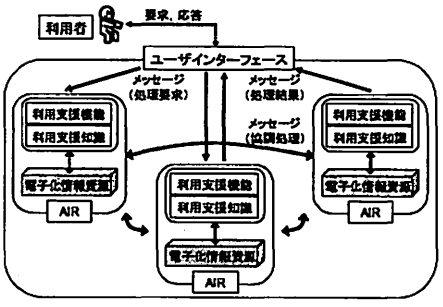

図2 AIRの構成

ノウハウ間での整合性（製品バージョンの整合性）に関しては，(2) のステップで他のノウハウを探索する際，新たに獲得されるノウハウ中の製品バージョンと獲得済みのノウハウの製品バージョンを比較することで，整合の取れたノウハウの組み合わせの選択を実現する．

3.2 能動的情報資源（AIR）

前節で述べたように，提案手法では，ユーザからのクエリに適合した各ノウハウが，自身には不足している知識を得るために，他のノウハウに対する検索を行い，これを繰り返すことで，クエリを完全に満たす（クエリに含まれる製品を全て含む）ノウハウの集合を決定している．また，製品バージョンの整合性の検証は，ノウハウ検索の際に行っている．このような機構の実現には，選択された各ノウハウ自身が能動的にクエリの生成・バージョンの検証を行うシステムアーキテクチャが有効であると考えられる．そこで，提案手法の実現にあたり，能動的情報資源（AIR）の概念を適用することで，ノウハウ自身に能動性を持たせ，各ノウハウが自律的にクエリの生成・バージョンの検証を行うシステムの実現を図る．

AIRは，情報資源（電子化された各種データ）の構造を強化し，従来ユーザに委ねられていた情報資源に対する操作の一部を情報資源自身に代行させるものである．具体的には，図2に示すように，利用支援知識（情報資源に関するメタデータや情報資源を操作するための知識）と利用支援機能（情報資源を操作する機能）とを情報資源に付加したエージェントとして各AIRを構成する（情報資源のAIR化）．これにより，ユーザからの要求メッセージを受信したAIRは，自身の利用支援知識・機能を用いて，自身が管理する情報資源に対する操作を代行し，その結果を応答メッセージとしてユーザへ送信できるようになる．また，AIRは，単独で動作するだけでなく，他のAIRとメッセージ交換を行うことで，分散環境上で相互に連携・協調し，より複雑・柔軟な操作を代行することも可能となる．

ノウハウをAIR化し，各ノウハウに自律性を持たせ

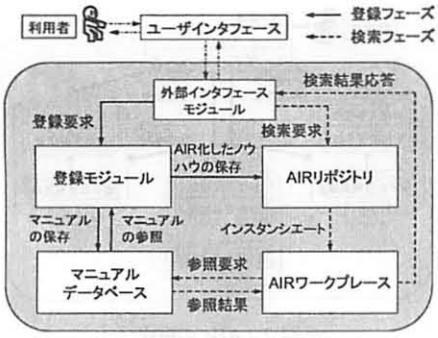

図 3　提案システムの構成

ることで，複数のノウハウの組み合わせの決定をノウハウ側に能動的に代行させることが可能となる．この結果，適切なノウハウの組み合わせの決定を効果的に代行できるシステムの実現が期待できる．

4.　AIR の概念を用いた知識検索システム

本章では，前章で述べた「複数の知識の検索」支援手法を AIR を用い実現するためのシステムを提案し，次に，このシステムを構成する AIR の設計，システムの動作について説明する．

4.1　提案システムの構成

提案するシステムの構成を図 3 に示す．本システムの動作フェーズには，新たな知識をシステムに登録するための「登録フェーズ」とユーザが複数の知識の組み合わせを検索するための「検索フェーズ」があり，これらのフェーズは，システムを構成する以下の 5 つのモジュールによって実現される．

