An Intelligent Robotic Engine Using Digital Repository of the DSpace Platform

Rafael Luiz de Macedo, Elvis Fusco
Computing and Information Systems Research Lab
Center University Euripides of Marilia
Marilia, Brazil
e-mails: {rafaelldemacedo@gmail.com, fusco@univem.edu.br}

Abstract - **This article presents an Artificial Intelligence Markup Language as a set of eXtensible Markup Language, which is able to represent and relate expressions in natural language. This will allow the creation of intelligent robotic engines capable of maintaining a simple dialogue; however, robotic engines are limited in the amount of questions they can answer, thus, failing to maintain a simple dialogue for a long time. This paper solves the problem of the limitation of robotic engines with the development of a software that will perform a search in the repository to find information on files for generating new questions and answers for robotic engines.**

Keywords-Artificial Intelligence; Engines Robotic; Semantic Web; Repository.

I. Introduction

Current technologies have increasingly allowed the perception that it is possible to make a machine be intelligent enough to answer questions asked by the user through robotic engines built with AI (Artificial Intelligence).

The ways in which humans and computers normally communicate are very different. Humans tend to spend a lot of time in chit-chat and informal dialogue with little or no effect. Computers are known to give accurate answers, true and logical. The rate of exchange of information of most human dialogues is very low, no more than 1kbit per second, but the computer communication is faster. Alicebot/AIML is an attempt to bridge this divide [2][10][18][19][20].

One of the languages that enables the creation and development of these intelligent robotic engines capable of maintaining a dialogue is the **A**rtificial **I**ntelligence **M**arkup **L**anguage (AIML) [10][12][19][20].

Robotic engines developed in AIML language have a great limitation on the amount of questions they can answer; the main reason is that they were developed based on one or more contexts and are not able to dynamically extend the amount of questions and answers in an information bank.

With the emergence of the Internet, information is increasingly accessible, so that, only with a machine that has access to the Internet it is possible to search for new information.

The amount of information that is available daily on the Web has facilitated an increasing access for people. Consequently, the number of online stores that have articles, monographs, newspapers, magazines and other types of digital files has grown. A platform that offers that kind of repository is the DSpace [14][15][17], which is rather used as an open source platform and has several tools available to be used in the repository, such as search engine, to access files through search requests to Web servers.

The main objective of this paper is to develop a metadata parser software of digital repositories of DSpace platform, which performs searches of digital files in repository metadata to find the files related to the topics the users are seeking [13][14][15][17]. The Analyzer software will generate new knowledge database of questions and answers for the engines on top of these robotic metadata from the repository; this new information might be useful for debugging purposes, or resolving the limitation of robotic engines in only answering questions that are contained in the context that were developed.

For the development of the metadata Analyzer software, the digital repository of Center University Euripides of Marilia – UNIVEM, called Univem Aberto [1] will be used. This is because the digital repository of Univem Aberto is based on the DSpace platform.

This article is structured as follows. Section 2 presents a preview of AIML. Section 3 shows the importance of institutional repository (university). In Sections 4 and 5, the types of metadata utilized in this work and in the repository of platform DSpace are shown. In Section 6, the results obtained in this work are presented. Section 7 presents a conclusion of the article.

II. Methodology

In this paper, the methodology used for the development is divided into a few steps, as follows: (i) the bibliographical study of languages AIML and XML, (ii) a survey of the types of institutional repositories platforms (iii) the development of the application, and (iv) the related works, testing and validating the results.

A. The Bibliographical study of languages AIML and XML

The study of markup languages are required for the knowledge on how intelligent robotic engine work and for the development of the application proposed in this paper.

B. The survey of the types of institutional repositories platforms

Among the types of repository platforms, we will choose only one to be used as a basis of collecting scientific files, and we specifically use an institutional repository to develop the application on this work platform.

ISBN: 978-1-61208-329-2

C. *The development of the application*

In this work, an application was developed that searches files in the server of an institutional repository; for the development of the application, the programming language Java was used.

D. *Related works*

The related research is an important step. The articles and documents found helped in understanding markup languages AIML and XML. The study also helps to evaluate the results obtained in this work.

