Development of an Intelligent Answering Machine based on LMS Knowledge

Sergio Martín, Elio Sancristobal, Rosario Gil, Gabriel Díaz, Manuel Castro, Juan Peire
Spanish University for Distance Education (UNED)
{smartin, elio, rgil, gdiaz, mcastro, jpeire}@ieec.uned.es

Abstract - **Nowadays, in university environments, communication between teachers and students cause, in many cases, an excessive demand on the teacher's response capacity. This demand is even stronger in Distance Education, as in the case of UNED (Spanish University for Distance Education) where usually this communication is via email. The main objective of the paper is to describe the research done to develop a tool that improves and optimises these relationships by means of more efficient use of the available information of the organizations. In order to achieve this goal, we are developing an intelligent manager able to answer the students' questions automatically, using the knowledge stored in Learning Management Systems (LMS) as dotLRN, Moodle, WebCT; indexed in search engines as Google or Yahoo; or data repositories as Wikipedia or institutional databases.**

Index Terms - interoperability, learning management systems (LMS), search engines, semantic web.

BACKGROUND

In university environments, the communication between teachers and students causes, in many cases, an excessive demand on the teacher's response capacity. This demand is even stronger in Distance Education, as in the case of UNED (Open University of Spain) where this communication is usually by email, phone or e-learning platform.

The difficulties involved in maintaining the personal communication flow between the group members make it impossible to achieve an appropriate level of monitoring of the students by the teacher [1].

The usage of these technologies creates new problems that require new and innovative solutions from both pedagogical and technological points of view.

As a consequence, UNED has developed many projects oriented to solve this problem, such as PedCare [2] in which a new methodology based on CRM, and named LRM [3] (Learning Relationship Management), was created. It optimized the treatment of the information flow and communication processes between student and teachers. It also monitored and evaluated the student process itself.

This methodology is completed with the implementation of intelligent tools based on an open and modular architecture that enables integration with most of the commercially available Learning Management Systems (LMS) offering new and complementary services that will satisfy the needs.

OBJECTIVES

The main goal of the system is to improve and optimise these relationships between the students and teachers, or in other environments, for example companies and customers, by means of a more efficient use of the knowledge stored in the new e-learning platforms [5].

The real problem in our organization is the communication and dispersion of information. We are using several e-learning platforms with information about courses and technical documentation, such as dotLRN, WebCT and Moodle. Our organization also has information available on the official Web Site, in the faculties Web sites and finally in every department Web Site.

When a student needs to know something about a course, or when a staff member is looking for some concrete information, usually a lot of time is spent, because it is like to looking for a needle in a haystack.

On the other hand, the students have many ways to communicate with the teachers. For instance, they usually use the phone or e-mail, but currently they are also using the internal web mail of the dotLRN or WebCT platforms, with the consequent problem of dispersion. Teachers have to look in too many places for the students' questions.

For that reason, UNED is working on the development of a system that has the capacity to answer automatically the students' questions. For that complex task information already available in all these knowledge sources will be used: Web Sites, Learning Management Systems (LMS), databases, etc.

LOOKING FOR SOLUTIONS

In first place, it seems appropriate to take a look at possible solutions to the problem. We will study the systems that can help us find the solution. For example, one important knowledge source in the Internet is Wikipedia, where people look for many different kinds of information. If we look for something closer to an answering machine, we will find the Indiana University 'Knowledge Base' or artificial intelligence-based languages as AIML. All these methodologies will be part of the final solution, helping in the design of the new system.

I. Wikipedia

Wikipedia [11] contains a huge amount of information. Here people can find answers to a lot of questions, using the internal search facility or by using a search engine such as Google.

The problem is that it is not appropiate for organizational information, such as professors’ timetables in University or product information in companies. So it seems necessary to look for solutions to enable new content to be added in an easy way.

In addition, Wikipedia is not able to give answers to questions in natural language. That is to say, if someone searches for ‘Mahatma Gandhi’, Wikipedia will return a correct answer, but if it is asked ‘hello, I would like to know where was Mahatma Gandhi born, please’, the results are not so positive. The reason is that Wikipedia searches every term in its knowledge repository, creating a list of results with a lot of noise.

