Design and Implementation of Conversational Agents for Harvesting Feedback in eLearning Systems

Karsten O. Lundqvist, Guy Pursey, Shirley Williams

University of Reading, School of Systems Engineering,
Whiteknights, Reading, RG6 6AY, United Kingdom
Tel: +44 118 378 7600
Fax: +44 118 378 7600
k.o.lundqvist@reading.ac.uk

Abstract. Traditionally conversational interfaces, such as chatbots, have been created in two distinct ways. Either by using natural language parsing methods or by creating conversational trees that utilise the natural Zipf curve distribution of conversations using a tool like AIML. This work describes a hybrid method where conversational trees are developed for specific types of conversations, and then through the use of a bespoke scripting language, called OwlLang, domain knowledge is extracted from semantic web ontologies. New knowledge obtained through the conversations can also be stored in the ontologies allowing an evolving knowledge base. The paper describes two case studies where this method has been used for evaluate TEL by surveying users, firstly about the experience of using a learning management system and secondly about students' experiences of an intelligent tutor system within the I-TUTOR project.

Keywords: Conversational Interface, Chatbot, Ontology, AIML

1 Introduction

The primary aim of the research behind this paper is to develop a method for implementing conversational agents, also known as chatbots, which can change domain knowledge easily and expand the base of knowledge through conversation for specific conversational situations.

This has been achieved by developing a new scripting language, called OwlLang, which can embed knowledge from the underlying pool into automated conversations and pull knowledge from these conversations into the underlying knowledge base.

Using this language together with a traditional chatbot development language a survey chatbot was developed. Two different knowledge bases were tested. The first was a survey of students' opinion of the virtual learning environment Blackboard. The second was an online survey of a research project website. The chatbot is currently being integrated in an Intelligent Tutoring System (ITS) inside Moodle as part of the I-TUTOR project (http://www.intelligent-tutor.eu).

2 Background

A chatbot is a programme that interacts with human users through automated chat. This interaction can be many types of interaction such as logs, spam, or entertainment [1]. The work here is concentrated on conversation between a chatbot system and human users for specific situations, and therefore requires some natural language processing.

There are many approaches to solving any natural languages processing problem [2] but when working on a conversational interface there are primarily two approaches: natural language parsing and conversation prediction.

Natural language parsing follows a traditional parsing approach where the text is analysed using a parser similar to a computer language parser but modified with more lookaheads and feature enriched grammars (e.g., in tense resolution), and utilizing statistical methods based on relevant word corpora to resolve ambiguities. World knowledge is usually supplied in the form of predefined ontologies which are used to determine the proper responses according to a domain knowledge pool [3].

Human conversation, however, is not random but follows a Zipf curve distribution of possible responses to preceding utterances [4]. For instance, if person A says 'Can I try?' to person B, it is very unlikely that person B will respond '2 plus 2 is 4'. The chatbot developer, therefore, can try to cover as many user responses to each chatbot output as possible, using a descriptive chatbot language such as the Artificial Intelligence Meta Language (AIML) [5] to build a map of possible responses to human interaction, in this paper termed conversational trees. One of the earliest examples of this approach is ELIZA which utilized the basic structure of Rogerian therapy sessions to emulate a psychiatrist [6]. This is a trial and error approach where the developer constantly monitors the chat logs to see if new chat patterns have been discovered that need to be added to the conversational trees.

Natural language parsing systems can be complex to build and maintain, because the pure parsing processes involved in natural language processing are much more demanding than computer language parsing [3]. This method requires a corpus of texts to build up the statistical knowledge behind the parsers; without corpora that reflect the expected natural language utterances appropriately, the error rate of the ambiguity resolution performed by the statistical methods will increase. The resulting product will be prone to conversational errors and misunderstandings [3].

World knowledge is described within ontologies. An ontology is an intentional semantic structure which encodes the implicit rules constraining the structure of a piece of reality [7], hence when looking at a knowledge base the ontology can be thought of as a specification of what can be in the knowledge base as well as the rules that govern what is allowed in it. This is one of the main attributes of ontologically specified knowledge bases that make the conversational interface system interesting; if different knowledge bases adhere to the same ontology that the conversational interface system uses, then they can both be used by the system without any changes to the knowledge bases. This allows the system to have interchangeable parts as long as the same ontology is used. It is, therefore, fairly uncomplicated to change the world knowledge of the chatbot without having to change any of its other parts, as long as

the domain knowledge of the knowledge base follows the ontological commitment specified. OWL, the semantic web ontology language has been used in this research [8].

