Interactive Chatbot Using AIML

Madhumitha.S
UG Scholar, Department of computer science, Velammal Engineering College.
Email: madhu.miley@gmail.com
Keerthana.B
UG Scholar, Department of computer science, Velammal Engineering College.
Email: bkeerthana15698@gmail.com
Mrs.Hemalatha.B
Assistant Professor III, Department of computer science, Velammal Engineering College.

----------ABSTRACT----------

Human-Computer Interaction that characterizes dialogue between human and computer is gaining momentum in computer interaction techniques. This type of program is called a Chatbot or chatterbot. This paper presents a survey on the methods used to design a Chat-bot. The purpose of this paper is to give technical overview of those two languages and provide comparisons between them according following parameters:Ease of implementation and complexity of language, access to external resources, knowledge acquisition, linking to customized ontologies and to build chat-bot for a Mobile application. To achieve this,main concepts that is in Pattern Recognition are described because the AIML uses such theoretical framework in their semantic and syntactic structures.A design methodology will be explained based on a novel algorithm to automatically create AIML knowledge bases or datasets starting from a frequently asked question free text file and a glossary of terms to a completely functional query response dispenser.

IndexTerms - **AIML, NLP, Ontology, ChatScript, Chat-bot**

I. INTRODUCTION

In the present globalized society, the increasing development and spread of Information Technology and the Internet have led to the creation of distinct ways of communication among users in virtual environments. The cognitive interfaces provide interaction between human and machine. The Graphical User Interface is based on navigation systems that use (i) hypertext or (ii) option selection with buttons/menus. These techniques require an additional cognitive effort by the users, since the natural language processing is the usual way of communication between humans. Hence more appropriate cognitive interfaces allow the foundation of dialogues in natural language and also make the human computer interaction more attractive and powerful in terms of empathy and interactive ability.

In the context of Ubiquitous Computing, its goal is to effectively integrate human beings and technology/machine, research has been made regarding the development of the “natural interfaces”. These interfaces make the communication more intuitive, because the idea is to use usual forms of interaction such as natural language, vision and gestures. Among the possible forms of natural user interfaces, this paper highlights the usage of chatbots or computer systems that are designed to simulate a conversation with humans. In a chatbot, conversation usually occurs by the exchange of text messages (in a chat environment), and the chatbot is organized to provide features that allow the usage of natural language, such as dealing during ambiguous situations and context-based message analysis.

Chatbots currentlyhave been used for various purposes. For example, the entertainment chatbots are designed to entertain and to amuse the user while maintaining a coherent conversation. For instance, the chatbot Susan is a virtual human that represents Dr. Susan Calvin from Isaac Asimov's Robot Serie. There are chatbots for mobile devices, such as the ChattyBot for Android platform. Penny is a virtual assistant to solve problems related to products and/or services offered by the Heat Gas Company Ltd. Also, educational chatbotsare used as an alternative to the teaching-learning process and can be used to supplement the information provided in the classroom and to clarify any doubts. For instance, the Einstein chatbot is responsible for to teach Physics and was developed by Artificial Life Company.

Along with the evolution of the chatbots, researchers noticed that the structure of chatbots modelled using the Natural Language Processing would present itself as a complex task. In order to help in such a task, dedicated technologies for chatbot making has been developed. Among these technologies, Wallace in collaboration with free software developers, communities can be noticed. From 1995 to 2000, the Artificial Intelligence Markup Language (AIML) is created, based on the concepts of Pattern Recognition or Pattern Matching technique. It is applied to natural language modelling for the interaction between humans and chatbots that follow the stimulus-response approach. For this purpose, a set of possible user inputs is modelled and for each one of these query, pre-programmed answers were built to respond to the user. The ALICE (Artificial Linguistic Internet Computer Entity) chatbot was the first to use the AIML language and interpreter. In ALICE, the AIML technology was responsible for pattern matching and to relate a user input with a response of the chatbot’s Knowledge Base (KB).