- **登録モジュール**
  ユーザによって入力された知識を登録するモジュールである．本モジュールは，「製品マニュアル」と「製品を連携させるためのノウハウ」を登録する 2 種類の登録モードを持ち，メタデータの RDF 化やノウハウの AIR 化等の処理を行う．
- **マニュアルデータベース**
  電子化されたマニュアルやそのメタデータを保存するデータベースである．知識の検索の際に AIR によって必要な情報の参照が行われる．
- **AIR リポジトリ**
  登録モジュールによって生成された AIR を収容する機構である．ユーザによるクエリに対して要求を満たすと判断した AIR は，AIR ワークプレースに呼び出される（インスタンシエート）．
- **AIR ワークプレース**
  AIR 同士が連携・組織化を行う実行環境機構である．AIR ワークプレースにインスタンシエートされた AIR 同士で知識の適切な組み合わせ決定処理を行う．
- **外部インタフェースモジュール**
  ユーザインタフェースからの要求に対して，システム内のモジュールに要求を振り分け，結果をユーザインタフェースに返信するモジュールである．

表 1　AIR の主な利用支援知識

知識の種類	名称
情報資源の名前，作成者	title, creator
情報資源の ID	identifier
情報資源が表す製品名	product-name
情報資源が表す製品のバージョン	product-version
情報資源が表す製品の OS	product-os
情報資源が表す製品の価格	product-price
情報資源の内容を構成するキーワード	keywords

表 2　送受信されるメッセージの種類

メッセージ名	メッセージの種類
request-assortment	組み合わせクエリの開始要求
assortment-query	組み合わせクエリ
reply-assortment-query	組み合わせクエリに対する応答
request-re-assortment	再組み合わせクエリの開始要求
re-assortment-query	再組み合わせクエリ
reply-re-assortment-query	再組み合わせクエリに対する応答

4.2　AIR の設計

前述の複数の知識の組み合わせの選択を AIR を用いて実現するためには，種々の利用支援知識が必要である．その一部を表 1 に示す．また，例えば，次のような利用支援機能が導入される．

- F1：不足する知識を判断する機能
- F2：不足する知識を探索するクエリを生成する機能
- F3：ドキュメント間の整合性を検証する機能
- F4：他の AIR と協調・連携する機能
- F5：データベースを参照する機能

知識に利用支援知識と機能を付加することにより，種々の処理を AIR が自律的に実行可能になる．さらに，表 2 に示すようなメッセージを他の AIR と送受信することによって，適切な知識の組み合わせの決定が可能になる．

4.3　登録フェーズ

登録フェーズには，「製品マニュアル」を登録するモードと「製品を連携させるためのノウハウ」を登録するモードがある．

- **「製品マニュアル」の登録モード**
  まず，ユーザは，ユーザインタフェースを通じて，電子化されたマニュアルを登録する．その際，製品名，バージョン，OS，価格などのメタデータを合わせて入力する．登録モジュールによって，入力されたメ

```
<?xml version="1.0"?>
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
         xmlns:ha="http://www.ka.riec.tohoku.ac.jp/~hatori/rdf#">
 <rdf:Description rdf:about="/html/sample.html">
  <dc:title>PostgreSQL+Tomcatの連携ノウハウ</dc:title>
  <dc:creator>hatori</dc:creator>
  <dc:date>Sun Jun 1 00:00:00 JST 2005</dc:date>
  <dc:identifier>kh1116254371412</dc:identifier>
  <ha:abstract> 要約<ha:abstract>
  <ha:keywords>
    <ha:keyword ha:about="文字コード"
     ha:evaluated-value ="1.0"/>
            ⋮
  </ha:keywords>
  <ha:products ha:os="Linux">
    <ha:product1 ha:name="PostgreSQL" ha:version="7.3" />
    <ha:product2 ha:name="Tomcat" ha:version="5.5" />
  </ha:products>
</rdf:Description>
</rdf:RDF>
```

図 4　RDF の記述例

タデータを RDF に変換し，マニュアルデータベースに保存する．また，入力されたマニュアルから自動的に取得できる情報（作成日等）も RDF に変換し，マニュアルデータベースに保存する．