Conversational trees are comparatively easy to create and the results can be tested instantly with users to enhance the capabilities of the chatbot. However, the world and domain knowledge is usually incorporated into the chat trees to ease the developmental phase for non-technical developers and avoid ambiguities. The result of this is, depending on the size of the conversational tree, that it is either impossible or at least incredibly difficult to re-use the chatbot in another functionality or world domain. For example if a chatbot has been developed for an FAQ (Frequently Asked Questions) section of a specific webpage, it would be difficult to transform this chatbot into another FAQ chatbot for another site without creating errors, it would usually be just as easy to start from scratch with a new chatbot.

There have been some attempts to make the conversational tree more flexible. One way is to allow injection of knowledge from an ontology into the conversations, for instance by injecting RDF (Resource Description Framework) statements [9], i.e. interchange knowledge referenced using RDF on the web, into AIML [10] or by using ontology tools to generate knowledge that is transformed into conversational trees with rules that are used to drive chatbots instead of AIML [11]. Both of these approaches provide more flexible conversational tree structures as they allow for knowledge injection at run-time, but they also lack flexibility as they do not allow for easy knowledge generation and retention within the underlying ontologies.

3 Method and Implementation

The developed conversational interface system is based on the use of the Zipf curve distribution [4] to build chat structures and sentences without any world knowledge. World knowledge is then embedded by integrating ontological queries into the conversational trees. The ontological queries retrieve the world knowledge on demand when the responses are created. This method therefore enables knowledge changes by changing the underlying ontologies, yet is still easy to extend through AIML. Figure 1 is an actual example from the chat structure of the conversational interface, here depending on the values of subject, predicate and object, many different questions can be asked. (The term ‘predicate’ here is used in the narrow ontological sense to relate the subject to the object.) The expected responses, however, generally will not depend on the values and, where they do depend on them, the words are usually part of the response allowing the chatbot systems to infer a meaning. If no meaning can be inferred the system has a series of escape questions which will assist in finding the opinion of the learner.

Fig. 1. Chat structure with integrated ontology query.

The complete system has 100 different questions of a similar structure in its conversational tree, and all of these have appropriate response trees, but the Zipf curve has also been applied on this level as most responses to questions follow the same basic pattern. The specific responses are then interpreted by the system and put into one of ten different states with different responses and actions. The states are listed below:

- Yes
- Definite no
- No, but verification needed whether user mean
  - Definite no
  - Unknown
- Unknown
- Unsure
- Unsure, but a certain (maybe unknown) value is given
- Evidence of a valid answer
- No understandable answer given

When a response has been fully explored with follow up questions, either a new question will be asked or the evaluation will end depending on the number of questions that are required.

This functionality has been achieved by developing a bespoke ontology query language OwlLang which is used within the interpreted AIML language. Figure 2 is a graphical representation of the complete system. At the heart of the system is the OwlLang scripting language, which is used to acquire the ontological knowledge bases to retrieve the domain knowledge, or in other words to fill in the gaps of the static chat patterns. The Language has been developed in Java using the compiler tool JavaCC (http://javacc.java.net) and the ontology tool Jena (http://jena.sourceforge.net/). The language itself will be described in more detail in the following section.

The chat patterns are described using AIML and are interpreted by a standard AIML interpreter ProgramD (http://aitools.org/Downloads/#programd). This application has only been slightly modified to allow incorporation of the OwlLang scripts into AIML and call the OwlLang interpreter to perform the retrieval of information from the ontological knowledge bases. The only deviation from the standard AIML language in the static structures has been the inclusion of a grammar check performed from within OwlLang instead of using the usual singular and plural resolution that decides whether ‘is’ or ‘are’ should be inserted into the response. This is due to the

fact that the AIML tags would normally have the actual known value of the subject of the sentence when run in ProgramD, but that would not be known by ProgramD when using OwlLang as it is obtained at runtime. This grammar check was implemented using Link grammar 4.1b (http://www.link.cs.cmu.edu/link/), which is an open-source grammar checker that is often used in applications using the parsing approach of natural language processing.