In the literature that presents the AIML concepts, there are tutorials that introduceseach concepts and present the language concepts in detail. These two options are not suitable for people who are new to AIML because they can't balance the amount of application and theory. Additionally there were other works that focuses on the usage of Pattern Recognition techniques, and the AIML language itself is addressed as a secondary topic. AIML language tutorial is presented in this work and concepts of Pattern Recognition are considered, but the main idea is to have a reference guide for AIML language's initial studies and to provide help in creation of chatbots.

This paper is organized as follows. Section 2 presents a brief history of the development of chatbots, followed by an introduction to Pattern Recognition. The next section introduces the AIML language and its main statements. Application of AIML language is discussed in section 4 for the treatment of conversations limitations between people and machines. In Section 5 computational tools used for the development of chatbots are classified and described. Lastly, Section 6 presents the conclusions of the work.

II. BACKGROUND

In 1950 Alan Turing published an article proposing the Turing test as a criteria of intelligence. This criteria evaluates the ability of a computer program that communicates in real time with a person to effectively mimic human interaction, enough to be indistinguishable from a human conversational partner in real life. Joseph Weizenbaum's program ELIZA that was published in 1966 could fool users into believing that they were conversing with a human. Authors agreed that the best way to facilitate Human Computer Interaction (HCI) is by allowing users "to express their interest, wishes, or queries directly and naturally, by speaking, typing, and mentioning it".

Numerous chat-bot architectures have been developed since then. Some of them are: MegaHAL , CONVERSE , ELIZABETH, HEXBOT and ALICE. A.L.I.C.E. which is abbreviated as the Artificial Linguistic Internet Computer Entity, was first implemented by Wallace in 1995. Alice's knowledge about English conversation patterns is stored in AIML files as log results. AIML is a derivative of Extensible Mark-up Language (XML). It was developed by Wallace and the Alicebot free software community from 1995 onwards to enable people to input dialogue pattern knowledge into chat-bots based on the A.L.I.C.E. open-source software technology.

A.L.I.C.E has advanced pattern recognition techniques and a good source of datasets.The third generation of chat-bots is based on the concepts of Pattern Recognition or pattern matching technique. It applies natural language processing for the dialogue between humans and chat-bots that follow the stimulus response approach. For this purpose, a set of possible user inputs is modelled and, for each one of these sentences (stimuli), pre-programmed answers were built to be shown to the user.

The ALICE (Artificial Linguistic Internet Computer Entity) chat-bot was the first to use the AIML language and interpreter. In ALICE, the AIML technology is responsible for pattern matching and to relate a user input with a response of the chat-bot Knowledge Base (KB).

III. AIML

3.1 Artificial intelligence markup language 1.0

The main aim of the AIML language is to simplify the modelling process of dialogue. Itgives access to stimulus-response. In addition tothat, AIML is an XML-based markup language. AIML works by defining a class object that is responsible for modelling the patterns of conversation. It is the most-used chat-bot language due to its simplicity, ease of learning, ease of implementation and the availability of pre-authored AIML sets. It is a simple word pattern-matcher that generates a response based on a query.

The AIML robot responds according to connection between the questions set by the user and knowledge located in AIML files.

AIML has its advantages and disadvantages. Some of the advantages are: Easy to implement and learn,user-friendliness and simplicity of the system of dialogue, the use of XML for the formal, computer readable representation.

Some of the disadvantages are that the knowledge is presented as an instance of AIML files. If knowledge is created based on data collected from the internet, it will not be automatically updated and must be updated periodically. Original AIML has no extension possible.

AIML has relatively poor matching patterns and it is very difficult to maintain it. Though the content is very easy to enter, large amount of content which must be manually entered to create a functional chat-bot poses as a problem.

AIML 2.0 fits the requirements of new generation of chat bots and hence it will be focussed in this paper.