E. *Testing and validation of the results*

The application developed in this work was tested in a real scenario to search files from the institutional repository at Univem Open of Center University Euripides of Marilia.

III. ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE

The AIML language is a set of XML elements capable of representing and linking natural language expressions, allowing the creation of engines capable of maintaining a simple dialogue. Each set of AIML elements has one or more elements referred to as category. The categories are developed on top of a context; a category is formed by all the elements and template pattern, which are the elements responsible for interpreting the message sent by the user and send a reply message back to the user.

The Chatterbots (Chat = Chat and bot = Bot), as currently known, are robotic intelligent engines that interact with users by means of questions and answers. Currently, the amount of intelligent robotic engines used in online consultations and through voice communication devices is growing.

Companies have used these robotic engines to perform an auto-customer service for the purpose of improving the attendances and saving time. The use of these robotic engines is because they recognize the issue the customer wants to deal with in the company and redirect the customer towards the competent sector. Robotic engines are used for online sessions and for telephone calls.

A very well known robotic engine used to educate people in caring for the environment is the Ed robot, belonging to the company Conpet, along with the support of PETROBRAS [21].

The emergence of the AIML language was the result of the development of the software of PLN known as The Artificial Linguistic Internet Computer Entity (ALICE) [2]; the license of AIML is under the GNU GPL.

The robotic engines developed in AIML have two modules, namely, language and engine. The language is all the knowledge that these engines have, natural language and information developed in AIML language. The engine performs the communication between the two languages, natural and AIML, the recognition of the information contained in the two languages, so that the robotic engines can recognize written questions made by humans and answer these questions.

Robotic engines are limited in the amount of questions they can answer because the robotic engines can not reshape new questions dynamically, as a human. With this limitation, the robotic engines do not maintain a long dialogue with a human.

Alan Turing (1912-1954), in his famous essay Computing Machinery and Intelligence [12], suggested that, instead of asking whether machines can think, we must ask if machines can undergo a behavioral intelligence test, which came to be called the Turing test [11].

The Turing Test is performed by means of questions and answers made to a machine with robotic and intelligent engine for a human; the machine and the human are located in separate rooms. An evaluator will hold the questions for these two rooms, but the evaluator does not know in which room is the machine with the robotic engine.

According to the responses that the evaluator will receive, one can find out which one is the machine room and which one is the room with the human. Intelligent robotic engines that recognize natural language can't create new questions and answers autonomously, causing a disadvantage by not being able to recognize a question that is outside of the knowledge or repeat an answer already used in different questions.

The machine that can pass the Turing test is considered smart. However, currently, there are no machines capable to go through this test.

The AIML language is open source [2][19][20], thus enabling the use in research of improvement in creations of robotic intelligent engines able to recognize information written in natural language.

To create a robotic motor using the artificial intelligence markup language, you need the use of a development platform. There are several open sourceplatforms, paid and developed in various programming languages such as Java, Python, C and C + +.

The platform ProgramD is the most used platform when it comes to developing robotic engines in AIML language and the most complete resource on the language, because the platform being developed in Java language and be open source [4].

A. *Funcionality of engine with AIML*

In this subsection, we present some commands (elements) of AIML used in the creation of a new base of questions and answers for robotic motors.

In the creation of a new base of questions and answers, it is necessary to utilize standard elements of the language AIML. The standard elements are: category, pattern, template, star, aiml and xml.

Figure 1 shows the base of questions and answers organized with elements of AIML language, utilized for robotic engines.

