IMPLEMENTATION SENTENCE ANALYSIS VIA VERB [+]

تحليل الجمله بواسطة الفعل

Maitham A. Naji [*]

Abstract:

The objective of this thesis provides computerized technique to design relational database. The program divides the entered sentence into three parts, Subject Group, Verb Group and complement group. Subject and complement group are representing the source and destination entities or attributes with its relationship. Verb group is representing the relation name that connects the source and destination entities. The program matches each word in the sentence with verb dictionary. When the verb has been found, the program stores it in a list and replace the verb group with any symbol therefore it become very easy to get subject group which any thing before symbol while complement group every thing after symbol.

المستخلص:

الغرض من هذا البحث توفير تقنية حاسوبية لتصميم قواعد بيانات علائقيه. البرنامج يقسم الجملة المدخلة إلـــى ثلاثة أجزاء. مجموعة الفاعل، مجموعة الفعل ومجموعة المتمم. مجموعة الفاعل والمتمم تمثل كينونات أو خصائص المصدر والمقصودة مع علاقاتها. مجموعة الفعل يمثل اسم العلاقة التي تربط كينونات الفاعــل والمــتمم. البــرامج

[+] **Received on 24/3/2010 , Accepted on 3/2/2011 .**

[*] **Assist Lecher /Collage of Electronics and Electrics Technical**

يطابق كل كلمة في الجملة مع قاموس الفعل، عندما يوجد الفعل البرنامج فانه يخزن الفعل في قائمة ويستبد الفعل في الجملة إلى رمز معين لذلك يصبح من السهل الحصول على مجموعة الفعل وهي أي شيء قبل الرمز بينما مجموعـــة المتمم هي أي شيء بعد الرمز.

Introduction:

One of the most important steps in developing a database application is the design of an effective database structure. The complexity and size of the problem usually makes it a difficult and time consuming task. Inadequate designs have present many problems. The failure for example to specify clearly the organizational goals and requirements has resulted in databases of limited scope and usefulness which are unable to adapt to change. In many cases these problem databases have prevented database management system from being effective data processing tools [1,2].

Traditional manual techniques for database design rely heavily on the designer's skill and experience, an approach that is neither efficient nor effective. In addition the large size of the design problems makes the design decisions time consuming and error prone. There is a need to develop methods of representing information that capture a complete and accurate picture of the database and its application environment. Owing to the size and complexity of problems faced, computer support is required. Computer Aided techniques can be successfully applied to the database design problem. When used with appropriate interface they can help the database administrator produce more quickly a better quality design with this assistance. The human database designer can make the necessary design decisions more intelligently and usually more quickly [3]. The author in [4] deals with German natural language dialogue tool for supporting the data design process while in this paper use English

natural language. Most previous papers and researches use Semantic Network (SN) [4,5]. In [5], the author uses the Semantic Network (SN) to represent the Entity-Relationship Model (E-R Model) using Prolog language while this paper uses Visual Basic language for Natural Language Processing (NLP). The artificial intelligent language like (Prolog, Lisp) does not support with tools for relational database processing like creating tables, relationships, structure query language (SQL) and designing Forms, Reports [6,7].

Natural Language Syntax:

Natural Language Processing (NLP) is a subfield artificial intelligence linguistic, devoted to make computers "understand" statements written in human languages [8]. The author in [2] allow the database administrator to describe there application in a subset of English language.

The syntax in this language is classified to three groups:

<Subject Group> <Verb Group> <Complement Group>

1- Subject Group: It's every thing before verb groups.

2- Verb Group: The verb groups are classified in two types Known Verbs and

Unknown Verbs

- Known Verbs: Include the following verbs:
  [TO BE, TO HAVE, TO BELONG, TO CHARACTERIZE, MUST, MAY]

- All other verbs are considered as Unknown Verbs.

3- Complement Group: It's every thing after verb groups.

Examples of possible sentences are given below [2].

students and employees are persons

a student is enrolled in one or more courses

a class is given by an instructor

a person has a name , an age and one or several addresses

one or more classes are associated with a course

a course is given by an instructor

a course has a number , a name , a day and an hour

a class has a number , a day and an hour

a student has a student number

an employee has a social security number and a phone no.

a staff member has a work address

an employee works in a department

The system accepts the above statements; the subject group presents the source entities and the verb group presents the relationship name while the complement group presents the destination entities or attributes.