3.2 Artificial intelligence markup language 2.0

AIML 2.0 is backed up by the technological and social changes that is taking place in the mobile era. the ALICE A.I. Foundation released the AIML 2.0 specification.

1) To provide Ease of implementation and complexity of language
AIML 2.0 is an initiative done to address the shortcomings, while balancing the original goal of keeping the language as simple as possible. The AIML 2.0 specification is designed to be backwards-compatible with

the AIML 1.0 and earlier standards, in that way it preserves the simplicity of the original language.

One of the very important improvements seen in AIML 2.0 is pattern matching. In addition to the wildcards _ and * already displayed, # is now available. These matches on 0-N words allow more compact spellings of frequently used matching rules. Previously, in the absence of 0-N wildcards, several patterns were necessary to intercept the cases where the word is at the beginning or end of the sentence. The sets and maps are also introduced in version 2.0 and itsused to integrate knowledge from external files.

Sets are set of objects, which are compiled and stored line-by-line as plain text in embedded file. The 'set' tag is used to access the set by their filename in the pattern. In the case of hit, it can be included in the template with the already known star tag. Sets and maps have a higher priority than * when pattern matching, but the remaining operators have priority.

Maps consists of assignments from one term to another, for example, a list of countries and their capital cities. In sets there is a powerful possibility to define a lot of knowledge and a large amount of interactions with the chatbotin a small space. To achieve the same functionality in AIML 1.0, a separate category would have to be created for each entry.

In AIML 2.0 explicit loops were introduced with the loop tag. The syntax is a bit unconventional but in concept it is just a mixture of couple of Do-While Loops and GoToStatements.

2) Access to external resources

AIML 2.0 introduces <obb> tags - out of band commands for device Virtual Assistant responses. Those tags enable users toSend SMS, Send email, Dial, search, maps etc. AIML 2.0 also enables access to various websites like Wolfram Alpha, Weather Service, Shopping sites, Trueknowledge.com, Answers.com, other chat-bots and Netbase. Netbase is a semantic web database. Netbase knowledge Graphconsistsof over 600 million nodes and statements gatheredfrom the Internet's sources, including Freebase, Wikidata, Yago, Wordnet,DBPedia, as well as many custom GraphDBs. Netbasehas the Wordnet ontology and it also has built-in relations so that you can quickly query with synonyms, transitive class hierarchies, part meronyms etc.. In AIML 2.1 there is a facility to create ontologies that enables the chat-bot to make use and reason with knowledge. AIML can extract domain knowledge from semantic Web ontologies using a scripting language called OwlLang and to store new knowledge obtained from the conversations in the ontologies. Listing 1 Shows code for accessing external resources.

Listing 1.

```
<category>
<pattern>WHAT IS THE
WEATHER</pattern><template>
<sraix service="pannous">WHAT IS THE WEATHER
</sraix>
</template>
</category>
```

The category tag defines a question-response pair. The question is defined within the pattern tag pair (<pattern></pattern>) and the response is defined in the template tag pair (<template></template>). The chat-bot response could be from categories within its own knowledge base, those of other bots or other knowledge sources. The *sraix* element allows a chat-bot to call categories that exist within another chat-bot, and return response as if it was its own. In the code shown the chat-bot reports weather information that is retrieved from external sources provided by the pannous web service.Panonus web service aggregate knowledge from DbPedia, WordNet, Weather Service, Shopping sites etc.

3) Knowledge acquisition

The *learn*tag allows the user to generate new category blocks from the conversation. This powerful functionality allows the users to teach the chatbot new information. Categories generated by the learn element are stored in the memory, and it isonly accessible with the client name that was used to create them.

There is a tool called Pattern Suggester that is part of Program AB. It is the most recent reference implementation of AIML 2.0. Pattern Suggester supports in automating the process of creating new patterns through a type of unsupervised learning.

The *learnf* element is identical to the learn element syntactically, however, the generated category is written to an AIML file that you may specify with the property learn filename. Learning example is given in the Listing 2 below.