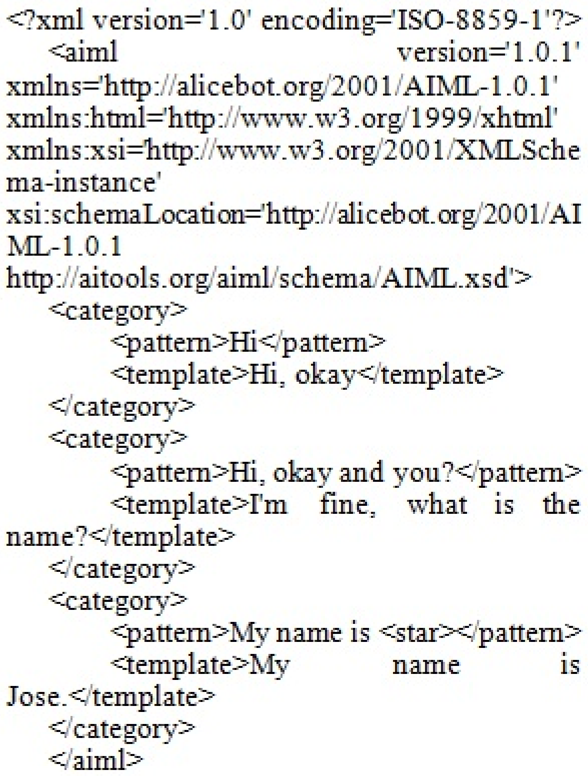
A new database of questions and answers is started with elements XML and AIML. These elements define which versions of XML and of AIML language is being used at the base [20].

The category element is a set of questions and answers; each category has elements pattern and template.

The pattern element is where a possible question is declared that the human can ask to chatterbot.

ISBN: 978-1-61208-329-2

The template element is localization of an answer of a question declared on a pattern element.



```
<?xml version='1.0' encoding='ISO-8859-1'?>
    <aiml                          version='1.0.1'
xmlns='http://alicebot.org/2001/AIML-1.0.1'
xmlns:html='http://www.w3.org/1999/xhtml'
xmlns:xsi='http://www.w3.org/2001/XMLSche
ma-instance'
xsi:schemaLocation='http://alicebot.org/2001/AI
ML-1.0.1
http://aitools.org/aiml/schema/AIML.xsd'>
    <category>
          <pattern>Hi</pattern>
          <template>Hi, okay</template>
    </category>
    <category>
          <pattern>Hi, okay and you?</pattern>
          <template>I'm   fine,   what   is   the
name?</template>
    </category>
    <category>
          <pattern>My name is <star></pattern>
          <template>My         name         is
Jose.</template>
    </category>
    </aiml>
```

Figure 1. Based of questions and answers.

IV. DIGITAL REPOSITORY

With the need to disseminate the works produced by institutions, without relying on a Publisher to publish the work, institutions began to spread this information on the Internet on their own by creating several tools called digital repositories [13][14][15][17].

The first digital repositories by institutions began to be developed in 2002; the digital institutional repositories began to play the role of the Publisher.

Digital repositories have the main functions of storage, dissemination and durability of digital files, making it easy to access the files that are submitted to these tools.

The institutions have used the repositories of different platforms, disseminating their scientific journals, monographs, Ph.D. theses and other work carried out in the institutions over 11 years; after the development of the first digital repository, repositories used is the platform DSpace.

DSpace is an Open Source software platform, responsible for the storage, dissemination and durability of digital files. It was created by MIT (Massachusetts Institute of Technology) and Hewlett-Packard [14].

The DSpace platform is geared to the academic area, the purpose of which is serving as a basis for the future development to address the long-term preservation of files and access problems [14].

The registration site of providers of OPEN ARCHIVES repositories is a table with information where digital repositories are registered; they are currently registered and the record repository #2140 is the Center University Euripides of Marilia, Univem Aberto Repository.

The Center University Euripides of Marilia-DATA has deployed in 2012 the digital repository DSpace platform for the purpose of disseminating the scientific papers, monographs, theses of master's and doctoral degrees from the faculty and other work carried out by its students, teachers and researchers.

The Institutional Repository of UnB is a set of services offered by the Central Library for the management and dissemination of scientific and academic production of the University of Brasilia. The content is publicly available; it is widely accessible [8].

To include their scientific production in the repository, teachers, researchers and students graduating from UnB must complete and sign a term of authorization and return it to the Management of Digital information (GID). This document is signed, scanned and sent along with the file by email.

With the huge amount of files that are made available in institutional digital repositories, it is necessary to standardize rules to manage and identify each file that is submitted in the repositories.

When a file is submitted for digital repository DSpace platform, some information about the file is needed.