3- System Implementation

The program has three main stages to split sentence to three parts. The flowchart of the main program is shown in Figure (1).

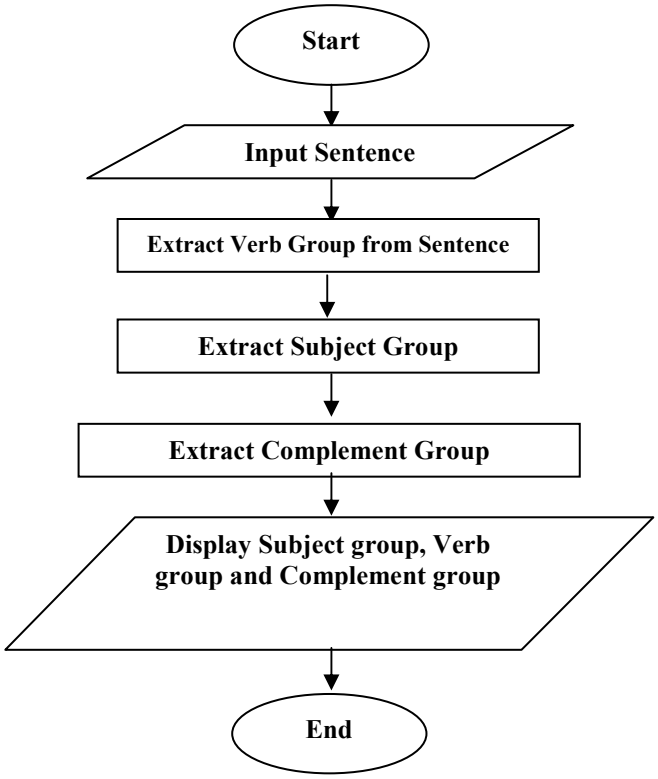

Figure (1) Main Flowchart of Sentence Analysis

The sentence was converted to list in order to deal with each word in sentence. The program first try to find verb that consist of three words. These words were compared with verb dictionary. If the verb was found, the program will change the default flag value from false to true and exit from the verb searching program as shown in Figure(2)

