

Semantic Web に基づいた対話エージェント

木村美香子^{†a)}

北村 泰彦^{†b)}

YuriTijerino^{††c)}

高野 敦子^{†††d)}

Conversational Agent Based on Semantic Web

Mikako KIMURA^{†a)}, Yasuhiko KITAMURA^{†b)}, Yuri TIJERINO^{††c)}, and Atsuko TAKANO^{†††d)}

Abstract.

Web ページを自然言語を用いて対話的に説明するエージェントは利用者に対してより積極的に情報発信することが可能である。これまでもいくつかのエージェントが研究されてきたが、Web ページの内容にエージェントの発話内容を変化させることは困難であった。本研究では、ユーザの Web ページに関する質問に対して、Semantic Web 技術を用いてエージェントの返答を自動生成する手法を提案する。Web ページ情報は予め RDF で記述されているという前提のもとで、質問文中のキーワードと RDF グラフを用いて、ユーザの求めている情報を同定し、その内容を返答する。プロトタイプとして、研究室ホームページ案内エージェントを開発した。

Keywords. Human Agent Interaction, Semantic Web, RDF, SPARQL, Embodied Conversational Agents

1. はじめに

キャラクタエージェントとはコンピュータの画面上で人間のように振舞うエージェントのことであり、例として図 1 に示す Microsoft Agent [1] がある。キャラクタエージェントにより、人間にとってより親しみやすいユーザインタフェースを提供できる。その一例として、キャラクタエージェントを利用した Web ページ案内システムの開発が行われている [2]。エージェントはユーザと対話しながら、ユーザの望む情報を同定し、それに応じたページを表示して、その内容を説明する。

従来のキャラクタエージェントシステムでは、システム開発者がエージェントの行動をプログラムとして



図 1 キャラクタエージェント (Microsoft Agent)

直接書くことで制御していた。従って、Web ページの内容を更新した場合、その変更がプログラムには反映されないため、システム開発者がその都度プログラムを変更する必要があった。

そこで、本研究ではこの問題を解決するために、Semantic Web 技術を用いた返答自動生成方法について研究する。まず、RDF 記述を利用して、Web ページの内容をエージェントが理解可能にする。ユーザの質問文中のキーワードを形態素解析によって抜き出し、RDF グラフとマッチングすることで、ユーザの必要としている情報を同定し、その返答を自動生成する。これにより、Web ページの内容が更新されても、それに応じたエージェントの返答が自動生成されるため、システム開発者は手作業でエージェントのプログラムをその都度更新する必要がなくなる。本論文では以下、2 章において本研究で用いている Semantic Web 技術について述べる。次に 3 章でユーザの質問文と RDF

[†] 関西学院大学理工学部 〒 669-1337 兵庫県三田市学園 2-1
School of Science and Technology, Kwansai Gakuin University 2-1 Gakuen, Sanda-shi, Hyogo, 669-1337 JAPAN

^{††} 関西学院大学総合政策学部 〒 669-1337 兵庫県三田市学園 2-1
School of Policy Studies, Kwansai Gakuin University 2-1 Gakuen, Sanda-shi, Hyogo, 669-1337 JAPAN

^{†††} 兵庫大学 〒 669-1337 兵庫県加古川市平岡町新在家 2301
Hyogo University 2301 Shinzaike, Hiraokacho, Kakogawa-shi, Hyogo, 669-1337 JAPAN

a) E-mail: mikako.kimura@ksc.kwansei.ac.jp

b) E-mail: ykitamura@ksc.kwansei.ac.jp

c) E-mail: yuri@tijerino.net

d) E-mail: takano@humans-kc.hyogo-dai.ac.jp

グラフからユーザの望む情報を同定し、それを返答する仕組みについて説明する。4章ではプロトタイプとして開発された研究室ホームページ案内エージェントについて述べる。5章で考察を述べ、6章で本論文のまとめとする。