All the information entered in the fields of metadata is used to catalog a file in the repository, using the standard Dublin Core metadata (DC) [3][13].

To integrate intelligent robotic engines developed in AIML, questions with the information about the files that the repository user is seeking should be defined. With this information collected, robotic engines integrated with metadata Analyzer software developed in this work carry out requests for search of Univem Aberto files in the repository. Metadata Analyzer software called XML2AIML treatment will apply to the information that is contained in the response from the digital repository Univem Aberto in order to generate new knowledge bases to intelligent robotic engines, in the form of questions and answers.

V. THE STANDARD METADATA DUBLIN CORE

With the great increase of digital files posted on the Internet, there was the need to develop standards that identify the exact description of each piece of information from the files, i.e., to develop metadata standards.

Metadata means data about data. The metadata are forms of cataloging all information from a file, the same way they are made in real libraries: each book has a tumble of identification, are arranged by area and alphabetically by title. The metadata are intended to document and organize digital files in a structured way, making it possible to identify files through standardized data such as author, title, and summary.

Metadata is defined here as given that describe attributes of a resource, characterized their relations, and enables its recovery and effective use and its existence in electronic environment. Metadata usually consists of a set of data elements, where each element describes an attribute of the resource, its administration or use [5][9][16].

All the reasons why the indexing and cataloging are required for printed sources apply even more strongly to the metadata for electronic documents [6][7].

Digital repositories have metadata for cataloguing all files that are submitted. The standard used in digital repositories is the standard Dublin Core which are collections of metadata [6][7].

DC is not for replacing some richest models with AACR2/MARC Code, but only provides basic sets of elements of description, which can be used by catalogers or non-catalogers for a simple description of information resources [3][20].

With the files catalogued on the use of metadata, the same files end up being most used by users than the files that have no metadata cataloging, because of the ease in which the metadata provide the tools of Internet searches to find these files.

The concept of metadata is not something new, but the use of this term in digital environments and a variety of patterns and shapes is new. The bibliographic records that have been created by information workers in a long time must be regarded essentially as metadata. They provide analytical and descriptive information about an object [5].

Some different metadata standards to identify certain files are presented below:

- Government Information Locator Service (GILS) – Government information;
- Federal Geographic Data Committee (FDDC) – description of geospatial data;
- Machine Readable Card (MARC) – bibliographical cataloging;
- Dublin Core (DC) – data on Web pages, and
- Consortium for the Interchange of Museum Information (CIMI) – information about Museums.

New standards of metadata can be created according to the informational needs of an organization and contribute to documenting the data of the digital file.

These patterns can be viewed as metadata content standards, standards for exchanging data by electronic means and, in lastly, standards for data models [9].

Metadata is structured data blocks; each block contains information, such as author, title, where it was published, etc.; also, some information is a field that can be set for field name, type of field information, the format that is accepted by the field and other descriptions that identifies the information passed in this field.

Creating a metadata schema must establish a standard framework and terminology. Declarations of labels such as creator, author, sculptor or composer have little representatively if these fields, who all have the same function, cannot be mapped to the same unique concept. A form should be established, either through a list of authority or Affairs of a controlled vocabulary standard and so relationships will map out alternative ways to the established form [5].

In digital repositories, it is possible to develop software able to get files through requests made to repositories. These requests make use of DC metadata sets that are passed by the request.

VI. Dublin Core

The DSpace platform enables the digital repository web server to receive requests from external media, off the server. With requests, we are able to have access to all digital files that are submitted to the repository.

When we perform a request, sets of metadata elements are used, i.e., Dublin Core is used to define what are the parameters that will be passed by the address, the values passed by all parameters will be used to identify which file, or group of files is being sought in the repository.

The request made to the web server is parsed by the repository and all files that fit the information passed by DCs will generate an XML response that is sent to the agent (software) that made the request. The software developed in this work is called XML2AIML; it will be responsible for making requests to the web servers of the repositories, which in this case is the open repository web server.

An XML response from the repository contains information from the catalog file, such as author (s), title, and publication date, date on which it was submitted to the repository, summary, keywords and other information used to catalogue. This information is recorded in fields defined the DC.