Our solution must be able to accept natural language, because it will receive the questions that students send to a teacher.

II. AIML (Artificial Inteligence Markup Language)

The Artificial Intelligence Markup Language is a derivative of XML (eXtensible Markup Language) [9]. Its goal is to enable pattern-based, stimulus-response knowledge content to be served, received and processed on the Web and offline in the manner that is presently possible with HTML and XML. This technology is usually used to create conversation applications (bots) about different themes.

AIML describes a class of data objects called AIML objects and partially describes the behavior of computer programs that process them [10]. AIML objects are made up of units called topics and categories, which contain either parsed or unparsed data. Parsed data is made up of characters, some of which form character data, and some of which form AIML elements. AIML elements encapsulate the stimulus-response knowledge contained in the document. Character data within these elements is sometimes parsed by an AIML interpreter, and sometimes left unparsed for later processing by a Responder [8]. This Responder handles the human-to-bot or bot-to-bot interface work between an AIML interpreter and its objects.

In order to have the system ready to work, it is neccesary to create an AIML file with all the rules involved in the conversations or used to answer questions. To build this AIML file, in first place, every expected question must have been defined, having in mind how to make the pattern matching using predicates. In the figure 1 it is showed a simple example of this usage:

```
<category>
 <pattern>WHAT IS YOUR NAME</pattern>
 <template>My name is
 <bot name="name"/>.
 </template>
 </category>

 <category>
 <pattern>WHAT ARE YOU CALLED</pattern>
 <template>
 <srai>what is your name</srai>
 </template>
 </category>
```

FIGURE 1
AN AIML FILE SHOWING PATTERNS AND TEMPLATES.

As can be seen in this example, the template is applied if the question matches the defined pattern. The first category simply answers an input "what is your name" with a statement of the bot's name. The second category, however, says that the input "what are you called" should be redirected to the category that matches the input "what is your name", in other words; it is saying that the two phrases are equivalent.

Templates can contain other types of content, which may be processed by whatever user interface the bot is talking through. So, for example, a template may use HTML tags for formatting, which can be ignored by clients that don't support HTML.

This kind of technology is used to create conversation bots about several themes, but it is necessary to define, one to one, every rule, specifying the pattern to match and the template to apply. It is hard to maintain because a plain AIML text file with several thousound entries is impossible to manage.

III. Knowledge Base: A Custom-built content manager of Indiana University

The Indiana University Knowledge Base (KB) is an online collection of short answers to questions about information technology (IT), especially those relevant to students, faculty, and staff [18]. The Knowledge Management team writes the documents that make up the Knowledge Base and publishes them to the web.

The ‘Knowledge Base’ is made up of more than 12,500 files (each a one KB document), most consisting of a question and an answer, formatted in Knowledge Base Markup Language (KBML), a proprietary text markup language described using SGML.

Thousands of KB documents have been written by telephone, email, and walk-in consultants. Many documents originate from questions asked of the consultants by the computing community. Also, as technologies change and new computing tools are introduced at Indiana University, UITS system and service administrators and others from departments across the entire Indiana University system revise and contribute information to help prepare for changing support requirements [12]. It can be seen an example of how it works in the figure 2.

FIGURE 2
INDIANA UNIVERSITY KNOWLEDGE BASE.

How does Knowledge Base find answers to questions? Basically with a search engine that search in the KBML documents source. This solution is very practical because indexing new contents is very easy. The search engine will do it for you.

In this case, it is not neccesary to create inference rules, associating a question to an answer. The search engine will do the pattern matching automatically and return a list of results, with the best documents to answer the question.

However, this system also has several limitations relating to scalability and the neccesity of a team refreshing the content.

The use of XML-based format documents with both the question and the answer enables easy searching processes, but, in fact, this system is not the natural processes. Company's information does not appear in short documents with both the question and the answer, but in long document with a lot of information. So, it is neccesary to create custom-built content in order to maintain this system.