2. Semantic Web

2.1 RDF データの生成

Semantic Web とは、メタデータ(データについての情報を記述したデータ)を Web ページに持たせることでエージェントがその情報を理解し、自動的に処理できるようにする枠組みである[3]。この枠組みは、Web の創設者であるの Tim Berners-Lee によって提唱された次世代 Web モデルである。エージェントが Web ページの内容を説明するには、まずエージェントが Web ページの内容を理解できる必要がある。そのため、RDF を使って Web ページの内容をエージェントが理解可能な形式に記述する。RDF とは Resource Description Framework の略で、情報についての情報(メタデータ)の表現方法についての枠組みである[3]。RDF 記述によって、図2のように Web ページの内容を有向グラフとして表すことが出来る。このグラフを RDF グラフという。

このグラフは、リソース、クラス、プロパティ、リテラルの4つから構成される。図2では、空白の円がリソース、矢印がプロパティ、四角がリテラル、そしてプロパティrdf:typeの先にある円がクラスを表している。リソースとは、RDF で記述される対象であり、URI で指定される。クラスとは、リソースのタイプもしくはカテゴリを表す。プロパティとは、リソースを記述するために用いる属性である。リテラルとは、プロパティの内容を表す記述である。例として、図2の RDF グラフには「木村」という人物のリソースが存在している。このリソースのクラスは”foaf:Person”であり、これは人物を示すクラスである。また、このリソースの姓を表す属性は、”foaf:family_name”というプロパティで表される。このプロパティの指す内容、つまりリテラルの値が姓である「木村」となっている。

図2のグラフを RDF データで表したものが図3である。この例では、FOAF[4]と Dublin Core(DC)の2つのオントロジが使われている。FOAF(Friend Of A Friend)は、人々とそのつながりを記述するオントロジで、foaf:Person クラスを使うことで、名前、性別、その人のプロジェクトなどの人に関する情報を記

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<rdf:RDF
.....
<foaf:Person rdf:nodeID="me">
  <foaf:family_name>木村</foaf:family_name>
  <foaf:currentProject>
    <foaf:Project>
      <dc:subject>Semantic Web</dc:subject>
      <dc:title>Semantic Web に基づくエージェン
    </dc:title>
    </foaf:Project>
  </foaf:currentProject>
</foaf:Person>
.....
</rdf:RDF>
```

図3 RDF データの例

述することができる。また、Web 上のリソースについて記述するために Dublin Core を使う。Dublin Core とは、メタデータを記述する標準仕様であり、Web や文書の作者、タイトル、作成日といった書誌情報を表すための共通語彙として用いられる。これによって、人物に関連している実体の情報について記述することができる。

2.2 SPARQL による情報抽出

RDF データを取り扱うことができる JAVA のクラスライブラリである JENA [6] には、RDF クエリ言語である SPARQL(Simple Protocol And RDF Query Language) [7] を使った検索機能があり、それを用いて検索条件を記述することにより、RDF データから必要な情報を抽出することができる。

図4は、「Semantic Web」に関する研究を現在している人の姓を抽出するクエリである。クエリ中の?x,?y,?z は変数を表している。このクエリを図3の RDF データに対して実行すると「木村」という結果が得られる。

3. RDF グラフからの返答生成

本研究では、Web ページを表す RDF グラフから質問サブグラフを抽出し、ユーザの質問を同定する。

図2の RDF グラフと「Semantic Web の研究をしている人の名前は何か?」というユーザの質問文から、図5のような質問サブグラフを得ることが出来る。質問サブグラフとは、RDF グラフからユーザの質問にマッチする部分を抜き出したものである。質問

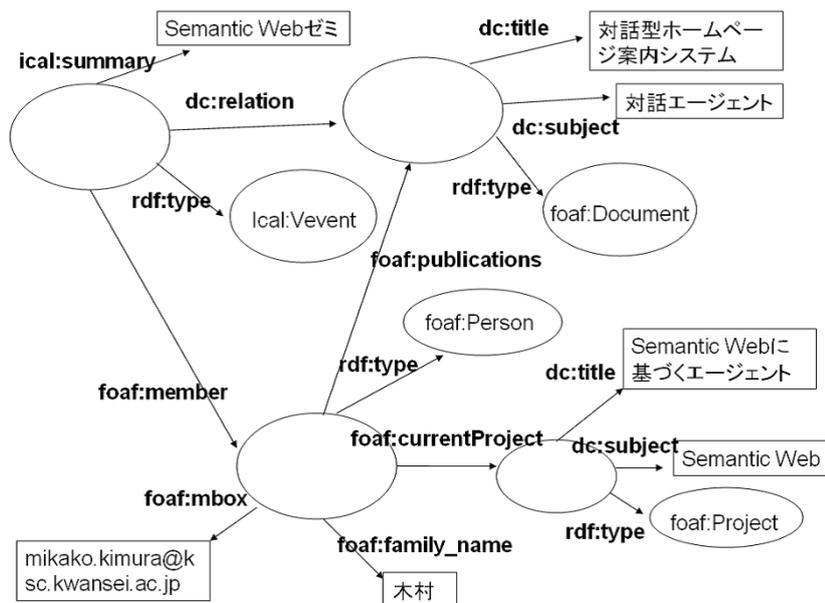


図 2 メタデータを表す RDF グラフ

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE
{
  ?x ?z "Semantic Web".
  ?x rdf:type foaf:Project.
  ?y foaf:currentProject ?x.
  ?y rdf:type foaf:Person.
  ?y foaf:family_name ?name .
}
    
```