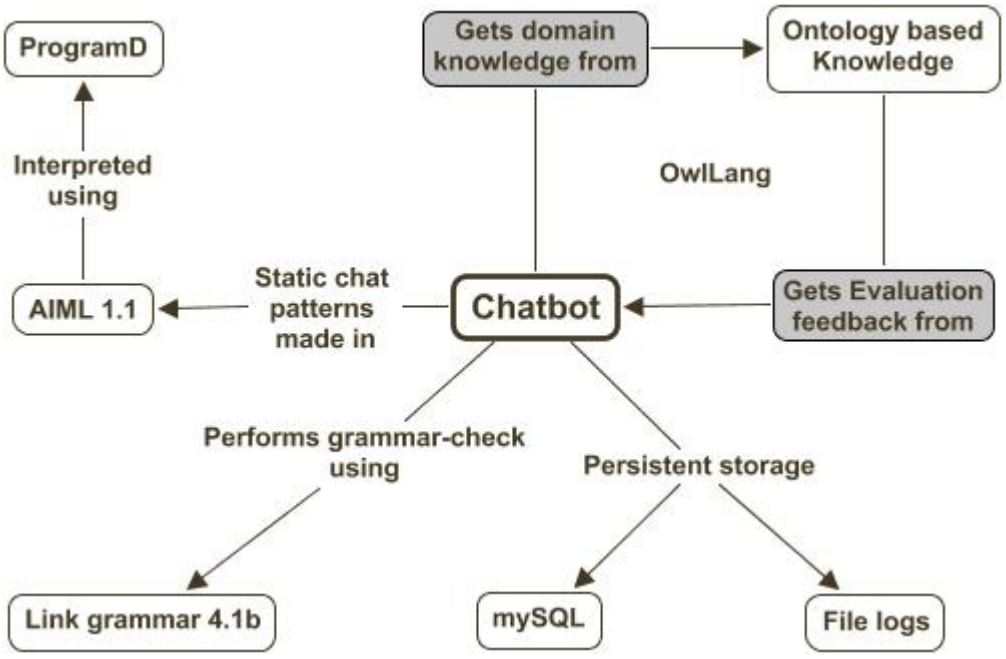

Fig. 2. The Complete Conversational System

Persistent storage of ProgramD internal data structures is done in a MySQL database (http://www.mysql.com/). Chats, ontology and the complete knowledge bases are stored in file logs, enabling easier readability for the chatbot developer; in future this should however be stored in the MySQL database for easier maintenance.

At the core of the system is OwlLang. The main functionality of OwlLang is to retrieve a string from ontological knowledge bases depending on specified operands and logics; if several strings match the specifications then a random string from that list of strings will be retrieved. The following example is not intended to be a comprehensive instruction guide to OwlLang, but rather an introduction to the capabilities of the language. The three main string retrieval commands are;

- ONTIND – gets the string from a known individual within the ontology
- ONTPROP – gets the string from a known property within the ontology
- ONTCLASS – gets the string from a known class of the ontology

The default string that will be returned is the URI (Uniform Resource Identifier) of the entity; however alternative strings can be specified following the command. There are three main commands that can be used from within the parentheses of the commands which determine what kinds of sets are used in the selection:

- ind – individuals
- prop - properties
- cls – classes

So for instance if cls is used within ONTIND the string will come from one of the individuals and the matching classes, on the other hand if ind is used within ONTCLS the string will come from one of the classes in which the matching individuals are defined.

Lately the chatbot system has been open sourced and it has been taken up and used as a feedback mechanism within an ITS being developed through the EU LLP funded project I-TUTOR. Through this project further work has been made on the core chatbot system and upgraded it to a new release of the ProgramD system, and a Moodle block plugin has been developed which creates a connection to the chatbot server.