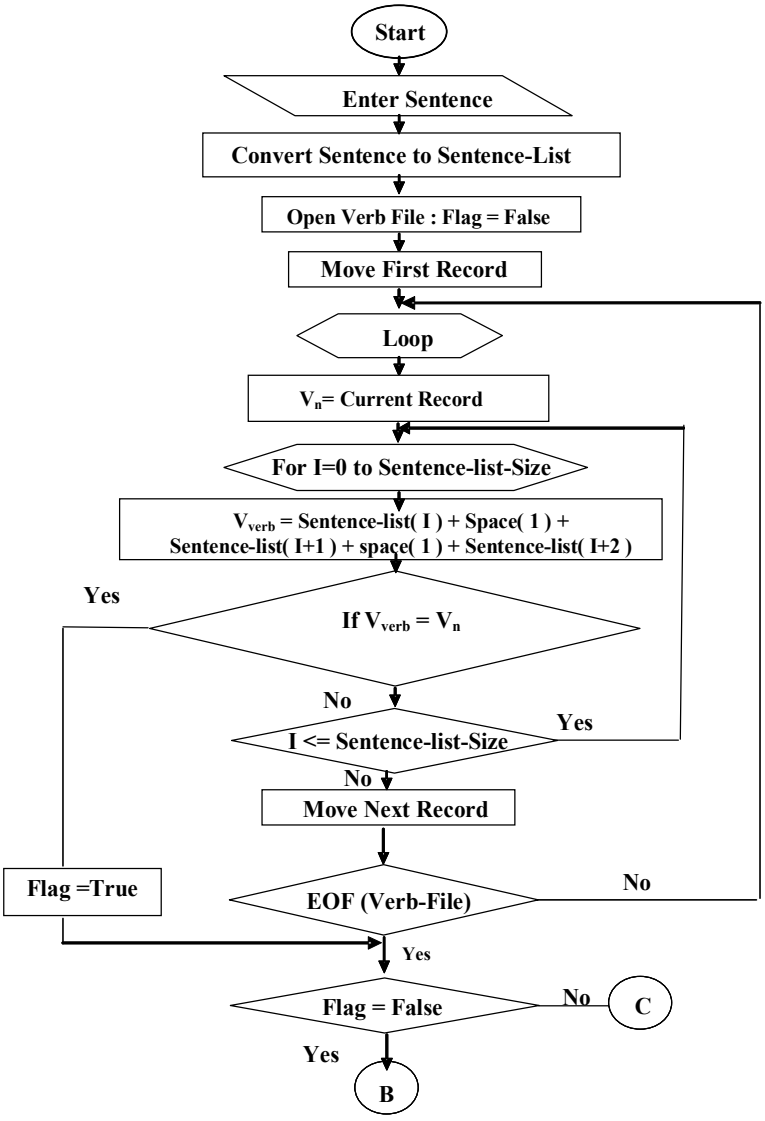

Figure (2) Verb Extraction

If the flag remains false, the program execute another codes to find verb that consists of two or one word. In case of failure that means the verb does not exist in the dictionary and the program asks the user to save the verb in dictionary and the user should start again to analysis sentence as shown in Figure (3).

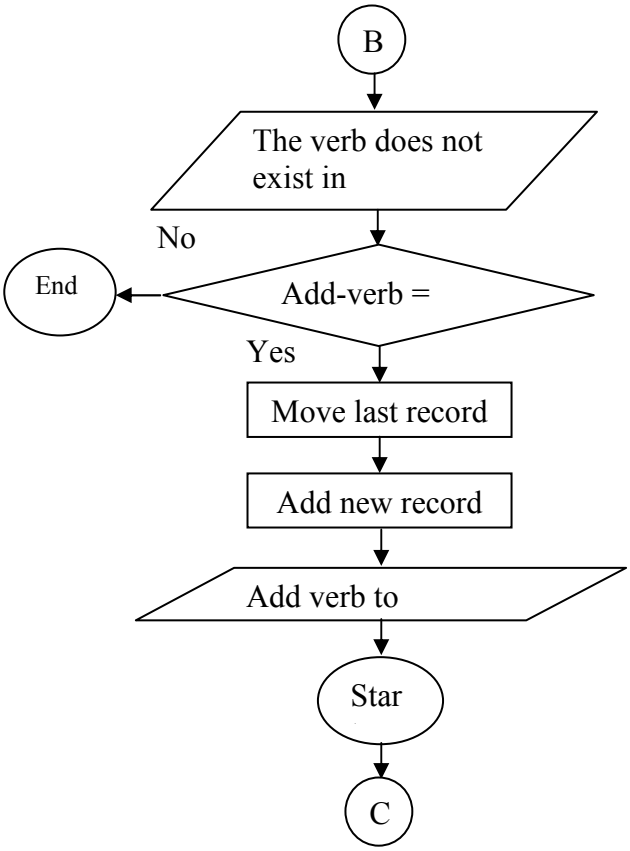

Figure (3) Adding Verb to Dictionary

The verb group was converted to a verb list. Each array's element was compared with sentence list, the program sets an asterisk (*) to the sentence list if two words was similar. In this way, the program can split between subject and complement groups as shown in Figure (4).

The following example illustrates the behavior of the system:

Input Sentence: " a student has a name , an age and one or several addresses"

Output Sentence list: " a student * a name , an age and one or several addresses"

The output of figure (4) will be an input to figure (5) in order to extract the subject group. This is done by taking all elements of the sentence list before the asterisk (*) and storing it in

a new subject list, the words in the sentence list are replaced with a semicolon (;) as shown in Figure (5). For example:

Input Sentence list:" a student * a name , an age and one or several addresses"

Output sentence list: "; ; * a name , an age and one or several addresses"

Output subject list: "a student"

