International Journal Of Computer Science And Applications Vol. 1, No. 1, June 2008
ISSN 0974-1003

An Approach to Deal with Ambiguity in Requirement Elicitation

Md. Rizwan Beg
Dept. of Computer Sc. & Engg.
Integral University, Lucknow, India
+91-0522-2890812
rizwanbeg@gmail.com

Dr. Qamar Abbas
Dept. of Computer Sc. & Engg.
Integral University, Lucknow, India
+91-0522-2890812
qrat_abbas@yahoo.com

Alok Joshi
Dept. of Computer Sc. & Engg.
Integral University, Lucknow, India
+91-0522-2890812
joshialok@rediffmail.com

ABSTRACT
Requirements elicitation has received little attention in past and it has been ignored by most of the software engineering research community, the requirements are often ambiguous, incomplete, inconsistent & informal. Out of which the problem of ambiguity at time of requirement elicitation is difficult to deal with. So in order to solve the problem of ambiguity between the stakeholder and client at the time of interview, this paper proposes a method which is to be used, as soon as a new requirements is expressed by the client in natural language the system converts it to the set of production rules, where each rule is corresponding to the particular context. Then the variables of the each production are converted into bipolar representation, which is used as single vector denoting the sentence of the interview session. The stakeholder brain is considered as associative networks of the bipolar vectors. Upon applying the sentence to the network if we get the same output vector as the input vector then there is no ambiguity in the requirement just expressed.

Categories and Subject Descriptors
D.2.1 [**Requirements/Specifications**]: Elicitation method.

General Terms
Algorithms, Theory

Keywords
Ambiguity, requirement elicitation, stakeholder, client, interview, natural language, grammar, production rules, associative nets.

1 INTRODUCTION
According to Standish research around 31% of projects are canceled before they can be completed and around 52 % of projects costs around 189% of the actual cost just around 16% of projects get completed within time and in budget, it was found that incomplete and changing requirements are the big concern so requirement engineering if not tackled seriously can create a big problem [15].

The success and failure of the software product depends on whether it meets the requirement for which it has been built or not. According to [5], "Requirement engineering is the process that identifies whether software is a success or a failure by identifying the stake holders and their needs and then documenting it in form of agreement for the further implementations".

Following activities are involved in the requirement engineering phase [9]-

(1) Requirements elicitation, (2) Requirements analysis, (3) Requirements specification and (4) Requirements validation.

The goal of requirements engineering is to produce good quality requirements specification as that can only minimize the project cost and can assure the on time delivery of the product. According to [IEEE 84] good software requirements specification must be: unambiguous, complete, verifiable, consistent, modifiable, traceable, and usable during operations and maintenance.

As all the above mentioned characteristics are important and may affect the production of quality requirements, according to [6], "ambiguity can occur at three levels namely requirements elicitation, requirements documentation, and requirements validation". We are here concentrating ambiguity removal on the requirement elicitation part of the requirement engineering phase.

As "Requirements elicitation is the process of discovering the requirements of a software project" [2].Some researchers says, Requirements elicitation is related to the gathering of requirements [12].

As there are various ways to gather the requirement and various persons are involved in it, some are using the system , some have financial interest, some maintains it and some pays for the system they all are named as Stakeholders [7] so generally they can be any of the following customers/sponsors, users, developers, quality assurance teams, and requirements analysts [5]. Here we are considering the Stakeholder as a developer or a requirement analyst.

There are various ways for gathering the requirements, for the purpose, various elicitation techniques are used, these techniques are the methods used by requirement engineers to determine the needs of the stakeholders [3] [4], the criteria for the selection of the particular technique is not fix, it can be the favorite technique for requirement analyst or it can be a favorable in particular domain or application [1]. The elicitation techniques can be classified into the four categories- Traditional Techniques,

Collaborative techniques, Cognitive techniques, and Contextual approaches [5].

This paper deals mainly with the Interview technique which falls under traditional approach. It is the technique which is used to collect the information from stakeholders in a project. As interview technique is purely a traditional approach, in which at one side of the table, there is a stakeholder or many stakeholders sitting as an interviewer or interviewers and client as an interviewee, and natural language is the mode of conversations and generally context free questions are asked with the client. But as the mode of conversation between stakeholder and client is a natural language so ambiguity may be introduced in the interview session and as stakeholder pen down the requirements, there might be a possibility that this ambiguous requirement can move further to the specification phase and can create a conflict at later stages of the project between a client and stakeholder so in the next section we have devised the approach to remove the ambiguity at elicitation level of requirement engineering, the idea of the approach was influenced by [8],in their paper they have suggested the two methods for the removal of ambiguity at elicitation level i.e. “Context must be established and agreed by stake holders” and “The requirement engineer replies the user’s statements in his own words, it is an effective way so that customers and users can spot their own ambiguities”.