The DCs are used for adding in the DSpace platform keywords (metadata) that reference the main catalog information from a file, i.e., it is possible to locate a particular file via their own metadata.

The variables used in the request address of search of the web servers of the repositories are standardized by the DSpace platform, but you can create new metadata (DC) directly from a function that offers digital repositories. This way, each institution may define multiple metadata according to each need, but for which the XML2AIML software that is developed in this paper can do search request to the servers of the repositories, you will use the metadata already created standards for the DSpace platform.

VII. ResultS

The main objective of this work is to cover the knowledge of robotic engines developed in AIML language, with new bases of knowledge created autonomously by software responsible for seeking information in digital repositories, via metadata, and treated to generate the new bases.

Digital repositories have a limitation in the material search tool, for not being very precise in finding the files when a user uses the repository. So, the software developed in this paper solves two problems: making use of Chatterbots to help repository users in searches of files and , with the deployment of Chatterbots in repositories, the information contained in the Dublin Core (metadata) of the repositories will be used by the software to cover with new AIML knowledge bases.

The creation of new questions and answers to the robotic AIML language engines are required to use new bases of information in order to analyze and treat, to generate new questions and answers on top of new information.

Figure 2 presents the operation of the chatterbot with metadata software analyzer XML2AIML.

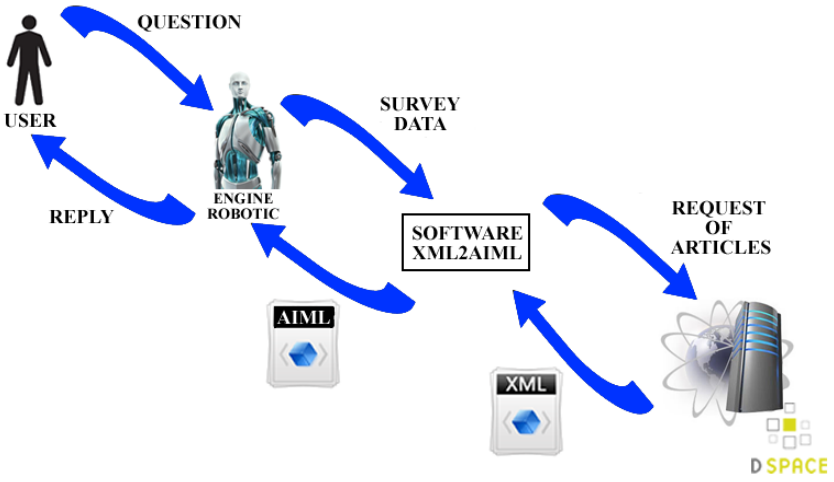


Figure 2. Model search functionality in digital repository file through a chatterbot AIML.

The functionalities are divided into four steps, namely (i) collecting user data, (ii) requesting files search, (iii) presenting questions and answers, and (iv) creating new knowledge base. The scenario creating (to test) a file search conversation in the repository is shown below.

A. *Collecting user data*

The first step is to gather as much information from the user about the material he/she is looking for. One could get this information as the chatterbot already contains questions that were created to make this collection. As the search will be driven by DC variables, the chatterbot has to collect information that can be used in these variables, that is, information such as author, title, publication date, area or a period of publication.

In the test scenario, chatterbot explains to the user whether he/she wants to look for a material, and what rules to follow. These rules were set for the chatterbot as it can understand what kind of information the user is passing/willing to obtain.

When a user is looking for materials development in AIML, the chatterbot saves that information in a variable and then to asks the user if he has any more information about this area. As the user does not have any more information, the chatterbot triggers a script element, which is responsible for calling the method colect_information (author, title, start_date, end_date, area); this method passes all the information of the chatterbot to XML2AIML software.

