

Web 知識共有エージェントの実現

山岡 孝行[†] 辻野 克彦[†]

三菱電機株式会社 先端技術総合研究所[†]

1. はじめに

従来のナレッジマネジメントシステムでは、文章による知識の登録やキーワード入力による知識の検索など、知識の共有・再利用のために利用者が通常業務以外の作業を行う必要があり、このような負担がシステムの利用を妨げる一因になっている。筆者らは、この問題に対して、利用者に負担をかけることなく知識の共有・再利用を支援する Web 知識共有エージェントシステムの構成法を提案している [1]。本稿では、この構成法に基づき、Web を利用して行う業務を支援するためのシステムをネットワーク上に実現する方法について述べる。

2. Web 知識共有エージェント

Web 知識共有エージェントは、Web を利用して業務を遂行している利用者の操作過程を監視し、その中から共有すべき知識を抽出・蓄積する機能と、知識を再利用して現在の業務に関連する情報を提示する機能を備える。本エージェントの基本構成を図に示す。本エージェントは、4つの基本機能モジュールで構成される。

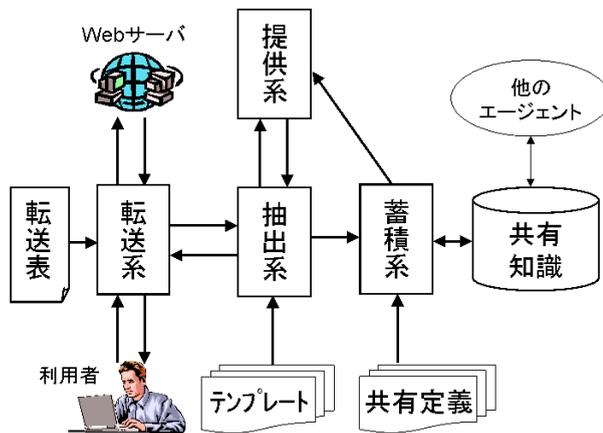


図1：システムの基本構成

- **転送系**： 利用者の操作過程を傍受し、抽出系に操作内容を転送するモジュール。
- **抽出系**： テンプレートに規定された規則に従い、操作過程からの知識抽出、および情報提供の呼び出しを行うモジュール。

- **蓄積系**： エージェント間の共有知識を蓄積、検索するモジュール。
- **提供系**： 共有知識を再利用して、利用状況に適応した情報を生成するモジュール。
本エージェントの動作を規定するために、システム運用者は予め以下の情報を設定する；
- **テンプレート**： 共有知識の抽出法と情報提供のタイミングおよび提供内容の生成方法を規定する。抽出系が参照する。
- **共有定義**： 共有知識の内容を規定する。蓄積系が参照する。
- **転送表**： 要求 URL 別に利用する抽出系を指定する。転送系が参照する。

3. Web アプリケーションによる実現

3.1. 機能モジュールの実装

4つの基本機能モジュールのうち、転送系はWeb プロキシサーバ[2]に抽出系への操作内容転送機能と転送表による制御機能とを追加することにより実装している。したがって、Web ブラウザに標準で備わっているプロキシ設定を行うだけで本エージェントの機能を利用することができる。

他の3つのモジュールは、それぞれWeb アプリケーションサーバ上で動作するサーブレットとして実装している。抽出系から蓄積系に共有知識の蓄積要求を行うときなど各モジュールの連携は、サーブレットチェーンによって実現できる。また共有知識データベースは、独立したデータベースサーバを利用する。

3.2. エージェントの動作設定

テンプレート

本実装でのテンプレート例を図2に示した。

Variable 要素では、テンプレート処理中に一時的に値を保持しておく変数名を宣言する。Operation 要素には、url 属性に指定された URL へのアクセスがあったときに処理する内容を記述する。Bind 属性が first の時は最初のアクセス時に処理を実行し、last の時は最後のアクセス時の内容で処理を実行する。Operation 要素の子要素には処理ロジックを記述する。現在の実装では、4つの処理ロジックを記述できる。Request 処理は、対象 URL 中の tag 属性で指定されたパラメータの値 (__URL__ の時は対象 URL 自体) を variable 属性に指定された一時変数に保持する。

“An Implementation of Agents for Sharing Knowledge on Web”

[†]Takayuki Yamaoka and Katsuhiko Tsujino, Advanced Technology R&D Center, Mitsubishi Electric Corporation