But as we want to explore the idea and suggested the system that adds a feature of cognitive and contextual approaches in the traditional approach of elicitation.

So here we have tried to built the system that concentrates on the structure of brain of the stakeholder, that has been modeled mathematically by many researchers [10], as an associative net or as a matrix model [11] [13] and we have tried to show, how the context and association affects the ambiguity, as ambiguity is a big issue in the natural language so in the next section we are concentrating on the smaller section of natural language grammar.

2 PROPOSED WORK

In the real interview scenario, we assume that the views of all the stakeholders are integrated and a representative for all the stakeholders is appointed. There could be many types of ambiguity related to natural languages as discussed above.

The process of ambiguity removal in an interview goes according to the following steps

The client tells the requirement and stakeholder listen it.

The stakeholder replies the statement to the client based on his memorization capability and present and past contexts.

If the reply of the stakeholder satisfies the client then there is no ambiguity otherwise there is an ambiguity.

Let there be a system that performs the same function like the stakeholder and follows the above process for removing ambiguity. We assume that the sentence system interprets are of declarative type, that follows a particular type of structure, here we are introducing the method for the removal of the one type of ambiguity that exist between singular subject and multiple pronoun and we are exploring that how this particular type of sentence that is delivered from client to the stakeholder that can be rightly or wrongly interpreted by the stakeholder’s memory so our work is in two parts, according to the first part, take a requirement and with the help of speech to text converter it is converted to the text and then assign the structure to the particular requirement of the client and prepare the grammar for it and in the second part relate the structure of requirement with stakeholder’s memory and check for the ambiguity.

In the First Part, the method is as follows

Step1:- The client tells the requirement and system listen it.

Step2:- The speech to text converter, converts the speech to the text (Assuming the conversation is in Declarative Sentence form)

Step3:- Identify the Verb, then Subject, Object and other verb patterns of the sentences and concatenate them, which results a new text.

Step4:- The new text follows any of the 16 forms(From DS1 to DS16) of verb patterns which is already stored in the system, as stated in Table1 [14].

Step5:- After identifying the form, grammar is prepared for the requirement [16].

Output of the First phase is a production rule corresponding to the requirement.

Table1:- The possible structure of the English Declarative type of sentences, according to verb patterns in English Language [14].

Structure (Excluding the case of Interrogative Sentences)	Verb Patterns				
DS1	S	V			
DS2	S	V	C s		
DS3	S	V	O d		
DS4	S	V	O i	O d	
DS5	S	V	O d	Preposition	Preposition Object
DS6	S	V	Noun/Pronoun	Adjective	
DS7	S	V	Preposition	Preposition Object	
DS8	S	V	To infinitive(as object)		
DS9	S	V	Noun/Pronoun	To infinitive	
DS10	S	V	Gerund		
DS11	S	V	Noun/Pronoun	Present Participle	
DS12	S	V	Noun/Pronoun	Bare infinitive(Without To)	
DS13	S	V	Noun/Pronoun	Past Participle	
DS14	S	V	Noun/Pronoun	To be	Complement
DS15	S	V	That clause(as object)		
DS16	S	V	Noun/Pronoun	That Clause	

In the Second Part our method, adopts matrix model proposed by [11] [13]. The standard term like items, context, association and cue in the context of elicitation, interviews are used.

In the Second part the method is as follows:-

Step1:- Initialize the matrix between the context and items with a cue and make it as a pre-existing matrix.

Step2:- Convert the production rule obtained from the first part into a vector form.

Step3:- Initialize the context vector in which the above vector of requirement has occurred.

Step4:- From the requirement vector take the cue and find that cue in the pre-existing matrix of memory, if it is found then get the information in form of a matrix (related to items and context).

Step5:-Multiply the input requirement vector with a matrix obtained in step 4

Step 6:- The Output of the step 5 will be in a vector form, convert it into natural language form with the help of grammar.

Step7:-Ask the client to analyze the output in natural language and make the decision whether it makes the same sense that was earlier told by the client, if yes then there is no ambiguity otherwise there is an ambiguity.

3 ANALYSIS AND RESULT

In the First Part:-

Step1:-Let the Client tells the requirement to the system as "Everybody must bring his/her notebook"

Step2:- It is converted from speech to the text form through the speech to text converter; the text is "Everybody must bring his/her notebook".