- **「製品を連携させるためのノウハウ」の登録モード**
  電子化されたドキュメント（ノウハウ）の登録とメタデータ（2 つの製品分）の入力は，ユーザインタフェースを介し，「製品マニュアル」の登録モードと同様に行われる．ノウハウの登録モードでは，さらに，以下に述べる手順で，ノウハウの AIR 化と，ノウハウとマニュアルの組み合わせの決定を行う．
  - ノウハウに記述されている内容を KeyGraph [8] を用いて解析することにより，ノウハウの内容を構成するキーワード情報を抽出し，RDF に追加する．ここで，KeyGraph とは，形態素解析によって分割された単語の共起頻度を基に，文書中からその文書が主張（説明）しようとしている主題となるキーワードを抽出する手法である．図 4 に，生成される RDF の記述例を示す．
  - ノウハウを情報資源とし，これに利用支援機能と知識を付加することで，ノウハウを AIR 化する．その際，RDF に変換されたメタデータを，利用支援知識の一部として追加する．
  - AIR 化されたノウハウは，自身の利用支援知識を基に製品マニュアルに関する情報を記述した RDF から製品名とバージョンが合致するものを探索し，自身とリンクを形成する．

4.4 検索フェーズ

検索フェーズでは，ユーザからの初期クエリに適合した各 AIR（ノウハウ）が，自身には不足している知識を得るために，他の AIR に対する検索を行う．これを繰り返すことで，初期クエリを完全に満たす（初期クエリに含まれる製品を全て含む）AIR の組み合わせを決定している．また，製品バージョンの整合性の検証も，その際に同時に行っている．

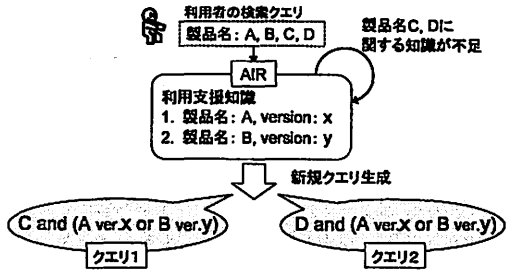

図 5　AIR による新規クエリの生成

（1）　初期クエリの入力

ユーザはユーザインタフェースを介し初期クエリを入力する．初期クエリは，システムを構築するために連携させる製品の製品名，OS，結果をソートするためのキーで構成される．入力された初期クエリは，外部インタフェースモジュールを通じ，AIR リポジトリ内の全 AIR に送信される．

（2）　初期クエリに適合する AIR の決定

初期クエリを受信した各 AIR は，自身の利用支援知識と初期クエリを比較し，以下の条件が満たされる場合，自身が初期クエリに適合すると判断する．

- 利用支援知識に記述されている 2 つの製品名が共に初期クエリに含まれる
- 利用支援知識に記述されている OS が初期クエリ中の OS と一致する

初期クエリに適合すると判断された AIR は，ワークプレース内へインスタンシエートされる．初期クエリに 3 つ以上の製品名が含まれる場合，1 個の AIR が持つ製品の知識だけでは，ユーザの要求を満たすことができない．そこで，ワークプレース内の各 AIR は，ワークプレース内の他の AIR へ新たなクエリを送信することで，自身に不足している知識の獲得を繰り返し，初期クエリを完全に満たす AIR の組み合わせを能動的に決定する．

（3）　AIR による組み合わせの能動的な決定

ワークプレース内の各 AIR は順に以下の処理を行う．

- AIR は，自身の利用支援知識，および，初期クエリに含まれる製品名に基づき新たなクエリを生成し，ワークプレース内の他の AIR へ送信する．生成されるクエリには，獲得済みの知識に関する情報 **ak-list**（製品名およびバージョンのリスト）と，不足している知識に関する情報 **uk**（1 つの製品名のみ）が含まれる．複数の製品に関する知識が不足している場合，

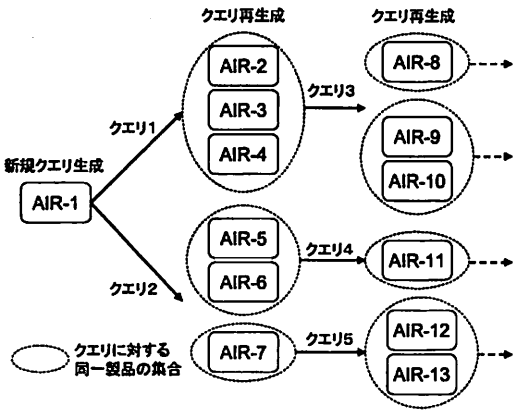

図 6　クエリ生成の動作ステップ例

各々について 1 つずつクエリが生成・送信される．

- クエリを受信した各 AIR は，自身の利用支援知識とクエリを比較し，以下の条件が満たされる場合，自身がクエリに適合すると判断する．
  - 利用支援知識に記述された製品名の 1 つのみが **ak-list** に含まれバージョンも一致する
  - 利用支援知識に記述されている他方の製品名が **uk** と一致する

