Extending knowledge of AIML by using RDF

Mon-Tin Tzou[a], Chun-Hung Lu[a], Chin-Chien Wang[a], Cheng-Wei Lee[b], Wen-Lian Hsu[a]
[a]Institute of Information Science, Academia Sinica, Taiwan, R.O.C
[b]Department of Computer Science, National Tsing-Hua University, Taiwan, R.O.C
{mttzou,enrico,vincent,aska,hsu}@iis.sinica.edu.tw

ABSTRACT

Although Artificial Intelligence Markup Language (AIML) is a popular chat robot and is used in many applications, it has some drawbacks that need to be improved. The general AIML robots store all knowledge in AIML files and do not have the capability of term expansion. We propose an extended AIML specification that resolves the problem. To help the AIML robot obtain knowledge from the RDF ontology, we add a new element, <reference>. Knowledge on the web can be referenced by using the new element. The knowledge can be data on the web or an instance from term expansion. The enhancement raises the scalability of patterns and requires less effort in knowledge construction.

Keywords

AIML, RDF, Semantic Web, Dialogue

INTRODUCTION

Dialogue systems are becoming increasingly popular and are now used in various domains. They can be classified into many levels according to their complexity [1]. Form-based dialogue system which lead learner to fill all slots aims to finish an objective discourse. The dialogue system with the lowest complexity is the chatting robot, such as AIML.

Artificial Intelligence Markup Language (AIML), the most famous chatting robot, is an XML-compliant language that defines the stimulus-response message of the interaction between a user and an AIML agent [2]. We propose an approach that extends the AIML specification and references knowledge from the Semantic Web.

Semantic Web Activity [3] is based on RDF (Resource Description Framework) and OWL (Web Ontology Language). RDF is designed to be understandable by computers [4].

The remainder of this paper is organized as follows. In Section 1, we consider the shortcomings of AIML. In Section 2, we propose an approach to enhance AIML. The results of the enhancement are discussed in Section 3. Then, in Section 4, we present some concluding remarks and indicate the direction of our future work.

1. REVIEW OF AIML

AIML is used in many applications. According to the purpose of the dialogue system, AIML is usually as a chat robot [2]. Rollo Carpenter, the creator of Jabberwacky, won the Loebner Prize in 2005 for the most human-like computer [5]. AIML can be used in tutoring systems and an authoring tool for learning design on e-learning [6][7]. Some researches adapt machine learning methods to extract knowledge from online forums. Those knowledge is translated into AIML format in chat robot [8]. However, the knowledge gathered from forums is not actually stored outside of the AIML files.

The AIML robot responds according to the user’s input and the knowledge stored in the AIML files. To be a reasonable interaction constructed by the knowledge in the AIML depends on the author’s effort in construction. The original AIML specification has both advantages and disadvantages. Some of the advantages are:

- The complexity of dialogue system is very low and easy to understand.
- The formal syntax of knowledge presentation can be read easily by computers and humans.

Some of the disadvantages are:

- The knowledge performs as an instance in AIML files.

If knowledge is acquired from data on the Web, it will not be updated automatically; thus, the maintainer needs to update it periodically. Because knowledge changes constantly, fixed knowledge is not a good strategy to update knowledge dynamically.

- The original AIML does not have term expansion capability.

It is expensive to establish knowledge, even when the knowledge is almost in the same form. Each pattern only with one word different is needed to have an independent category. This is not efficient; moreover, it is not easy to manage similar knowledge presented in different categories.

In order to solve the above drawbacks, we propose an extended method to enhance the AIML robot. To reduce the effort required to create and manage knowledge, we make the AIML specification more powerful and shorten the authoring process. All the interaction scripts are in an AIML file or an AIML set. The proposed approach extends the AIML knowledge source from a closed set to an outside RDF ontology. Term expansion and referenced knowledge are achieved by adding a new element to the AIML specification.

2. AIML ENHANCEMENT

2.1 Evaluation of enhancement

We now consider the issues involved in adding a new element to the original AIML specification.

- How to reference knowledge?

What will happen if we add a new element to the AIML specification? The AIML interpreter must have a definite manner of processing the newly added element and define the conditions of use.

- Ambiguity problem

The referenced knowledge may encounter an ambiguity problem. What if the referenced data has more than two literals that accept the reference constraint? A method of solving the ambiguity problem is thus needed.