4 Trialling the method

When considering the quality of eLearning products and procedures it is evident that evaluation is important, as the evaluation process can reify the perceived quality for the stakeholders, hence provide a basis for comparing different eLearning products and procedures [12]. Much research has been undertaken in this area in recent years such as in the Sustainable Environment for the Evaluation of Quality in E-Learning project [13] where, amongst other things, a core quality framework was created which emphasised evaluation as one of the core learning processes. The Learning Technology Dissemination Initiative consultative group has also contributed to this research by producing a guide to evaluation of learning materials in general [14], which contains a comprehensive review and appraisal of the different tools that are available in the evaluation of learning materials. Commonly used tools within e-learning are online (and sometimes even offline) Likert-style attitude surveys [15], checklists, and questionnaires, which all (because the questions rarely change) provide static predefined evaluation with no opportunity to provide feed-back mechanisms to all users. Offline face-to-face focus groups and interviews are also used, which can be more dynamic and offer opportunities for users to engage by providing value-added appraisal of the system. However, as such groups are expensive to accommodate and facilitate, they are rarely carried out on big numbers of participants. This creates a dilemma in the evaluation process between the quality of the feedback, the number of participants

(representation), and the cost of evaluation. This dilemma was the reasoning behind developing a survey chatbot system using OwlLang.

4.1 Ontologies and Knowledgebases

The system uses two different ontologies as described in figure 3. One describes quality in e-learning and one describes teaching products. By using these two ontologies the e-learning product/procedure to be evaluated is described and then several claims are made about it. These claims and predetermined 'facts' form the basis for questioning. The actual answers to these questions are added to the knowledge base as claims, which extend the pool of information that the questions are taken from.

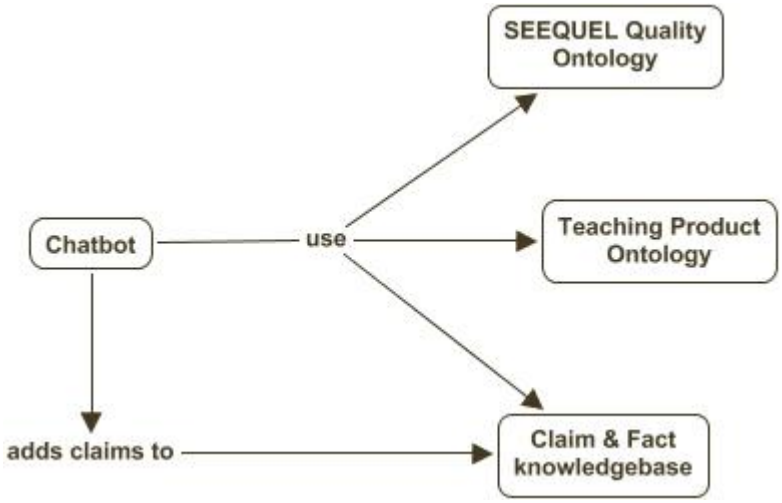

Fig. 3. Ontologies used by the System

The SEEQUEL Quality ontology is an ontology based on the internal findings of the SEEQUEL project [13]. In this project a framework for quality in eLearning was built, the ontology is based on an internal SEEQUEL ontology, which was created to better understand the framework. For this research it has been transformed using the web ontology language OWL [8] in order to allow the use of Jena to manipulate it programmatically. The ontology describes different quality descriptors and how they interrelate. Descriptors include: 'interesting', 'fun', 'boring', 'too long', 'off topic' and 'engaging'. It is important to highlight that these "loose" concepts are described using ontology tools, which are used to describe formal structures, however in this work the ontologies are describing non-formal descriptors. This approach allows the developer to use the vast amounts of Semantic Web toolsets to create knowledgebases, even in scenarios where users might be disagreeing on formal definitions and opinions. The descriptors form the basis of the chatbot's vocabulary when it needs quality terms, but if other terms are needed in other evaluation tasks the open nature

of ontologies would allow the user to add the appropriate terms to the ontology. The following code is an example of one of the interrelational facts in the ontology.

```
:boring          :impliesNot     :interesting .
:fact103         a               :Fact;
                 :madeBy         :seequel;
                 :object         :interesting;
                 :predicate      :impliesNot;
                 :subject        :boring .
```

These OWL lines describe that if something is boring then it is implied that it is not interesting. It is also reified as a ‘fact’ in the ontology made by SEEQUEL. This ‘fact’ can then be used by the chatbot to infer additional questions to ask the users. The statements from SEEQUEL are not formal ontology, although they are used within a formal ontological toolset. The statements of quality are user independent and ratings of quality can be inconsistent both between users and even for individual users. For instance a teaching module can both be boring and interesting for a single user. This tool has therefore relaxed the formality of the statements, the aim of doing this is to allow the use of the available knowledge engineering toolset found in the semantic web tool chain for situation lacking the formal definitions that are normally required in traditional ontology engineering.