Step3:-After finding the verb patterns through rules, they are stored in the following variables like

<Singular Pronoun/Subject>← Everybody

<Modal Verb>← must

<Main Verb>← bring

<Possessive Pronoun (Singular)>← his/her

<Object>← notebook

Now we concatenate the above variables which give the new text.

Step4:-Compare the new text with the stored pattern as in Table 1, it belongs to DS2 category.

Step5:- Grammar for the above sentence is like

<SingularPronoun>→ I/You/He/She/It/Everybody/Anybody/……

<Modal Verb>→ Can/Could/Should/May/Might/…..

<Main Verb>→ Bring/…..

<Possessive Pronoun (Singular)>→ His/Her/…..

<Possessive Pronoun (Plural)>→ Their/……

<Object>→notebook/…

So the Output of the First part for the requirement is

"Everybody must bring his/her notebook" will be in the form-

<SingularPronoun><ModalVerb><MainVerb><Possessive

Pronoun (Singular)><Object>

In the Second Part [11] [13]

Step1:- Initialize the pre-existing matrix of the past memories same as given in the Figure1.

Ca, Cf...	M11	Cd...	M12	Cg...	M13	.....
Cb, Ck...	M21	Ce, Cb, Ck.....	M22	Ch...	M23	.....
Cc....	M31	Cf....	M32	Ck...	M33	.....
.....	.....	.....	.....	.....	.....	.....

Figure1:- Arrangement of Preexisting memory

Where any individual matrix Mi j is in the form of m * n bipolar vectors of either requirement item with requirement item or of requirement item with context vector, say Mi j is in the form

$$\begin{bmatrix} 1\ -1\ -1\ \ 1\ \ 1 \ldots \\ 1\ \ 1\ \ 1\ -1\ -1 \ldots \\ \ldots\ldots\ldots\ldots \end{bmatrix}$$

To associate the particular variable with the memory the word in terms of variable must be stored in the pre-existing memory then only the system can recall that word, for that we are choosing the static variable 'cue', as in the above example, cue is "notebook", which can be associated with the context where it happened or it can be related to any of the item also, that cue relate the particular variable/bipolar element to particular matrix that make sense with respect to context or between set of items i.e. those items which have built the matrix.

The cue is stored with respect to each element of the bipolar vector/variable of all the rules or events that happened in past.

Like for a vector Vi = {1 -1 -1 1 1} say the last element of the set of vectors is associated with the variable "notebook" and the item "Rohan has won the notebook" that came in past as a requirement or event in the system that means cue "notebook" is associated with "Rohan" in past as a singular pronoun reference.

Step2:- The requirement which is in form of rule, is converted to bipolar form (1 -1 -1 1 1){Assuming that any variable in a structure of the sentence can take at most two forms of the values, like Singular Pronoun can take only two values i.e. Everybody and I}.

Step3:-As the context means, in which the item has occurred, suppose we are taking the context "classroom" which is related to the notebook, the vector for the context classroom may be in the form {0 0 0 0 1}, now the context vector is multiplied by the above item vector then the matrix obtained in the multiplication gives the information that notebook is related to the plural thing.

Step4:-We get the cue from the requirement given in the step 1, it will compare the values of the cue associated with each element of vector and matrices stored as a pre-existing memory, there

might be a possibility that a cue may belong to more than one matrix as in Figure 1. There are two cues Ca, Cf for memory M11, now check the strength of cue as more the strength, better will be the association of incoming requirement with the particular matrix of the system or brain of the stakeholder.

For the retrieval of the matrix Mij, the cue "notebook" is compared with the pre-existing matrix and we get three instances, where the "notebook" has been used.

(a) Might be related to the item vector i.e. his/her.

Might be related to the context i.e. say "class of the school" as a context which is giving the plural pronoun sense.

Might be related to individual student Rohan, which is in a singular pronoun reference.

Say it gets the best or maximum value for the cue that is related to Rohan that gives the context of singular pronoun reference, we get the matrix Mi j .

Step5:-Multiply the input requirement vector (1,-1,-1, 1, 1) with a matrix obtained in step 4.

i.e. {1 -1 -1 1 1} * Mi j

Step 6:- The Output of the step 5 is (1 1 1 1 -1), which when converted into the natural language form with the help of grammar gives the output say as "Everybody should bring his/her notebook".

Step7:-System ask the client to analyze the output in natural language, as the above output gives the suggestion sense and does not create any ambiguity.