On the other hand, the management of several thousand documents is a huge task that can become impossible as the number of documents increases.

THE SOLUTION: A CUSTOM-BUILT SEARCH ENGINE USING THE INFORMATION AVAILABLE IN OTHER SYSTEMS

Thanks to the conclusions obtained from these systems (Wikipedia, AIML, Knowledge Base) we know that if we are looking for a system able to find answers to a question, we will need a huge repository, containing hundreds of thousands answers which is easier to maintain. In other words, distributed systems made up of knowledge repositories with the particularitity of contain information in several formats.

For example, organizations used to have a lot of available information using Microsoft Word documents or PDF files. In this way, our system must be able to obtain answers using the information already available in each organization, ensuring easy management, because the employees will be the responsible for each document.

The best way to reach this goal is to use a custom-built search engine that will access the information hosted on other platforms. For example, the information of wikipedia, the knowledge base of Indiana University or course contents in Learning Management System.

Thanks to this methodology, it will be possible to use dozens, or even more, distributed and heterogenous knowledge sources.

In addition, the use of custom-built search engines integrated with e-learning platforms becomes a powerful tool. It allows, on one hand, the use of several languages (English, Spanish, Portuguese, German, Czech, etc.) without the need to re-programme the tool or change the configuration. On the other hand, it makes possible to add new accessible knowledge in real time. In this way, the custom-built search engine will provide instant and updated answers for the students using the content stored in the e-learning platforms.

But the question is: how to obtain information from several heterogeneous sources? The answer is very close to the concept of a metasearch engine. In first place, it is necessary to build a search engine for every knowledge source, or if a good search engine already exists, to make use of it.

Once every knowledge source is indexed with a search engine we must obtain the generated information and filter it to obtain only the answer to our question. Finally, if it is possible, we must compare it with the result from other sources to get the best result.

In order to reach this goal it seems necessary to create a system between all the search engines [13]. This software will know where to look for an answer, how to filter it and how to compare it with answers from another sources.

The search engines can be created using any technology, because the middleware system will be able to obtain the results from every source. However, after a period of research, the chosen solution for the creation of search engines is the Lucene technology.

Lucene is a free/open source information retrieval library, originally implemented in Java. It is supported by the Apache Software Foundation and is released under the Apache Software License. While suitable for any application which requires full text indexing and searching capability, Lucene has been widely recognized for its utility in the implementation of internet search engines and local, single-site searching [14] [15].

Thanks to this technology, the creation of search engines becomes easier. In addition, there are several free Lucene-based search engines available already developed in the Internet.

In addition, this system will allow several methods of communication. That is to say, the user will have freedom to decide which method use to ask a question: an e-mail, a message in a WebCT forum, an internal e-mail in dotLRN, etc.

ARCHITECTURE

The design of the architecture is very modular and open, enabling a high level of reusability [6]. It has been designed using two middlewares, which provide compatibility, on one hand between the methods to send and to receive the messages: POP3 and SMTP communication standards for email, accessing to the Moodle databases, etc. And on the other hand, between the methods to access the knowledge sources: HTTP1.1 to access to the data of the web sites as Google or Wikipedia, to access to the dotLRN databases, etc.

Consequently, in order to add a new input/output method or a new knowledge source for the tool, the middleware to interact with it must only be developed, like a little "driver" that knows how to interact with both sides.

In our architecture, these middlewares are internally designed using several interfaces (Figure 3). In first place, the communication with the user is made up of two interfaces that communicate via email, dotLRN, Moodle, WebCT, etc.

To achieve this goal there are several classes that implement the methods of this interface, for example, dotLRNImplementation class, which connects with the dotLRN database and obtains the possible user messages to the teacher. This class will obtain the main question words thanks to a filter process and finally will send them to an AnswerManager that will be able to obtain the most suitable response.

In the same way, a moodleImplementation class and a pop3Implementation class are implemented to allow the communication with Moodle and with the e-mail inbox.