B. *Request files search*

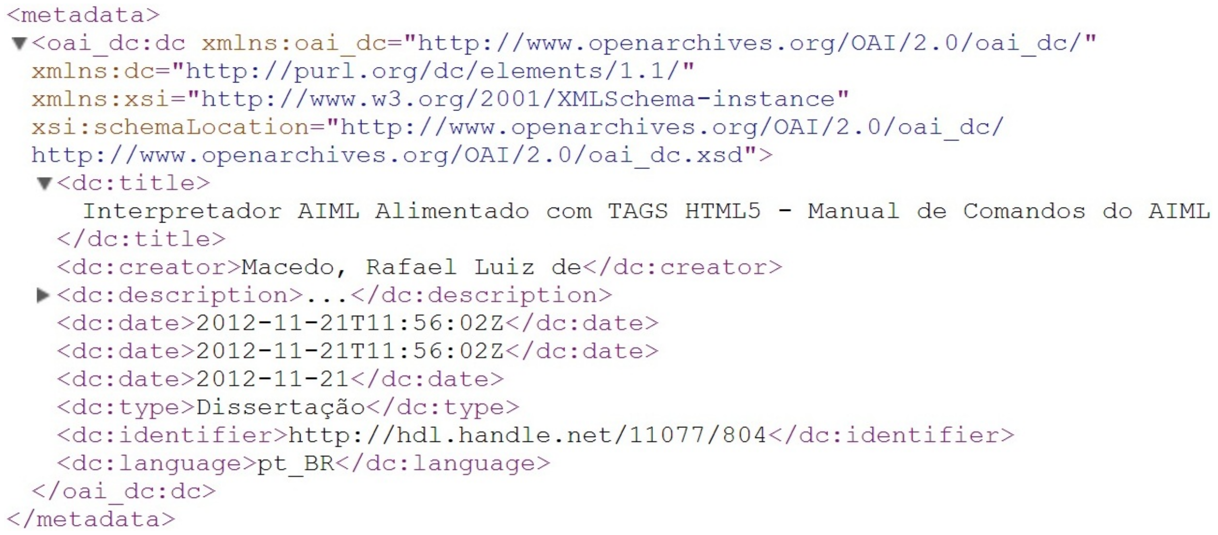
The user information was passed to the XML2AIML software; with this information, the software analyzes what types of information was received, according to the type of information, and the correct method is called.

In this scenario, the user passes information to the chatterbot on the material sought, then the software will invoke the method responsible for creating an ADDRESS that contains the variables parameter DC author and title area; even though the chatterbot has not passed information about the author and title. The XML2AIML is already scheduled to put DC in the ADDRESS; only the variables that are assigned some value in time to call the method are created in the scope of the parameter passing method.

The software with the address ready will call the functions InputStreamReader and BufferedReader of the Java library, which are responsible for making the request to the repository server and receive the response from the server. [22] is a link generated by software XML2AIML for search about the articles on institutional repository.

The server receives the values of variables from parameters passed in the address and searches the repository database for any file related to that information; after the search, it generates an XML to send as reply for the machine that made the request.

Figure 3 presents a response of a request from a server for software XML2AIML. The response received is in XML format.



```
<metadata>
▼<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
 http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  ▼<dc:title>
     Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML
   </dc:title>
   <dc:creator>Macedo, Rafael Luiz de</dc:creator>
  ▶<dc:description>...</dc:description>
   <dc:date>2012-11-21T11:56:02Z</dc:date>
   <dc:date>2012-11-21T11:56:02Z</dc:date>
   <dc:date>2012-11-21</dc:date>
   <dc:type>Dissertação</dc:type>
   <dc:identifier>http://hdl.handle.net/11077/804</dc:identifier>
   <dc:language>pt_BR</dc:language>
 </oai_dc:dc>
</metadata>
```

Figure 3. Response received of server of institutional repository [22].

The software XML2AIML, with the request response assigned to a variable of type BufferedReader, first parses the response that contains information on any file. In the case when it does not contain the response, the software will pass to the next step, which is to create the new information base; only after containing this new basis of questions and answers related to the point that no file was found in the repository within the information the user passed to the chatterbot.

C. Preset questions and answers

When the XML2AIML software parses and generates the new questions and answers on the information contained in the metadata of the XML response, consultations are held to a MySQL database; only information for pre-defined questions and answers is used in this work.

The database contains parts of questions and answers, which will be concatenated with certain kinds of information from the metadata that is contained in the XML to generate the questions and answers that will be used in the new AIML knowledge bases.