The Teaching Product Ontology is a simple non-formal ontology, which enables developers to describe any teaching product enabling them to establish claims about it using the SEEQUEL Quality Ontology. This ontology is based on three core entities; teaching product, units of a teaching product and functionality of teaching products. The ontology defines the two inverse properties ‘is’ and ‘is not’ and the property ‘no relation’, which can be used to describe a defined entity using a quality descriptor from the SEEQUEL Ontology. Using these tools the chatbot developer can then define the entities which make up the product which needs to be evaluated and add the claims which form the basis of the claims knowledgebase. This knowledgebase stores all the claims and facts in the system and uses them to find the domain knowledge when the conversational interface creates another question. The following code shows one such claim, here it is claimed by a user named Karsten that the functionally called ‘discussion boards on Blackboard’ is easy to use.

```
:bbDB           :is                 seequel:easyToUse .

:claim1         a                   seequel:Claim ;
                :object             seequel:easyToUse ;
                :predicate          :is ;
                :subject      :bbDB ;
                :madeBy             seequel:Karsten .

:bbDB           a                   :DiscussionBoard ;
                rdfs:label          "discussion boards on
                                    Blackboard"@en .

:DiscussionBoard rdfs:subClassOf    :Functionality ;
                rdfs:label          "discussion board"@en
.
```

All statements made by users thoughtout the case study were added to the existing ontology and was used to generate new statements to other users.

5 Results and Discussion

The initial chatbot was seeded in this way with 57 statements, which included statements of all the functionalities which needed to be surveyed. 37 users tested the survey in a voluntary survey of students using the managed learning environment Blackboard for a module on Programming. The survey showed that generally students liked the content on Blackboard, but that the navigation and functionality of the service could be improved.

Following is an actual example of a user interaction. The user initially answers with a ‘no’ that needs verification resulting in a follow up question that is answered by a ‘definite no’:

chatbot>
When asked, some people claim that the discussion board in programming is easy to navigate. Do you support that view?
user>
no
chatbot>
Do you say that because you have not got an opinion?
user>
i have got an opinion, but frankly, it's not a positive one
chatbot>
Really!

When analysing all the chat logs there is one common trend throughout; the user takes a few introductory statements to reply with anything other than yes or no, just

like in the previous example, and for that reason the chats probably ought to have been made longer than in the pilot. The cause of this is probably that the users need time to familiarise themselves with the interface of the chatbot, however when the student starts to communicate more with the chatbot, more values are being retrieved out of the evaluation process than is normally the case. For instance, in the following example the user is asked if she agrees that the content of the assignments is challenging. The response is that they are challenging enough:

chatbot >
So you agree. Would you agree with this statement, the content of the assignments in programming is challenging?
user>
It is challenging enough
chatbot >
It is challenging enough, so what would you say to this? Would you say that the content of the assignments in programming is challenging more often than not?
user>
yes
chatbot >
I'll remember that.
This is all for this time. Thanks for your help in our research. Have a nice day.
user>
and you

Here the chatbot wants to clarify whether that means they are challenging more often than not, which they are. This kind of communication shows the human evaluator that not only does this user believe they are challenging, which a multiple choice question would elucidate too, but also that they are challenging the user to the limit, as this user clearly is stretched in the assignments. During a normal evaluation process this kind of knowledge would be difficult to determine, as a grading system would only show that the user finds that they were difficult with some score.