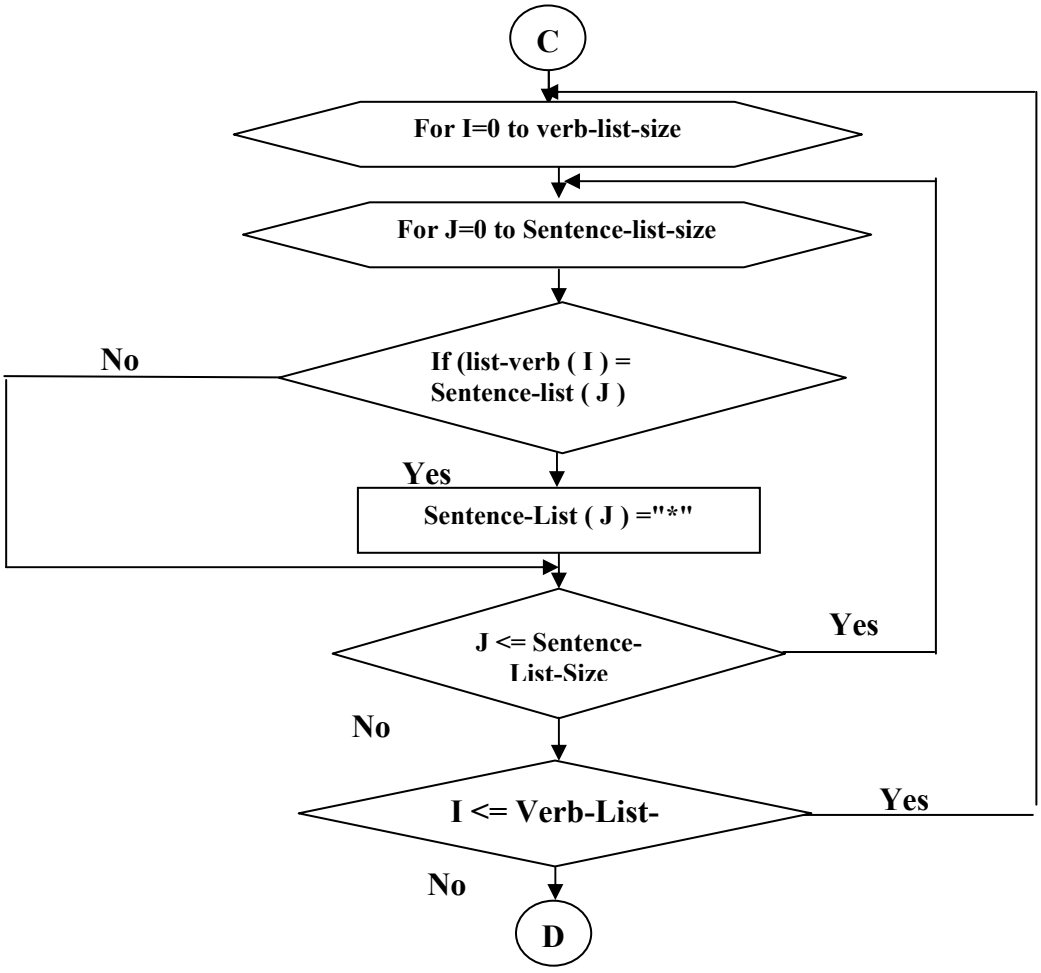

Figure (4) Verb Group Replacement

The subject list may had many subjects, the program compares each element with symbols ("and", ";" and ","), if the element was not equal with these symbols, that means we get the subject and add it to another subject list; otherwise the element is added to the string variable to get the subject of two or more elements. This process continues until to find all subjects' groups. As shown in figure (5).