Listing 2. AIML 2.0 LearningExample

```
<category>
<pattern>* LIKES *</pattern>
<template>
    I will remember that you like coffee.
    <learn>
      <category>
      <pattern>WHAT DO *
      LIKE</pattern><template>You like
      <star/></template></category>
    </learn>
  </template>
  </category>
```

The category tag defines a question-response pair. The keyword LEARN instructs chat-bot to create a category for the question-response specified within the <learn> element. It is done to be able to provide response to such pattern subsequently. <star/> tag is used to retrieve variables that is denoted by * and it is supplied by the users to the chat-bot.

4) Linking to customized ontologies

Program AB is a Java application. It is structured to enable developers to easily extend AIML with custom tags. Program AB uses AIML 2.0 features including the capacity to connect to remote bots and web services through the new tag.

Program AB also uses some memory optimization techniques that make it possible to run a chatbot on a mobile device or an embedded system. It serves as a reference implementation of AIML 2.0.

Listing 3. Querying the customized ontology

```
<category><pattern>SPEEDOF *</pattern><template>
<get var="? x">
<tuple><first><select>
<q><subj><star/></subj><pred>hasSpeed</pred><obj>
?x</obj></q>
</select></first></tuple>
</get>
</template>
</category>
```

In the code that is shown, the SPEEDOF keyword is defined in the <pattern> tag.It triggers the query defined in the <template> tag for the entered entity variable that is denoted with *. Query variable is initialized in the <getvvar="? x> tag. In tag <q> query code is presented. Query returns *hasSpeed* attribute value of the entity variable entered in the <pattern> tag. Entities and their attributes are defined in a external file named *example.txt* and it is shown on Listing 4.

Listing 4.example.txt file

```
CEETAH:hasPurpose:to hunt
wildlife
CEETAH:hasSize:7
CHEETAH:hasSpeed:10
CHEETAH:hasSyllables:2
CHEETAH:isa:Animal
CHEETAH:isa:Wild Animal
CHEETAH:lifeArea:Physical
```

5) Building a chat-bot for a Mobile application

The CallMom app for Android was the first mobile assistant app to use multiple chat-bot personality choices and a sophisticated learning feature. In 2013, the successor app CallMom became the first virtual assistant to integrate the AI function directly on a mobile device. The AIML 2.0 includes several features that support the development of mobile applications with the chatbot functionality.

IV. CHATSCRIPT

ChatScript is a scripting language that is designed to accept the user text input and generate a corresponding text response. Chat proceeds in volleys like tennis. The program takes as input one or more sentences from the user and outputs one or more sentences back as a response. ChatScript is a system for manipulating natural language. Hence it isnot just for building a chabot. ChatScript starts by transforming the input words using substitutions files. It has files for texting, spellings, common spelling mistakes, abbreviations interjections mapped as speech acts. These are all stored as live data. This means that they are not stored in the dictionary, but they get loaded on start-up of the application. The trailing punctuation mark is removed, with bits set to reflect punctuation status like question, statement, exclamation.

1) Ease of implementation and complexity of the language

ChatScript is both a scripting language and an engine. Implementation can be more difficult than AIML.

Fundamentally a script is a series of rules. A full rule usually has a label,kind, a pattern, and a corresponding output.

Rules can be constrained to statement inputs or when the chatbot takes control of the conversation and wants to volunteer something. Rules are bundled into a set of collections called topics.

A Topic can invoke other topics. The procedure to exactly process an input is controlled by a control script which by itself is a topic.

Pattern is a set of more specific conditions. It tries to match words of the current input sentence, but it sometimes considers the prior history of the conversation.

The output is what this rule does if it is allowed to execute. Since the overall goal is to generate a response, the simplest output is merely the words to say or the corresponding reply to the query.

More complex outputs can perform conditional execution, loops, function calls, etc. The system normally runs the rules in a specified order until it not only passes pattern restrictions, but also actually generates output to the user. Once user receives the output,the system is done.