It could be argued that the chatbot just works as an automated interviewer, and a human evaluator still has to read through all of the logs, however the chatbot automatically stores what it infers from the communication using the knowledge in the ontologies. These ontologies can be examined using semantic web tools and statistical tools. What the human evaluator needs to do is extract the gist of the value-added information from within the chat, as this is information which would be extremely difficult to automate. A core feature of semantic web ontologies is the automated inferencing that it enables. Normally formal ontologies do not contain contradicting statements, but because the knowledge base is grown based on the users' inputs, this ontology inevitably contains contradicting statements, and a new semantic layer is built up through the use of the reifications about who-said-what-when. This means that standard OWL inferencing engines cannot be used to infer new knowledge from the ontology, as they do not utilise the reifications when making inferences. To utilise this feature a new inferencing engine is needed, which operates on clusters of

knowledge or potentially uses statistics on the statements as well as standard inference rules.

An important issue when examining if this is a viable evaluation method is to see if what the chatbot infers is correct. There are, in principle, two different mistakes that can be made by the chatbot; firstly, it can infer something incorrectly from the chat and, secondly, it could overlook something within the chat and thereby fail to infer the intended meaning. Clearly of the two different mistakes the first is the one with the severest consequences as the statistics resulting from the chatbot would be wrong, whereas the second would leave out information which possibly could be rectified by the human evaluator. In our pilot a total of 60 questions were asked, out of these 58 were understood correctly leaving a 96.7% success rate. Both of the incorrectly inferred answers were due to misunderstood communication between the chatbot and the human (the user did not understand the chat correctly), and they were both stored with a wrong interpretation. Although a very high success rate, it would be valuable to examine the two misinterpreted answers to see if or how it would possible to change the system so that it would infer them correctly.

The first misinterpreted answer is a situation where the user clearly (when analysing it as a human) does not have an opinion. The chatbot tries to elaborate by asking further questions and in the end the user answers by saying how the user believes it should be. This is an excellent example of the value-added information that comes through when using the evaluation chatbot. However an answer indicating how it should be, inferred to 'how it is', does skew the results. This kind of misinterpretation is probably difficult to rectify, as the value-added information provided by the user clearly states that she believes the assignments ought to be challenging. However, it is the user causing confusion to the inference process, because she does not know whether the assignments are challenging or not. She could have answered that she did not know, and that would have been inferred correctly. As it is her answer now possibly provides us with more information, even though the statistical result is skewed, as we now know that this student wants challenging assignments, and that they sometimes are that but not always.

The other example where the answer is misinterpreted is a situation where the user apparently has misread the question, answering with an agreement although stating that he disagrees

chatbot>
Let's try a new question. It has been observed that the content of the discussion boards in programming are not off topic. Do you support that sentence?
user>
No they are on topic.
chatbot>
The content of the discussion boards in programming are off topic! I'll remember that.

Ambiguities in natural language are probably one of the most serious and difficult problems within natural language processing (Allen, 2004) and it certainly does not

help when the human response is based on a human error. Because the human seemingly disagrees with the statement (starting with a no), the chatbot infers that he disagrees, without checking that the user is actually disagreeing with the opposite of what he was asked (the human error). What can be learned though from this example is that the structure of the sentences could be kept simpler than they are. For instance having double negatives is not a good practice, as they might add to the confusion of the users, so removing 'not' and changing 'off topic' to 'on topic' in the knowledge base would probably enhance the usability and inference capabilities of the chatbot.

The generally positive results led to a smaller second experiment, created to test the versatility of the ontological approach. Another ontology was designed, which described generic quality of webpages. This new ontology was easily integrated into the old conversational tree (which only took three to four hours), and was then used to evaluate the website of an EU project. The exercise showed that this approach allows easy exchange of domain knowledge through the use of the domain ontologies. The results were, however, quite disappointing. Although the chatbot was externally available for a period of a month, only four users decided to try it, and an additional user tried to crash the chatbot by looping chats back into the chatbot system, creating an infinite loop. A lesson from this is that chat surveys are not well-suited to completely open environments such as online webpages, where people are not expecting to be surveyed, and that users should be signed-in or give their details as part of the chat to limit abuse. The exercise showed, however, that the scripting language coupled with ontologies which provided the domain knowledge is a very promising method. The development time was minimal compared to implementing a completely new set of conversational trees to accommodate for the new domain questions and expected answers.