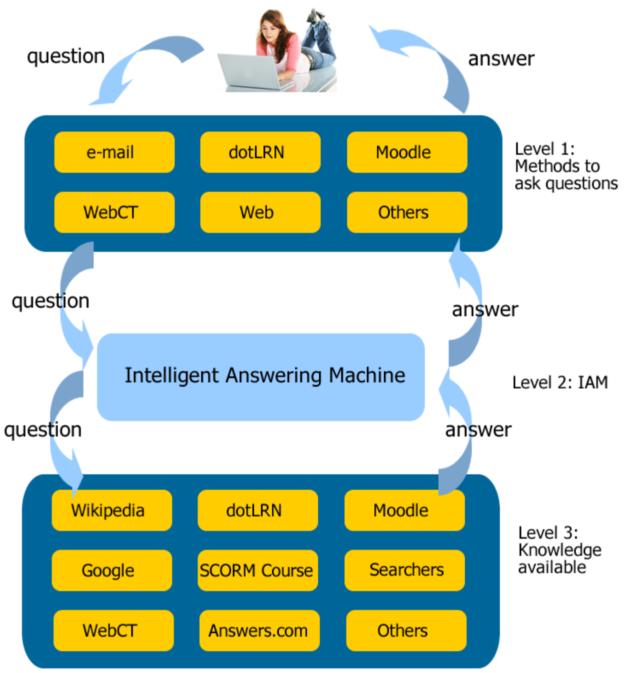

FIGURE 3
SYSTEM'S ARCHITECTURE.

On the knowledge repositories side there is another interface that controls communication with the knowledge sources, searcherInterface. In the group of classes that implement this interface searbloxInterface and googleInterface know how to interact with the correspondent searchers, dotLRN and moodleInterface are able to obtain data from these mentioned platforms.

CURRENT STAGE

In this moments, UNED is working on a prototype, with a preliminary design, that enables some of the said functionalities. It allows students to ask question by e-mail or by an on-line platform. In the same way, the answers are sent using e-mail or web protocols.

Currently, the prototype is available to students in a experimental stage in several environments inside UNED: Industrial School, Psychology Faculty and Electrical Department. But it is under an exhausted control by the teachers because it still commits some mistakes.

Nowadays it is able to answer questions related to administrative tasks and information about teachers and subjects. But it is not able to answer technical question yet, due to currently there is not an available repository suitable for the tool.

THE MOST IMPORTANT THING: THE SOURCES

It seems obvious that no matter how good, complex or last-generation technology your system is, the most important thing, when we are working with a question-answer system, is the quality of the information (Figure 4).

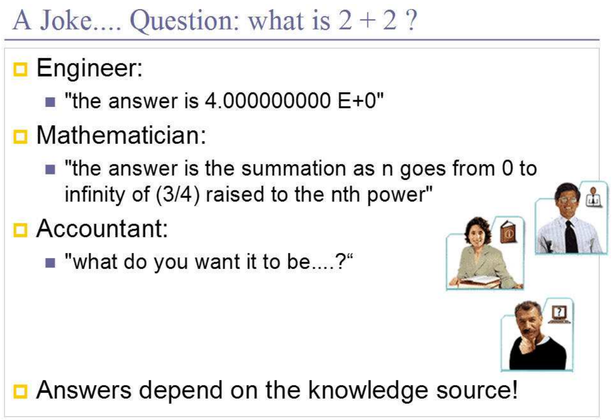

FIGURE 4
A JOKE SHOWING THE IMPORTANT OF THE QUALITY OF THE SOURCES.

However, in practice, this concept is very subjective. How is it possible to determine if a result is good or not? or how can a result be rated? Of course, it depends on who is assessing it.

USERS POINT OF VIEW

In this line of work, for instance, if a student has a question for a teacher or a tutor, he will be able to ask him via email or via an e-learning platform. The tool will receive the question, after reading the suitable inbox, it will find the most appropriate knowledge source for the response. Sometimes, it may be an institutional database, a content search engine such as Google, an e-learning platform or another kind of data repository such as the Wikipedia web site. Finally the tool will obtain and forward the most suitable response to the student.