2) Access to external resources

WordNet is a lexical database of the English language. It is created and maintained at the Cognitive Science Laboratory in Princeton University. It is used for spell-checking. It groups words into sets of synonyms or synsetseach expressing a distinct concept. WordNet provides short and general definitions. It records the semantic relations between synsets. ChatScript has the ability to read structured JavaScript Object Notation(JSON) data from websites and Postgres support for big data or large-user-volume chatbot. ChatScript usually has several routines for calling external resources. Script code that is shown on Listing 5 enables acquiring and processing data from the URL (unified resource location) or Web.

Listing 5. Requesting data from the Web

```
^tcpopen (kind url data 'function)
```

3) Knowledge acquisition

ChatSript has an option for storing or memorizing certain kinds of information and uses that information in the future conversations. ChatScript memorizes parts of chat and stores it in long-term data as variables and facts. When the variable is once set, its written with the user

data unless it is explicitly erased. Variables can be used in the entire chat. The variable also remains for other rules to use.

4) Linking to customized ontology

ChatScript can create facts—subject, verb, object triples— and to organize them into tables. The data saved as facts and in tables may then be recalled using queries. ChatSriptcan connect to external ontologies and can acquire knowledge from the ontology and write this knowledge into local concept files or create a custom dataset.

5) Possibility to build chatbot for a mobile application

To build a mobile app, ChatScript should be integrated within another program. This means that ChatScript is not the main program. In this scenario, the main program is controlling the mobile application and invokesChatScript for conversation or control guidance. Depending on mobile application platforms, the ChatScript memory should be reduced for efficient functioning. The full dictionary may take a storage limit of 25MB. For mobile applications, dictionary size should be minimized.

V. COMPARATIVE REVIEW OF AIML AND CHATSCRIPT

The goal of this study was to identify the best technology for building personal chat-bot. Choose the chat-bot languages that are based on the open source platform. Next step is to compare the selected languages as per the requirements that are defined.

Table-1 compares AIML and Chatscript

Table 1.Comparative review AIML and ChatScript.

	AIML 2.0	**ChatScript**
Ease of	Very simple process of	Implementati on
implementation	dialogue modelling	is more
		difficult than
		AIML
Complexity of	Balancing between	Complex,
language	keeping the language	system for
	as simple as possible	manipulating
	and the technological	natural
	and social perquisites	language and
	of mobile era.	conversation
		agents
Connection to	Freebase, Wikidata,	WordNet,
external resources	DBPedia, Yago,	ConceptNet
	Wordnet	
Learning	Defined syntax for	Yes, variables
	learning new	and facts
	patterns	
	and templates.	
Personal ontology	JENA library	Yes, script code
implementation		
Mobile App	YES	YES, but with
		restrictions

ChatScript is created to provide natural language understanding capabilities for characters in games, but has also been used for the chat-bot. ChatScript is meaning-based and have supporting sets of words called concepts to represent synonyms.

If the user goal is to create the illusion that chatbot understands the user, this means that the user will try to minimize instants of situations where the chatbot says a response that is completely unrelated to the user query and maximize the cases when the chatbot responds completely appropriately, then the ChatScript is a right solution.

AIML is a widely-used markup language for creating chat-bots.

The <topic> tag can now be set on a category to make it easier to add categories to topics. Zero or more wildcards like ^ and # pattern match zero or more words. Pattern priority - $ pattern marker map a pattern wordand has highest priority. Attribute tags - Template tag attribute can be set using a sub-element.

<set> tag evaluates a pattern based on words that are defined on a predefined set.

<map> tag allows the lookup of the element value in a predefined mapping and returns the mapped value.

Condition patterns - wildcards can be used to provide default conditions in condition values.

<loop> tag is used to loop a conditional statement.
<var> attribute for variables scoped to a category.

Remote requests:

<sraix> tag is used to make a remote request to another bot instance, or service.