Response 処理は、応答ページ内の path 属性 (Xpath で記述) で参照される内容を variable 属性に指定された一時変数に保持する。Call 処理は、argument 要素で指定された一時変数の値を引数として、url 属性で指定された提供系サブレットを呼び出す。Store 処理は、argument 要素で指定された一時変数の値を引数として、url 属性で指定された蓄積系サブレットを呼び出す。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<template name="diagnosis">
  <variables>
    <variable name="device" />
    <variable name="trouble" />
    <variable name="manurl" />
    <variable name="result" />
  </variables>
  <operation url="http://server1/diagnosis/result.do*" bind="last"> .....
    <response variable="device" path="/html/body/*table/tr[2]/td[1]" />
    <response variable="trouble" path="/html/body/*table/tr[5]/td[1]" />
    <call url="http://agent2/generator/servlet/call">
      <argument tag="device" variable="device" />
      <argument tag="trouble" variable="trouble" />
    </call>
  </operation>
  <operation url="http://server2/manual/man-.*\.html" bind="last"> .....
    <request variable="manurl" tag="__URL_" />
  </operation>
  <operation url="http://server1/report\jsp?.*(result=+).*" bind="first"> .
    <request variable="result" tag="result" />
    <store url="http://agent3/accumulator/store" key="trouble">
      <argument tag="device" variable="device" />
      <argument tag="trouble" variable="trouble" />
      <argument tag="manurl" variable="manurl" />
      <argument tag="result" variable="result" />
    </store>
  </operation>
</template>
```

図 2 : テンプレート

共有定義

共有定義では、共有知識データベースのスキーマ定義および関連情報の生成方法を指定する。

```
<schema-def name="diagnosis">
  <data-schema name="diagnosis">
    <data-column name="device" type="String" />
    <data-column name="trouble" type="String" />
    <data-column name="manurl" type="String" />
    <data-column name="result" type="String" />
  </data-schema>
  <relation xsi="http://agent2/generator/xsl/diagnosis.xsl"
    classname="agent.DiagnosisResult">
    <input name="device" />
    <input name="trouble" />
    <output name="manurl" />
    <output name="result" />
  </relation>
</schema-def>
```

図 3 : 共有定義

蓄積系は、共有定義が定義されるとデータベース中に data-column 要素で指定された項目を持つ schema-def 要素名のテーブルを作成する。一方、提供系からの知識の検索要求に対しては relation 要素で指定された情報を返す。class 属性は提供系が関連情報生成のために使用するクラスを指定する。xsi 属性は、関連情報をブラウザ

に表示するときのスタイルファイルを指定する。input, output 要素は、検索時入力項目と出力項目の指定である。

4. 適用事例

図 2、3 は、機器の保守作業において故障診断結果に対する対処方法をマニュアルから探すときの知識共有を行う例を記述したものである。この例に従い、本システムの動作例を説明する。

Web 上の診断アプリケーションにより故障診断結果を分析した操作に照合。応答ページの中から対象機種と故障原因を抽出し、一時変数の値として保持。その 2 つの値を引数としてサーバ agent2 上の提供系を呼び出す。この提供系は既存共有知識を検索し、図 3 の共有定義に従い、関連マニュアルの URL と報告事例を関連情報として利用者のブラウザに表示する。利用者は、マニュアルを検索することなく、過去に参照されたマニュアルページを見ることができる。

マニュアル閲覧操作に照合し、報告書が完成する直前に参照されていたマニュアルページの URL を一時変数に保持。

報告書アプリケーションの完了操作に照合。入力された結論を一時変数 result に保持。agent3 上の蓄積系を呼び出し、一時変数の値を共有知識としてデータベースに蓄積。

5. まとめ

利用者に負担をかけることなく知識の共有・再利用を実現するエージェントシステムの実現方法を提案した。システムは、以下の特徴を持つ。

- 利用者は、Web を通常と同じ方法で利用することにより、知識の共有・再利用が可能。
- 既存の Web コンテンツやに変更を加えることなく、知識の共有・再利用機能を追加することが可能。

今後の課題には、システム導入・運用の負担を軽減するためにエージェントの動作設定を簡便化する GUI ツールを開発することがある。

参考文献

[1] 辻野, 山岡: 知識共有 Web プロキシエージェント, Joint Agent Workshops & Symposium 2003(JAWS2003), pp.291-297 (2003)

[2] 小俣: C 言語による TCP/IP ネットワークプログラミング, ピアソン・エデュケーション (2001)