In this way, the student will have freedom to decide which kind of method to use to ask the teacher, and our agent will decide which is the most suitable knowledge-base to answer him, making possible the interoperability between several e-platforms.

Another possible scenario could be a student sending a question to the teacher via a message in dotLRN platform and the response will be obtained from a SCORM course hosted in Moodle.

In another situation, this student would send the question via Moodle and the response would be obtained via dotLRN repository. The answer would be forwarded by email. The number of possibilities is really very high, giving a great versatility to this tool.

BUSINESS POINT OF VIEW

From a business point of view, these kinds of systems offer advanced added-value services to existing e-learning platforms and due to the independence of the tools from specific technologies, it is ensured that they have a large potential market.

This methodology can be offered to customers who are interested in improving or revitalising their actual learning processes. Possible applications are for organisations that want to improve relationships with their learners and for those who want to improve their evaluation methods of students progress or course quality.

Other potential clients are education centres (private or public universities), public administration and organisations with authoring Virtual Campus as well as e-learning providers taking advantage of the added-value services for: improving management of communication channels with students, improving monitoring of students activities in the learning and evaluation process, or offering personalised learning contents.

FEEDBACK: AN IMPORTANT ALLY

In addition, the tool also provides a quality control system than allows to improve and monitor the student's activities, thanks to a feedback system. It provides information about the quality of the answers given to the user.

Every time a student receive an answer he can valuate it depending on its goodness or badness, as it can be seen in the figure 5. In addition, the same answer is sent to a teacher, who can valuate if it is suitable or not for the student's question.

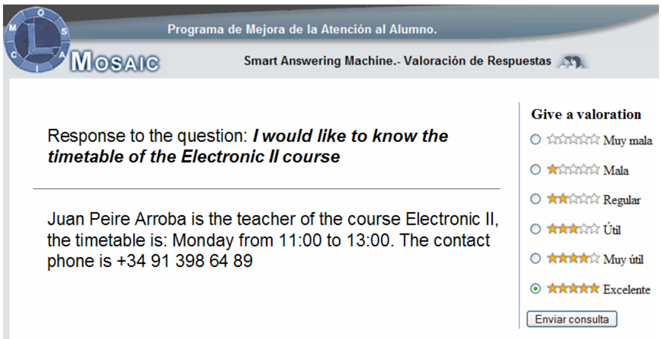

FIGURE 5
A SAMPLE OF ANSWER. IN THE RIGHT CAN BE SEEN THE VALUATION FORM.

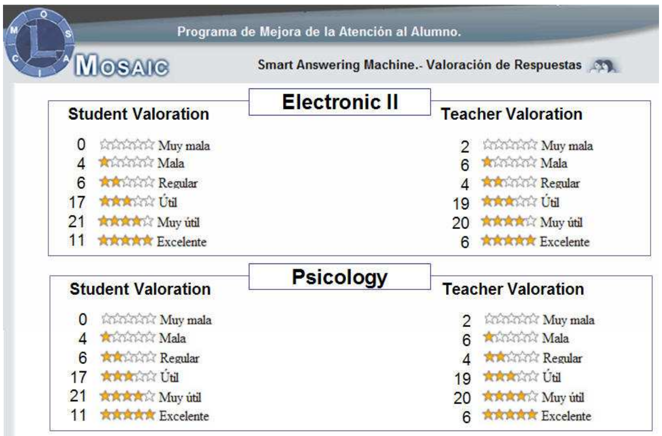

FIGURE 6
VALUATION SUMMARY.

Both valuations provide useful information to improve or change the knowledge sources, as it can be seen in the figure 6. For example, if a teacher realises that the valuation of one course is bad he will change the knowledge source or will improve the way in which the information is retrieved.

CONCLUSION

This paper describes the research and the given steps needed to design and develop a system able to answer questions automatically. For that task, the solution is a middleware that uses the information of several custom-built search engines. This engines will know where to look for an answer, how to filter it and how to compare it with answers from other sources.