<normalize> and <denormalize> tags are used to convert special character into words, and back again. Date formats - New formatting options for dates.

<request> tag returns the users previous input request.

<response> tag returns the bots previous response.

<learn>, <learnf>, and <eval>are all learning tags that are used dynamically train a chat-bot with new responses.

<explode> tag is used to split a word into its constituent characters.

<oob> stands for out of band tag to support client and mobile device commands.

Program AB implements the features in AIML 2.0 including Sets and Maps, Zero+ wildcards and capability to connect to the remote bots and web services through the new tag.

Program AB implements some memory optimizations and makes it possible to run a chatbot on a mobile device or an embedded system. Program AB has already been tried on Android phones and tablets.

If the user goal is ultimately to build a customized chatbot that fulfils technical requirements for connections among people, fetch information, software and follow the social changes caused by mobile era, then AIML 2.0 is a logical choice.

VI. CHATBOTS TAKE TIME TO MATURE

No matter what framework one uses, designing a chatbot is an iterative process. One must not underestimate the time and resources that are needed for a bot to ripen or grow to its full potential for providing a good performance. Despite the marketing hype around chatbots, it is not a technology that operates out of the box. "A lot of pundits say that chatbots aren't good; that's only because people have decided to make them not good," noted one of the Aspect Software's customer strategy executives. "You might have to put some effort into it." Collecting data is part of it, but you must also expect to invest time in tuning performance once real users are involved. This might create a challenge if you want to evaluate multiple platforms against each other; it could be more useful to watch the literature as side-by-side comparisons on accuracy and effectiveness are executed and reviewed.

Table 1. Features of popular bot creation and distribution services.*

		Distribution services					Creation services			
		User base				Monetization (bots collecting payment from users)	Software foundations			
Platform		Size	Cost for users	Interaction mechanisms	Discovery		Cost or license	Tools or technologies	Platforms	Developer community
Slack (slack.com)		6 M daily & 2 M paid users (2017)	• Free • Paid	• Text • Commands • GUI	App directory	No	Free	• Web API • RTM API • App blueprints	Slack	Yes
Microsoft	Teams (teams.microsoft.com	125 K teams (2017)	Paid	• Text • Commands • GUI	Bot directory	No	Microsoft Bot Framework			
							OSS	• Bot Builder • Bot Connector • Service integrations • Azure Bot Service • .NET (C#) SDK • Node.js REST SDK	Multiple (Bot Framework Portal)	Yes
	Skype (www.skype.com)	3 M monthly users (2017)	• Free • Paid	• Speech • Text • Commands • GUI	Bot directory	Yes				
HipChat (www.hipchat.com)		90 K paid users (2017)	Paid	• Text • Commands • GUI	Marketplace	?	No	No	No	No
Messenger (www.messenger.com)		1.2 B monthly users (2017)	Free	• Text • Commands • GUI	• Facebook pages • Bot Explorer • QR codes	Yes	Free	• WebHooks • APIs • Design Kit • Wit.ai • RTM	Facebook	No
WeChat/Weixin (web.wechat.com)		889 M monthly users (2016)	Free	• Speech • Text • Commands • GUI	QR codes	Yes	No	No	No	No
Telegram (telegram.com)		100 M monthly users (2017)	Free	• Text • Commands • GUI	Bot store	Yes	OSS	• Bot API • Telegram API • BotFather	Telegram	No
Kik (www.kik.com)		15 M monthly users (2017)	Free	• Text • Commands • GUI	• Bot shop • QR codes	No	Free	API	Kik	No
Other technologies		**Bot-hosting platforms** • Alexa • Echo • Cortana • Line • Android • Discord • Cisco Spark • Messages • Viber • Intercom • Google Allo • Twilio • SMS • Web		**Extending interactions** • Google Cloud • Watson Conversation • Alexa Voice • Luis.ai • MindMeld	**Bot directories** • BotList • ChatBottle • Botwiki	**Payment SDKs** • PayPal • Stripe	**Bot-building tools** • Api.ai • Pandorabots • Chatfuel • Rebotify • Botkit • Gupshup • OnSequel • Flow XO • Botsify • BotMock • BotMan			**Online communities** • Botmakers • Chatbots.org • chatbotsmagazine.com • Stack Overflow