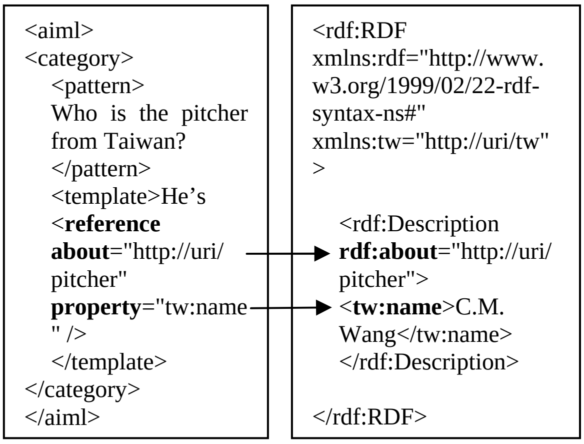
2.2 Our Approach

2.2.1 Template-side reference

The “reference” element is defined in Figure 1. There are two mandatory attributes in the “reference” element. The “about” attribute indicates which RDF resource should be referenced, while the “property” attribute indicates the property in the resource. The AIML interpreter should return the referenced literal or issue an error if the resource is not found. The ontology outside the internal AIML knowledge can be referenced by the specified “reference” element. Figure 2 shows an example of using the “reference” element to access an outside ontology, where the reference tag is replaced by the referenced knowledge. When the user asks “Who is the pitcher from Taiwan?”, the response will be “He’s C.M. Wang”.

```
<!-- Category: aiml-template-elements -->
<aiml:reference
about = uri-reference
property = name-of-property  />
```

Figure 1 Definition of Reference



```
<aiml>
<category>
   <pattern>
   Who is the pitcher
   from Taiwan?
   </pattern>
   <template>He’s
   <reference
   about="http://uri/
   pitcher"
   property="tw:name
   " />
   </template>
</category>
</aiml>
```

```
<rdf:RDF
xmlns:rdf="http://www.
w3.org/1999/02/22-rdf-
syntax-ns#"
xmlns:tw="http://uri/tw"
>

   <rdf:Description
   rdf:about="http://uri/
   pitcher">
   <tw:name>C.M.
   Wang</tw:name>
   </rdf:Description>

</rdf:RDF>
```

Figure 2 Example of Template-side Reference

2.2.2 Term Expansion

To achieve the term expansion feature, the “reference” element is defined as in Figure 3 and Figure 4. When the “reference” element is used in the pattern element, it has a mandatory attribute “about”, which indicates the URI of the RDF resource to be referenced. The AIML interpreter will fire all literals in the collection of this resource. In Figure 4, the “reference” element in the template element is replaced by the literal fired in the pattern-side referenced element. If more than two literals are fired, the AIML interpreter will adopt the first one. If no literals are fired, this pattern will not be fired.

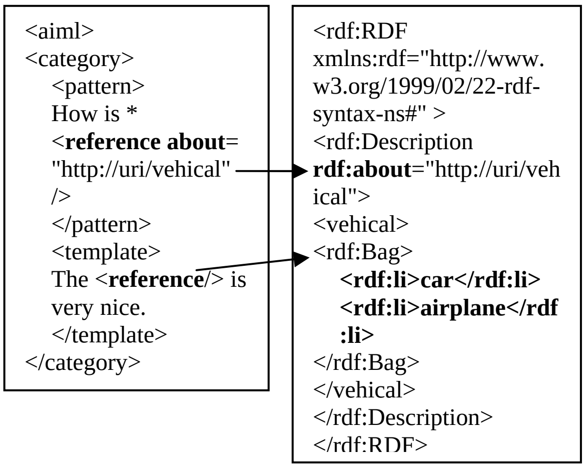
```
<!-- Category: aiml-pattern-elements -->
<aiml:reference
  about = uri-reference  />
```

Figure 3 Definition of Pattern-side Reference

```
<!-- Category: aiml-template-elements -->
<aiml:reference />
```

Figure 4 Definition of Template-side Reference

Figure 5 shows an example of this case. When the user says “How is the car”, “car” will be fired by the specified RDF resource. The AIML interpreter will replace “<reference/>” with “car” and the output will be “The car is very nice”. If there are thousand kinds of vehicles, the author needs to create thousands of category elements for each one. With this new element, all the vehicles can be listed in the RDF ontology, instead of manually editing categories with almost the same pattern.



```
<aiml>
<category>
   <pattern>
   How is *
   <reference about=
   "http://uri/vehical"
   />
   </pattern>
   <template>
   The <reference/> is
   very nice.
   </template>
</category>
```

```
<rdf:RDF
xmlns:rdf="http://www.
w3.org/1999/02/22-rdf-
syntax-ns#" >
<rdf:Description
rdf:about="http://uri/veh
ical">
<vehical>
<rdf:Bag>
   <rdf:li>car</rdf:li>
   <rdf:li>airplane</rdf
   :li>
</rdf:Bag>
</vehical>
</rdf:Description>
</rdf:RDF>
```

Figure 5 Example of Term-expansion

3. EXPERIMENTAL RESULTS AND IMPROVEMENT

3.1 Environment

The new AIML interpreter is implemented by modifying AIMLBot (Program #) with Microsoft Visual Studio 2005 in C# [9]. To evaluate the behavior of the extended AIML, we use a form-based dialogue system and an AIML robot to simulate the interaction of a travel agent with the AIML robot.

3.2 Experimental results

The form-based dialogue system we used is a ticket booking system based on a slot-filling strategy. The questions asked by the form-based dialogue system follow a script and have a strict asking sequence

[10][11]. When the system first asks a question, the AIML robot replies a message to the form-based dialogue system. Keep the chatting loop until all the slots in the form-based dialogue system is filled, as shown in Figure 6.