The goal of the use of a database in the software XML2AIML is to focus the new predefined questions and answers in a single location, using the database, to make it faster and to improve and increase the amount of these questions and answers that will be generated for each file found in the search of the repository.

This database also uses a table to focus the questions and answers, which are responsible for collecting information about the material that the repository user is seeking. These questions and answers have suffered many changes during the development of the software, to make the user clearly understand the questions and answers that the chatterbot will use during the dialogue.

D. Creating new knowledge base

The XML2AIML software parses the XML for the metadata for each file that is contained in the response. As seen in Sections 2 and 3, metadata is used to catalog every file, and, in the digital repository DSpace platform, the default DC metadata is used.

XML digital repositories are described according to the type of information about the file found, such as dc: title, dc: creator, dc: date, dc: type and other types of DC. DCs are passed by parameter to the method responsible for generating the questions and answers.

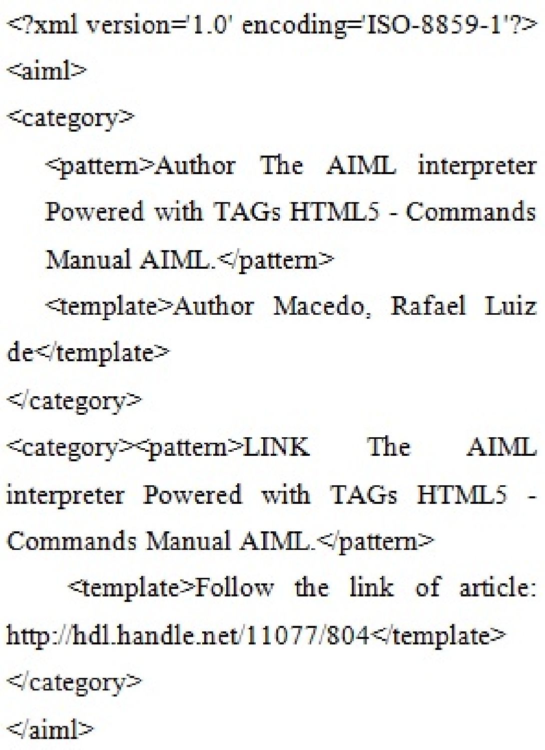
The parameter responsible for receiving the information about the file found in the digital repository performs a query to MySQL database software to pick up words that will be part of each set of questions and answers.

The questions and answers already concatenated with the information on the file are inserted automatically in the new AIML knowledge base; but, for each set of question and answer, XML2AIML inserts the elements that are part of the structure of a category to which the chatterbot can recognize and use these new questions and answers.

Each set of metadata represents a file found in the search in the repository. For each set, the software generates the same questions and answers that were used in the first set of metadata.

When the software has finished doing all the analyzing and applying the treatment of questions and answers, in all the sets of metadata, the software completes the new AIML knowledge base, with the element </aiml>.

Figure 4 presents a new base of questions and answers generated by XML2AIML software after analyzing the response received from the server.



```
<?xml version='1.0' encoding='ISO-8859-1'?>
<aiml>
<category>
    <pattern>Author The AIML interpreter
    Powered with TAGs HTML5 - Commands
    Manual AIML.</pattern>
    <template>Author Macedo, Rafael Luiz
de</template>
</category>
<category><pattern>LINK The AIML
interpreter Powered with TAGs HTML5 -
Commands Manual AIML.</pattern>
      <template>Follow the link of article:
http://hdl.handle.net/11077/804</template>
</category>
</aiml>
```

Figure 4. New base of questions and answers generated by software XML2AIML.

The software generated XML2AIML file is passed to the chatterbot, which submits the search result to the user from the repository. The chatterbot will continue with the conversation telling the user who found the material.

In the test scenario, the user askes questions about the material found by chatterbot, and the user terminates with the question that he wants to access the complete material. The chatterbot gives an answer the access link to the page of the material.