The results of this have been positive, and the underlying code has been open sourced (http://code.google.com/p/owllang/) so that others can use the system for generating both survey chatbots and completely new chatbots. This has led to it being used for feedback in an ITS system of I-TUTOR. The integration has technically proven to be simple. The biggest developmental change included upgrading the system to a newer version of the underlying chatbot system, and developing a chat interface within Moodle. There is still a substantial amount of human assessment needed of the resulting chats to harvest the results of the chats, so feedback cannot be used immediately within the system. Furthermore the chatbot system is limited to English whereas the full ITS system is multilingual. This is an opportunity in disguise as it will provide statistical results from users who use traditional feedback mechanisms and the new chatbot system. Unfortunately it is too early to provide results from actual usage of the system within the ITS, however there will be results to report for the presentation at the conference.

6 Conclusions

The chatbot system has proven to be a flexible system that can be used in many different situations where feedback is needed through a survey of users. The use has primarily been within eLearning settings where it has proven to provide deeper information about the users' opinions than normal Likert style surveys. It has been observed that the system is accurately concluding what users believe and that in the rare cases where it has misunderstood the user it has been caused by ambiguities introduced in the natural language. Although in most cases the system "realizes" this and therefore asks the user follow up questions about the particular question.

The system uses formal ontology tools from the Semantic Web to form knowledge bases. These are however used to store informal knowledge, i.e. knowledge of which there is disagreement or no stringent formal definition. This has been achieved using a closed world assumption where only things that are mentioned are known, and a heavy usage of reification, i.e. statements about statements.

7 Acknowledgements

The authors would like to acknowledge the EU LLP funding body which funded the initial work on this chatbot system through the Triangle project and recently the further development through the I-TUTOR project. Several of these projects' partners helped in providing feedback about the usage of the chatbot. This work would not have been possible without them.

8 References

1. S. Gianvecchio, X. Mengjum and W. Haining, "Measurement and classification of humans and bots in internet chat," in Proceedings of the 17th conference on Security symposium, Berkeley, 2008.
2. M. Bates, "Models of natural language understand," Proceedings of the National Academy of Sciences of the United States of America, vol. 92, no. 22, pp. 997-9982, 1995.
3. J. Allen, Natural Language Understanding, 2nd, Ed., Delhi: Pearson Education, 2004.
4. G. K. Zipf, "Selected Studies of the Principle of Relative Frequency in Language," 1932.
5. R. Wallace, "AIML Overview," 2012. [Online]. [Accessed 10 9 2012].
6. J. Wiezenbaum, "ELIZA - A computer Program For the Study of Natural Language Communication Between Man and Machine," Communications of the ACM, vol. 9, no. 1, pp. 36-45, 1966.

7. N. Guarino and P. Giaretta, "Ontologies and Knowledge Bases: Towards a Terminological Clarification," in Towards Very Large Knowledge Bases, IOS Press, 1995, pp. 25-32.
8. W3C, "OWL Web Ontology Language Overview," 2004. [Online]. Available: http://www.w3.org/TR/owl-features/. [Accessed 14 9 2012].
9. W3C, "RDF Primer," 2004. [Online]. Available: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/. [Accessed 17 9 2012].
10. T. Mon-Tin, L. Chun-Hung, L. Cheng-Wei and H. Wen-Lian, "Extending Knowledge of AIML by Using RDF," in Workshop on Semantic Technology for Learning, Hiroshima, 2007.
11. H. Al-Zuhaide and A. Issa, "OntBot: Ontology based chatbot," in IEEE Fourth International Symposium on Innovation in Information & Communication Technology, Amman, 2011.
12. R. Phillips, J. Bain, C. McNaught, M. Rice and D. Tripp, Handbook for Learning-centred Evaluation of Computer-facilitated Learning Project in Higher Education, Perth: Murdoch University, 2000, pp. 1.3-1.4.
13. U. Ehlers and J. Pawlowski, Handbook on Quality and Standardisation in E-Learning, Berlin: Springer, 2006, pp. 31-49.
14. J. Harvey, Evaluation Cookbook, Edinburgh: The Learning Technology Dissemination Initiative, 1998.
15. R. Likert, "A Technique for the Measurement of Attitudes," Archives of Psychology, vol. 140, p. 55, 1932.