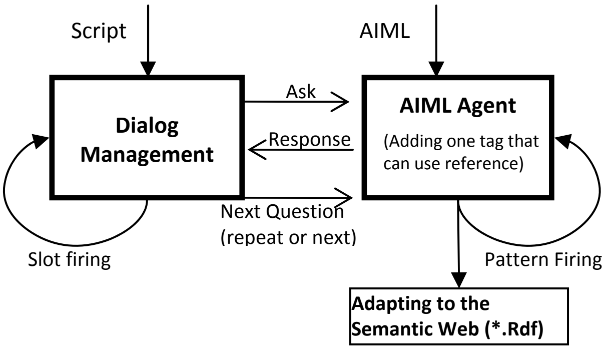

Script
AIML
Dialog Management
Ask
Response
AIML Agent
(Adding one tag that can use reference)
Next Question (repeat or next)
Slot firing
Pattern Firing
Adapting to the Semantic Web (*.Rdf)

Figure 6 Experiment Architecture

The snippet of results is shown in Figure 7. The second response from AIML is based on the reference from resource shown in Figure 8. The RDF resource "place" contains all possible places. Then, it only needs one category instance of AIML to represent all places, instead of creating a category instance for each place.

Dialog: This is the train ticket booking system. May I help you?
AIML: I need a one-way ticket
Dialog: What is the starting point?
AIML: Taipei
Dialog: What is the destination?
AIML: Kaohsiung
Dialog: What is the date and time?
AIML: 5/9 6:00 PM
Dialog: How many tickets?
AIML: One
Dialog: What is the identity? Normal, student
AIML: Normal

Figure 7 Experimental Results

```
<rdf:Description
rdf:about="http://iasl.iis.sinica.edu.tw/rdf#place">
<rdf:Bag>
            <rdf:li>Taipei</rdf:li>
            <rdf:li>Kaohsiung </rdf:li>
            <rdf:li>Tokyo</rdf:li>
            ……
</rdf:Bag>
</rdf:Description>
```

Figure 8 A Snippet of AIML Ontology

The approach can reduce the effort required to create and manage knowledge and use web ontology to keep the consistency.

4. CONCLUSION & FUTURE WORK

We propose an extended AIML specification that overcomes the drawbacks of AIML. The newly added ”reference” element allows knowledge to be stored outside of the AIML files. Then, the knowledge can be stored on the web and updated with time. The AIML robot can reference the RDF ontology and it is capable of term expansion through the "reference" element. Our contribution is that we simplify the original approach of authoring AIML knowledge so that it requires less effort. This enhancement raises the scalability of patterns and can be used in modeling students' learning behavior [12]. The AIML interpreter may also be able to understand the semantic context and respond with a higher level semantic response.

ACKNOWLEDGMENTS

We would like to thank Program # for providing tools. This research was supported in part by National Science Council under Grant NSC 95-2752-E-001-001-PAE, the thematic program of Academia Sinica under Grant AS 95ASIA02, and the Research Center for Humanities and Social Sciences, Academia Sinica.

REFERENCES

[1] James F. Allen, Donna K. Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent (2001) Towards Conversational Human-Computer Interaction, AI Magazine, 22, 4 (Winter 2001), 27-37.

[2] A.L.I.C.E. AI Foundation (2005) Artificial Intelligence Markup Language (AIML) Version 1.0.1, http://www.alicebot.org/TR/2005/WD-aiml/

[3] W3C (2007) Semantic Web, http://www.w3.org/2001/sw/

[4] W3C (2004) Resource Description Framework (RDF), http://www.w3.org/RDF/

[5] The Loebner Prize in Artificial Intelligence (2006) http://www.loebner.net/Prizef/loebner-prize.html

[6] Eliseo Reategui, Elisa Boff, and John A. Campbell (2006) Endowing a Virtual Character with Personalization Capabilities, ICTAI '06, 220-224.

[7] Dietmar Janetzko (2006) Dialogue-Based Authoring of Units of Learning, Proceedings of ICALT 2006, 436-440.

[8] Jizhou Huang, Ming Zhou, and Dan Yang (2007) Extracting Chatbot Knowledge from Online Discussion Forums, IJCAI-07.

[9] Program#, http://aimlbot.sourceforge.net/

[10] Trung H. Bui, Martin Rajman, and Miroslav Melichar (2004) Rapid Dialogue Prototyping Methodology, Lecture Notes in Computer Science, 3206/2004, 579-586.

[11] David Goddeau, Helen Meng, Joe Polifroni, Stephanie Seneff, and Senis Busayapongchai (1996) A form-based dialogue manager for spoken language applications, Proceedings of ICSLP 96, 2, 701-704.

[12] Eliseo Reategui, Elisa Boff, John A. Campbell (2006) Using virtual characters in personalized recommendations, Proceedings of the 38th SIGCSE, 180-184