VII. CONLUSION

In this paper, we have given a technical overview of the two most used open source languages for building a chatbot. We have also provided comparisons between those two languages according to various parameters. AIML 2.0 specification fulfills the need for new features but at the same time, it keeps AIML as simple as possible, especially for non-programmers thus making it a good beginner language in the creation of chatbots.

Theconsumers these days expect to be able to find the information that they are looking for on online quickly and easily. But when a business can't provide that type of experience they end up becoming frustrated. Chatbots are created to ease these frustrations by providing the on-demand real-time approach that consumers are seeking for.

The top three potential benefits of chatbots that consumers reported are: 24-hour service (64%), instant responses (55%), answers to simple questions (55%) When compared with other business communication channels, chatbots scored the second highest when it came to consumers expecting responses instantly, only losing out to online chat. By using chatbotsin combination with online chat various businesses can now deliver a level of real-time service that they had been unable to achieve using technology on its own.

REFERENCES

[1] Khanna, B. Pandey, K. Vashishta, K. Kalia, B. Pradeepkumar and T. Das, "A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence*", IJUNESST,* vol. 8, no. 7, pp. 277-284, 2015.

[2] TURING, "I.—*COMPUTING MACHINERY AND INTELLIGENCE", Mind*, vol., no. 236, pp. 433-460, 1950.

[3] M. Bruno Marietto, R. Aguiar, G. Barbosa, W. Botelho, E. Pimentel, R. Franca and V. da Silva,

[4] "*Artificial Intelligence Markup Language: A Brief Tutorial", International Journal of Computer Science & Engineering Survey,* vol. 4, no. 3, pp. 1-20, 2013

[5] Batacharia, B., D. Levy, R. Catizone, A. Krotov, and Y. Wilks. "CONVERSE: *a conversational companion." In Machine conversations, pp*. 205-215. Springer US, 1999

[6] M. McTear, C. Zoraida, and G. David. *"Conversational Interfaces: Devices, Wearables, Virtual Agents, and*

[7] *Robots." In The Conversational Interface,* pp. 283-308 Springer International Publishing, 2016

[8] Shawar, Bayan Abu, and Eric Atwell. "*Chatbots: are they really useful?." In LDV Forum*, vol. 22, no. 1, pp. 29-49. 2007

[9] Hutchens, Jason L.; Alder, Michael D. (1998), "Introducing MegaHAL" (PDF), NeMLaP3 / CoNLL98

[10] *Workshop on Human-Computer Conversation,* ACL (271): 274

[11] Lundqvist KO, Pursey G, Williams S, "Design and implementation of conversational agents for harvesting

[12] feedback in eLearning systems". In: Hernandez-Leo D, Ley T, Klamma R, Harrer A (eds) Scaling up learning for sustained impact. Lecture notes in computer science, vol 8095, pp 617–618., 2013

[13] RamonLópez de MántarasBadia, ARBOR Ciencia, Pensamiento y Cultura, Vol. 189-764, no 2013, a086

[14] ISSN-L: 0210-1963, doi:

[15] http://dx.doi.org/10.3989/arbor.2013.764n6009)

[16] Software bots IEEE paper by Carlene Lebeuf, Margaret-Anne Storey, Alexey Zagalsky

[17] HEXBOT chatbot website. http://www.hexbot.com/

[18] Richard S. Wallace, The Elements of AIML Style, ALICE A. I. Foundation, Inc. March 28, 2003

[19] Wallace, R. S. (2014a). AIML 2.0 Working Draft, https://docs.google.com/document/d/1wNT25hJRyupcG5 1aO89UcQEiG -HkXRXusukADpFnDs4/pub