The other output may be interpreted as-

A. Everybody must bring his/her notebook.

B. Everybody has to come with his notebook.

C. Everybody must have with him his/her notebook.

All the three interpretation are unambiguous, the other interpretation may be like-

W. Everybody must bring their notebook.

The above output is ambiguous because Singular Pronoun/Subject is used with Possessive Pronoun in Plural Sense, Plural Pronoun generally tells about the group but does not tells about the exact number "some" or "all", that creates the ambiguity.

X. Everybody must bring the notebook.

The above output is ambiguous because here Group can be assumed but in the group we don't know whether there are males or females present in the group.

Y. Everybody must bring his/her notebook

The above output is ambiguous because this sentence is in command sense and if previous context is of command type then the interpretation will be different.

4 CONCLUSION

Our main work is focused on the interview technique of the requirement elicitation phase, as the method deals with the problem of ambiguity and relates the context of a language with question-answering session and then devised a method that relates the previous context of stakeholders brain with the current answer given by the client, that stakeholder has to memorize and again has to reply to the user, so that both of them can easily spot the ambiguity, it can be concluded that ambiguity depends on the frame of mind of stakeholder who is participating in the interview. As there are many types of ambiguities and one solution can cover only one class of ambiguity so the other techniques should be tried on.

5 FUTURE WORK

Many things are assumed in this paper like, domain of Natural Language is vast so as the problem of ambiguity, for that we have to take the large grammar with many productions. For auto associative net we have used the appropriate representation like bipolar that consist of two values but two values are not sufficient for any variable as plural pronouns are very large in number. We have taken the simplest model of memory as complexity related to encoding the production rule into appropriate representation is not considered, complexity related to storage, association, selection of matrix from the memory or retrieval of information are related issues and will be considered in the near future.

6 ACKNOWLEDGEMENTS

First of all, we would like to acknowledge all those researchers, those who are unknowingly not credited in this paper, and then we would like to acknowledge our Hon'ble V.C. (Prof. S.W. Akhtar) and Hon'ble P.V.C. (Proff. S.M. Iqbal) for giving us the research environment in the University, we will also like to thank Mr. Afzal-Ur-Rehman for his valuable inputs.

7 REFERENCES

[1] Hickey and A. Davis, "Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes," Thirty-Sixth Hawaii International Conference on System Sciences (HICSS 2003), January 2003

[2] M. Hickey and A. M. Davis, "A Unified Model of Requirements Elicitation," Journal of Management Information Systems, vol. 20, no. 4, pp. 65-84, 2004.

[3] M. Hickey and A. M. Davis, "Elicitation Technique Selection: How do Experts do it," Proceedings: 11th IEEE International Requirements Engineering Conference, Monterey Bay, California, IEEE Computer Society, pp. 169-178., Sept. 2003.

[4] Allison Scogin, The Factors of Software Systems that Contribute to Requirements Elicitation. Software Engineering Research and Practice: 296-302, 2006

[5] Bashar Nuseibeh, Steve Easterbrook, Requirements Engineering: A Roadmap, International Conference on Software Engineering Proceedings of the Conference on the Future of Software Engineering, Limerick, Ireland, IEEE Computer Society, pp 35 - 46, 2000.

[6] Berry D.M., "The Importance of Ignorance in Requirements Engineering", Journal of Systems and Software 28(2), pp. 179–184, February 1995.

[7] Christel, M.G. & Kang, K.C., Issues in Requirements Elicitation. Software Engineering Institute (SEI) Technical Report CMU/SEI-92-TR-12, 1992.

[8] Daniel M. Berry, Erik Kamsties, Michael M. Krieger, From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity, 2003.

[9] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley and Sons, pp 3-10, 1998.

[10] Hebb, D.O, the Organization of Behavior, John Wiley and Sons, 1949.

[11] Humphreys, M.S., Bain, J.D., & Pike, R. Different ways to cue a coherent memory system: A theory for episodic, semantic and procedural tasks. Psychological Review, 96, 208-233, 1989.

[12] Hunt, A. & Thomas, D., The Pragmatic Programmer: From Journeyman to Master. Reading, MA: Addison Wesley Longman, Inc., 2000.

[13] Pike, R. A comparison of convolution and matrix distributed memory systems. Psychological Review, 91, 281-294, 1984.

[14] Randolph Quirk, Sidney Greenbaum, A University Grammar of English, 4th Edition. A Macmillan Publishing Company: New York. 1994.

[15] Standish Group, "The Chaos Report", www.standishgroup.com, 1995.

[16] William Scott, Stephen Cook, A Context-free Requirements Grammar to Facilitate Automatic Assessment, AWRE'04 9th Australian Workshop on Requirements Engineering, 2004.