VIII. Conclusion and Future Work

The AIML markup language has a limitation in creating robotic conversational engines, limiting the number of questions these robotic engines can be asked. Due to this limitation, the robotic engines cannot keep a simple dialogue with a human for a long time because the questions are held outside the context of knowledge in which these robotic engines were developed, and cannot be answered.

Another problem is users accessing digital repositories of DSpace platform; the users encounter difficulties in being able to obtain the materials. Then, with the help of a chatterbot to interact with users and gather information about the desired materials, users can learn in a short time if the repository has a particular material or not.

The objective of this work was to encompass the knowledge of robotic AIML language engines with new bases of knowledge generated by the XML2AIML Analyzer software, using the data contained in the metadata of digital repositories of DSpace platform. In addition, we also improved the search of files the user can perform in these digital repositories.

As future work, XML2AIML will be implemented in repository server at the Center University of Euripides Marilia and it will be an available software to users of the repository of the university.

Another future work is to enable the XML2AIML to find files in others institutional repositories.

References

[1] http://aberto.univem.edu.br/ [retrieved: March 13, 2014]

[2] http://www.alicebot.org/aiml.html, [retrieved: March 13, 2014]

[3] Rachel C. V. A. Semantic Web: A Focused Analisy using metadata, Dissertation (Master of Science in Information) – Faculty of Sciences – University Estadual Paulista – UNESP, Marilia, 2005.

[4] http://aitools.org/Programd, [retrieved: March 13, 2014].

[5] E. Fusco, Conceptual data models as part of the process of cataloging: use of FRBR perspective in developing digital bibliographic catalogues, Doctor thesis, UNESP – Marilia, pp. 60-74, 2010.

[6] D. Hillmann, Using Dublin Core – The Elements, Making it easier to find information, DCMI, 2005.

[7] J. Milstead and S. Feldman, Metadata Projects and Standards, vol. 23, no. 1, pp. 32-40, 1999. ISSN : 0146-5422.

[8] http://repositorio.unb.br/, [retrieved: March 13, 2014].

[9] G. P. Ribeiro, Digital Geospatial Metadata, Workshop on non-conventional databases, Niterói, Brazil, 1995.

[10] E. Rich and K. Knight, Artificial Intelligence – Second Edition. São Paulo, 1993.

[11] S. Russell and P. Norvig, Artificial Intelligence – Translation of Second Edition. Rio de Janeiro, 2004.

[12] http://apprendre-math.info/portugal/historyDetail.htm?id=Turing, [retrieved: March 13, 2014]

[13] M. I. F. Souza, and L. G. Vendrusculo and Geane C. Melo, Metadata for the description of electronic information resources: using the Dublin Core standard, Ci. Inf., Brasília, vol. 29, no. 1, p. 93-102, jan./abr, 2000.

[14] R. Tansley, et al. The DSpace Institutional Digital Repository System: Current Functionality, Joint Conference on Digital Libraries, pp. 87-97, EUA, 2003.

[15] M. I. Tomael and T. E. of Silva, Institutional Repositories: guidelines for information policy, VIII ENANCIB – National meeting of research in information science, Salvador, Bahia, Brazil, 2007.

[16] S. L. Vellucci, Bibliographic relationships. International Conference On The Principles And Future Development Of AACR. Toronto: American Library Association: Library Association Publishing, pp. 105-147, 1998.

[17] A. M. M. Vieira and G. David, Digital Repository and process management Platform: an integrated architecture, Faculty of Enginneering, University of Porto FEUP, 2011.

[18] R. S. Wallace, The Element of AIML Style, © ALICE A. I. Foundation, Inc., October, 2003.

[19] http://www.alicebot.org/documentation/aiml-reference.html, [retrieved: March 13, 2014]

[20] S. Weibel, The Dublin Core: a simple content description model for electronic resources. Bulletin of the American Society for Information Science, pp. 9-11, Oct/Nov, 1997.

[21] http://www.ed.conpet.gov.br/br/converse.php [retrieved: March 13, 2014]

[22] http://aberto.univem.edu.br/oai/request?verb=ListRecords&from=2012-11-22T00:00:01Z&until=2012-11-23T08:00:01Z&metadataPrefix=oai_dc [retrieved: March 13, 2014]