図 4 SPARQL クエリの例

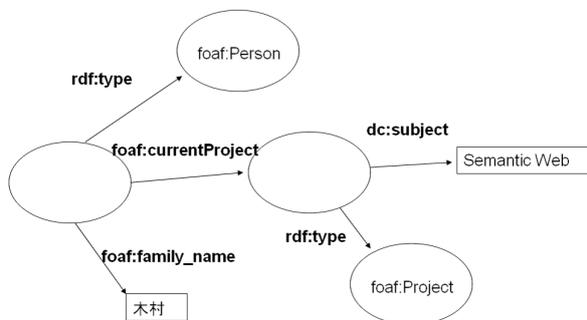


図 5 メタデータのグラフから抜き出された質問サブグラフの例

サブグラフは、質問文のキーワードとマッチするようなクラス、プロパティ、リテラルを最も多く含む最も小さいサブグラフであり、ユーザの質問中のキーワード、「Semantic Web」、「研究」、「人」、「名前」というキーワードが含まれている。

次に、質問サブグラフから、エージェントの返答に用いるリテラルの値を見つけて、それをを用いて最終的にエージェントの返答生成を行う。この例では、「木村さんが Semantic Web の研究をしています」となる。以上の方法によって、エージェントが自動的にユーザの質問文とマッチする質問サブグラフを求め、そこからユーザの要求する情報を検索し、返答生成を行うこ

とができる。以下、その詳細について述べる。

3.1 質問サブグラフの抽出

ユーザの質問を同定するために、RDF グラフから質問サブグラフを抽出する。まず、形態素解析システム sen [8] を用いて、質問文のキーワード抽出を行う。キーワードとは質問文中の名詞である。

次に、名詞と判断されたキーワードに該当する、クラス、プロパティ、リテラルを探し出す。本手法では、キーワードとクラス、プロパティの関係を表すクラス・プロパティ辞書を予め作成しておく。クラス・プロパティ辞書の例を表 1 に示す。

この辞書には、クラス、プロパティの名称、基本キーワード、補足キーワードが記述されている。ユーザの質問文中に含まれるキーワードとマッチするクラス、プロパティを求めるには、質問文のキーワードを表中の基本キーワードから探し、それに対応するクラス、プロパティを求めていくことで得られる。例えば、「Semantic Web の研究をしている人の名前は何ですか?」という質問文を形態素解析すると、「Semantic Web」、「研究」、「人」、「名前」というキーワードが得られる。これらのキーワードと、クラス・プロパティ辞書を見ると、「研究」は”foaf:Project”、「人」は”foaf:Person”、「名前」は”foaf:family_name”に関連していることがわかる。また「Semantic Web」は、RDF グラフからリテラル値と判断する。よって、これらが質問サブグラフに含まれるクラス、プロパティ、リテラルとして求められる。