  従って，例えば図 5 のように，初期クエリ（製品名 A, B, C, D）を受信した AIR（製品 A ver. x，B ver. y）は，A ver. x あるいは B ver. y 一方の知識と，C あるいは D 一方の知識を持つ AIR を探索することになる．
- クエリに適合すると判断した AIR は，自身の利用支援知識内の製品名・バージョンを **ak-list** に追加する．まだ不足する知識が残っている場合，更新された **ak-list** と **uk** を持つクエリを新たに生成し，再度ワークプレース内の他の AIR へ送信する．不足する知識が無くなるまで，この処理を繰り返すことで，初期クエリを完全に満たす AIR の組み合わせ（クエリが辿った AIR の集合）を決定する（図 6）．
- 組み合わせの重複を回避するために，初期クエリに適合し組み合わせを決定し終わった AIR は，以降，他の AIR からのクエリを破棄するものとする．

（4）　**検索結果の提示**

ワークプレース内の全 AIR が (3) の組み合わせ決定処理を完了した段階で，得られた AIR の全ての組み合わせが，外部インタフェースモジュール・ユーザインタフェースを介し，ユーザへ提示される．その際，提示される結果は，初期クエリで指定されたキーに基づきソートされる．

ここでは，2 つの製品を連携させるためのノウハウを例に提案手法について説明を行った．3 つ以上の製品を同一システムで用いる際に利用するノウハウについても本手法を拡張することで，容易に対応が可能になると考えられる．

5. ま と め

本稿では，複数の知識を組み合わせ利用する際のユーザの作業を支援するために，トピックの異なる複数の知識の検索を行い，得られた検索結果から適切な知識の組み合わせを選択しユーザに提供する手法を提案した．提案手法では，ユーザの要求に対し，各知識が不足する知識を探索することで，適切な知識の組み合わせを自動的に選択することが可能となっている．さらに，この提案手法を効果的に実現するためのシステムの提案を行った．提案システムでは，各知識を AIR として構成することにより，知識自体に能動性・自律性を与え，各処理を知識自身に能動的・自律的に代行させている．このため，複数の知識を組み合わせ利用する際に，従来ユーザに課されていた負担の大幅な軽減が期待できる．

現在，エージェントフレームワーク ADIPS/DASH [9] を用い，提案手法に基づいた試作システムの実装を進めている．この試作システムは，各知識がエージェント型の AIR として実装されたマルチエージェントシステムとして構成されてる．今後，この試作システムを用いた評価実験を行い，提案手法およびシステムの有効性を検証する予定である．

文　献

[1] "Dublin Core Metadata Initiative," http://dublincore.org/.

[2] M. Nagao et al.,"A new method of n-gram statistics for large number of n and automatic extraction of words and phrases from large text data of Japanese," In Proc. 15th Int. Conf. Computational Linguistics, pp.611–615, 1994.

[3] "Namazu: a Full-Text Search Engine," http://www.namazu.org/.

[4] 関 他, "ノウハウ蓄積システム FISH の実現とその評価", 信学論, Vol.J76-D-II, No.6, pp.1223-1231, 1993.

[5] 木下, "分散情報資源活用の一手法 — 能動的情報資源の設計—," 信学技報, AI99-54, pp.13–19, 1999.

[6] Y. Chiba et al., "A multiagent framework based architecture for academic information retrieval systems," 信学技報, AI2004-70, pp.41–46, 2005.

[7] 平 他, "AreaView2001: WWW からの構造化した領域総覧提示システム," 人工知能学会論文誌, Vol.17, No.3, pp.268–275, 2002.

[8] 大澤 他,"KeyGraph: 語の共起グラフの分割・統合によるキーワード抽出," 信学論, Vol.J82-D-1, No.2, pp.391–400, 1999.

[9] "DASH — Distributed Agent System based on Hybrid architecture —," http://www.agent-town.com/dash/index.html.