In this way, the system will ensure, in first place, the integration with several knowledge sources, independently of the technology: e-learning platforms using SCORM, LOM or IMS-LD, content searchers, institutional databases, etc. In the second place, the independence of the communication methods used by the user: e-mail, e-learning platforms, etc.

Finally, an improvement of the monitoring process of the student activities, thanks to the feedback system, that provides information about the quality of the answers given to the user.

The development of projects related to these technologies keeps the universities in the technological vanguard. It creates a new type of education based on an improvement of the attention provided by the educators and a personalization of the contents with the help of the new technologies. In addition, this solution makes possible to know the needs of the students in every case, allowing to solve the problems that appear in the learning process.

ACKNOWLEDGMENT

The authors would like to acknowledge the Spanish Science and Education Ministry and the Spanish National Plan I+D+I 2004-2007 the support for this paper as the project TSI2005-08225-C07-03 "MOSAICLearning: Mobile and electronic learning, of open code, based on standards, secure, contextual, personalized and collaborative".

REFERENCES

[1] Martin, S. et al "Experiencias e introducción de dispositivos móviles en la Enseñanza a Distancia", *Simposio sobre computación ubicua e inteligencia artificial ambiental*, 2005.

[2] Rivilla, I. et al "Development and Implementation of a Collaborative Environment for Education, eAdoption and the Knowledge Economy", *Issues, Applications, Case Studies/ IOS Press*, 2004.

[3] Arjona, M. et al , "Knowledge for Management Methodologies for a Global Distributed Organization", *6th International ITF Workshop & Business Conference*, 2001.

[4] Rosic, M. et al, "Intelligent tutoring interoperability for the new Web", Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean, Vol 2, 2004, Pages 739 – 742.

[5] Kecheng, L. et al, "Interoperability for digital learning resource management", *ICALT 2005. Fifth IEEE International Conference on Advanced Learning Technologies*, Pages 521 – 525, 2005.

[6] Arapi, P. et al, "ASIDE: An Architecture for Supporting Interoperability between Digital Libraries and ELearning Applications", *Advanced Learning Technologies, 2006. Sixth International Conference on Advanced Learning Technologies*, 2006, Pages 257 – 261.

[7] Jianna, J. et al, "An Example Oriented On-line Java Tutorial for University Students", Proceedings of the 3rd International Symposium on Principles and Practice of Programming in Java, 2004.

[8] Artificial Intelligence Markup Language (AIML) Version 1.0.1 . On-line. http://docs.aitools.org/aiml/spec/. Query on April 2007.

[9] AIML in Wikipedia. On-line. http://en.wikipedia.org/wiki/AIML. Query on April 2007.

[10] AIML Documentation. On-line. http://aitools.org/Documentation. Query on April 2007.

[11] Wikipedia. On-line. http://en.wikipedia.org/wiki/Wikipedia:About. Query on April 2007.

[12] Indiana University Knowledge Source, "How does the Knowledge Base work?" On-line: http://kb.iu.edu/data/acte.html Query on April 2007.

[13] Cafarella, M., Etzioni, O., "A search engine for natural language applications", *Proceedings of the 14th international conference on World Wide Web*. ACM Press.

[14] Lossau, N., "Search Engine Technology and Digital Libraries", *D-Lib Magazine*, Volume 10 Number 6, June 2004.

[15] Rocha, C., Schwabe, D., Poggi, M.,"A hybrid approach for searching in the semantic web", *Proceedings of the 13th international conference on World Wide Web*. 2004. ACM Press.

[16] Aleman-Meza, B., Halaschek, C., Arpinar, I., and Sheth, A. "Context-Aware Semantic Association Ranking", *Proceedings of SWDB'03*, 2003, 33-50.

[17] Deniman D., Sumner T., "Merging Metadata and Content-Based Retrieval", *Journal of Digital Information*, Volume 4 Issue 3, 2003, Article No. 231.

[18] Indiana University Knowledge Source Web Site. On-line: http://kb.iu.edu/ Query on April 2007.