また、補足キーワードは、基本キーワードだけではクラス、プロパティが特定できない場合に用いられる。例えば、質問文中に「現在の研究」という文節が含まれていた場合、キーワードとして「現在」、「研究」が得られる。表 1 から「研究」を表すクラス・プロパティとしては、「foaf:Project」、「foaf:currentProject」、「foaf:pastProject」の 3 つが挙げられる。「foaf:currentProject」の補足キーワードに「現在」が含まれるので、「現在」「研究」の 2 つのキーワードからは、「現在の研究」を表す”foaf:currentProject”プロパティが質問サブグラフに含まれることになる。

一方で、補足キーワードはそれ単独ではキーワードとはマッチせず、基本キーワードと組のときにマッチする。したがって補足キーワード「現在」は文節「現在の住所」とはマッチしない。

以上の手順で得られるクラス、プロパティ、リテラルをできるだけ多く含み、かつ最も小さなサブグラフを RDF グラフから抽出する。以上の手順によって得られたグラフを、質問サブグラフとする。

3.2 質問サブグラフからの返答生成

求めた質問サブグラフから、エージェントの返答生成に必要なリテラルの値を見つける。まず、質問サブグラフに存在するリテラルの値と、それぞれのリソースの代表値をエージェントの返答生成に用いる。質問サブグラフに存在するリソースの代表値を求めるには、デフォルト辞書を用いる。デフォルト辞書の例を表 2 に示す。この辞書には、リソースのクラスごとに、代表値となるプロパティが記述されている。例えば、人

表 1 クラス・プロパティ辞書の例

名称	基本キーワード	補足キーワード
dc:title	タイトル	
dc:creator	著者	
dc:description	内容	
foaf:family_name	名前 名字	
foaf:gender	性別	
foaf:mbox	メール アドレス	
foaf:Person	人 人物	
foaf:Project	プロジェクト 研究	
foaf:Document	本 文書	
foaf:currentProject	プロジェクト 研究	今 現在
foaf:pastProject	プロジェクト 研究	昔 過去

表 2 デフォルト辞書の例

クラス	デフォルトプロパティ
foaf:Person	foaf:family_name
foaf:Project	dc:title
foaf:Document	dc:title
ical:Vevent	ical:summary

を表すリソースのクラスは”foaf:Person”で、デフォルト辞書からその代表値は姓を表す”foaf:family_name”である。以下の三つの場合における返答について、返答生成の方法を述べる。

(1)「木村さんのメールアドレスは何ですか?」

図 2 の RDF グラフとこの質問文から、質問サブグラフを求める。この質問文を形態素解析して得られるキーワードは、「木村」と「メールアドレス」である。これらのキーワードと表 1 から「メールアドレス」を表すプロパティが”foaf:mbox”であり、RDF グラフから「木村」はプロパティ family_name のリテラル値であることがわかる。そして、RDF グラフから、プロパティ”foaf:mbox”、リテラル「木村」を含み、なおかつ最小の質問サブグラフを求めると、図 6 が得られる。次に、質問サブグラフに存在しているリソースの代表値を求める。図 6 の質問サブグラフには「木村」という人物を表すリソースがあり、クラスは”foaf:Person”である。表 2 から、このリソースのクラスである”foaf:Person”と対になるデフォルトプロパティは、”foaf:family_name”であることがわかる。よって、このリソースの代表値は、このリソースの”foaf:family_name”である「木村」であることがわかる。この例では、質問サブグラフに存在しているリテラルの値である「木村」と、求めた代表値は重複する。これらのリテラルの値と代表値を使うことで、「木村さんのメールアドレスは、mikako.kimura@ksc.kwansei.ac.jp です」というエージェントの返答を生成する。

(2)「Semantic Web に関するどんなプロジェクト

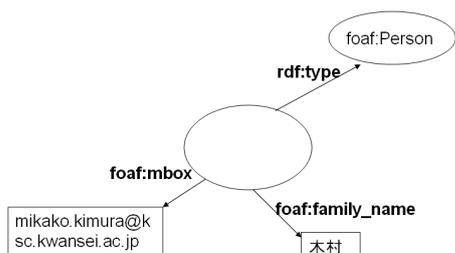


図 6 例(1)で得られる質問サブグラフ

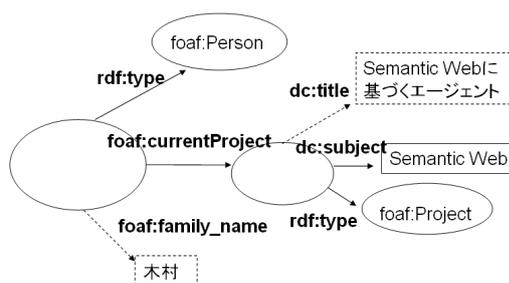


図 8 例(3)で得られる質問サブグラフ

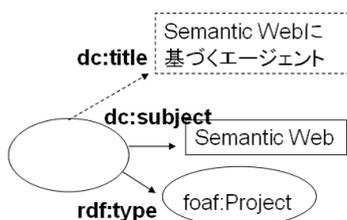


図 7 例(2)で得られる質問サブグラフ

トがありますか？」

この質問文を形態素解析して得られるキーワードは「Semantic Web」、「プロジェクト」である。これらのキーワードと表 1 から「プロジェクト」を表すクラスが「foaf:Project」であり「Semantic Web」はプロパティ「dc:subject」のリテラル値であることがわかる。そして、RDF グラフから、クラス「foaf:Project」、リテラル「Semantic Web」を含み、なおかつ最小の質問サブグラフを求めると、図 7 の実線部分のグラフが得られる。次に、質問サブグラフに存在しているリソースの代表値を求める。図 7 の質問サブグラフには、「Semantic Web」の研究に関するリソースがあり、クラスは「foaf:Project」である。表 2 から、このリソースのクラスである「foaf:Project」のデフォルトプロパティは、「dc:title」であることがわかる。よって、このリソースの代表値は、このリソースの「dc:title」を指す「Semantic Web に基づくエージェント」であることがわかり、これは図 7 の点線部分を指す。この代表値と、質問サブグラフに存在するリテラルの値である「Semantic Web」を用いることで「Semantic Web のプロジェクトとして、「Semantic Web に基づくエージェント」があります。」というエージェントの返答生成をする。

(3)「Semantic Web の研究をしている人は誰ですか？」

この質問文を形態素解析して得られるキーワードは「Semantic Web」、「研究」、「人」である。これらのキーワードと表 1 から「研究」を表すクラスが「foaf:Project」、「人」を表すクラスは「foaf:Person」であり、「Semantic Web」はプロパティ「dc:subject」のリテラル値であることがわかる。そして、RDF グラフから、クラス「foaf:Project」、「foaf:Person」、リテラル「Semantic Web」を含み、なおかつ最小の質問サブグラフを求めると、図 8 が得られる。図 8 の質問サブグラフには、「Semantic Web」の研究に関するリソースと、その研究をしている人物のリソースがあり、クラスはそれぞれ「foaf:Project」、「foaf:Person」である。表 2 から、クラス「foaf:Project」のデフォルトプロパティは「dc:title」、クラス「foaf:Person」のデフォルトプロパティは「foaf:family_name」である。よって、研究リソースの代表値は「dc:title」の指す「Semantic Web に基づくエージェント」であり、人物リソースの代表値は、「foaf:family_name」の指す「木村」である。これらの代表値と、質問サブグラフに存在するリテラルの値である「Semantic Web」を用いることで、「Semantic Web の研究をしているのは木村さんで、「Semantic Web に基づくエージェント」の研究をしています。」というエージェントの返答生成をする。

4. 対話型ホームページ案内システム

本研究では、ユーザの質問文と Web ページのメタデータを用いて、ユーザの求める情報を同定することでエージェントの返答生成を行った。その結果、Semantic Web に基づいた対話エージェントのプロトタイプとして、研究室の案内システムを開発した。このシステムでは、研究室の Web ページに興味を持っているユーザに対して、エージェントがその案内をしていく。この Web ページの内容は RDF データに記述

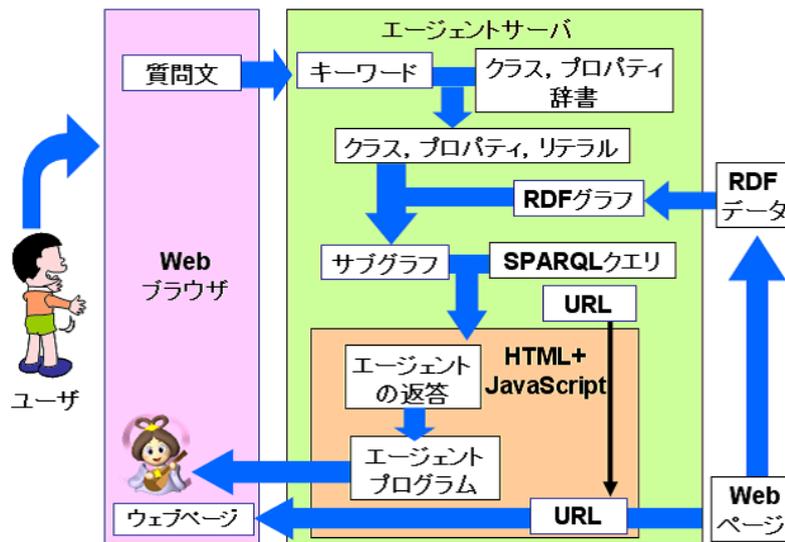


図 9 システム構成図



図 10 対話型ホームページ案内システム



図 11 エージェントの返答

されており、Web ページが更新されると、RDF データも自動的に更新されることを前提とする。システム構成図を図 9 に示す。

まず、図 10 のように、ユーザが Web ブラウザ上のダイアログボックスに質問文を入力することで、エージェントに質問する。エージェントが生成した返答はシナリオプログラムに挿入され、それを Web ブラウザ上で実行される。その結果、図 11 のように、エージェントはユーザの質問に対して返答する。

5. 今後の課題

提案方法では形態素解析によって抜き出された名詞

のみで質問サブグラフの抽出を行っている。そのため、「北村研究室で Semantic Web の研究をしている人は誰ですか?」と「北村研究室で Semantic Web の研究をしていない人は誰ですか?」という 2 つの質問文では同じサブグラフが得られてしまう。このため、後者の質問文にあるような「していない」という否定の言葉をどのように扱うかという点が今後の課題として残されている。

また、「22 歳以上のメンバーは誰ですか?」といった数の大小の扱い、「22 歳のメンバーは何人ですか?」といった数え上げの扱いも今後の課題として残されている。

6. ま と め

本研究では、ユーザの質問文と RDF グラフを用い

て、質問応答を自動的に行う対話エージェントの手法について述べた。まず、Web ページを表す RDF グラフから、ユーザの質問文とマッチする質問サブグラフを抜き出した。そして、質問サブグラフからリテラルの値を抜き出すことで、エージェントの返答生成を行った。これにより、Web ページの内容を変更してもそれに対応したエージェントの返答生成ができる。

文 献

- [1] Microsoft Corporation . Microsoft Agent Software Development Kit , Microsoft Press , 1999 .
- [2] Thomas Rist , et al . A Review of the Development of Embodied Presentation Agents and Their Application Fields , Life-Like Characters: Tools , Affective Functions , and Applications , Springer-Verlag , pp.377.404 , 2004 .
- [3] 神崎 正英 . セマンティック・ウェブのための RDF/OWL 入門 , 森北出版株式会社 , 2005 .
- [4] D. Brickley and L. Miller, FOAF Vocabulary Specification, 2005, [http:// xmlns.com/foaf/0.1/](http://xmlns.com/foaf/0.1/).
- [5] Dublin Core Metadata Element Set, Version 1.1: Reference Description. <http://dublincore.org/documents/dces/>
- [6] 情報処理相互運用技術協会編:セマンティック Web 入門 . オーム社 , 2004 .
- [7] <http://www-06.ibm.com/jp/developerworks/java/050616/j-j-sparql.html>
- [8] <http://ultimania.org/sen/>