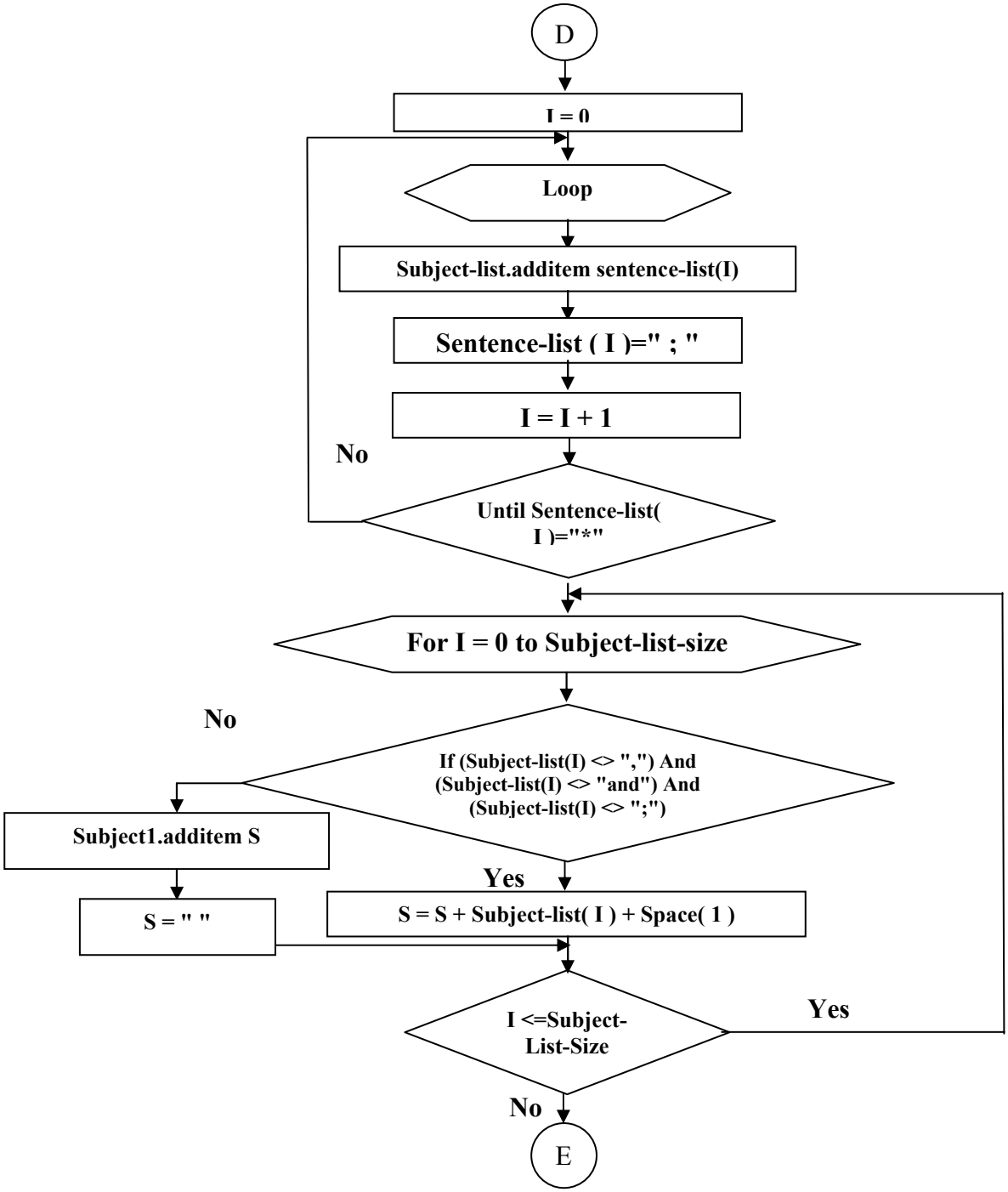

Figure (5) Subject Group Extraction

The program creates a complement list by comparing each element in Sentence list with the above symbol, the element is added to complement list if its not equal the symbols. The process is continued until the end of sentence as shown in Figure (6). For example:
Input Sentence list: ";;* a name , an age and one or several addresses"
Output Complement list " a name , an age and one or several addresses"
The complement list may had many complements, the program compares each element with symbols ("and", ";" and ","), if the element was not equal with these symbols, that means we

get the complement and add it to another complement list; otherwise the element is added to the string variable to get the complement of two or more elements.

This process continues until to find all complements' groups as shown in figure (6).For example
Input Complement list: "a name , an age and one or several addresses"

Output Complement1 list (0): a name
Output Complement1 list (1): an age
Output Complement1 list (2): one or several addresses

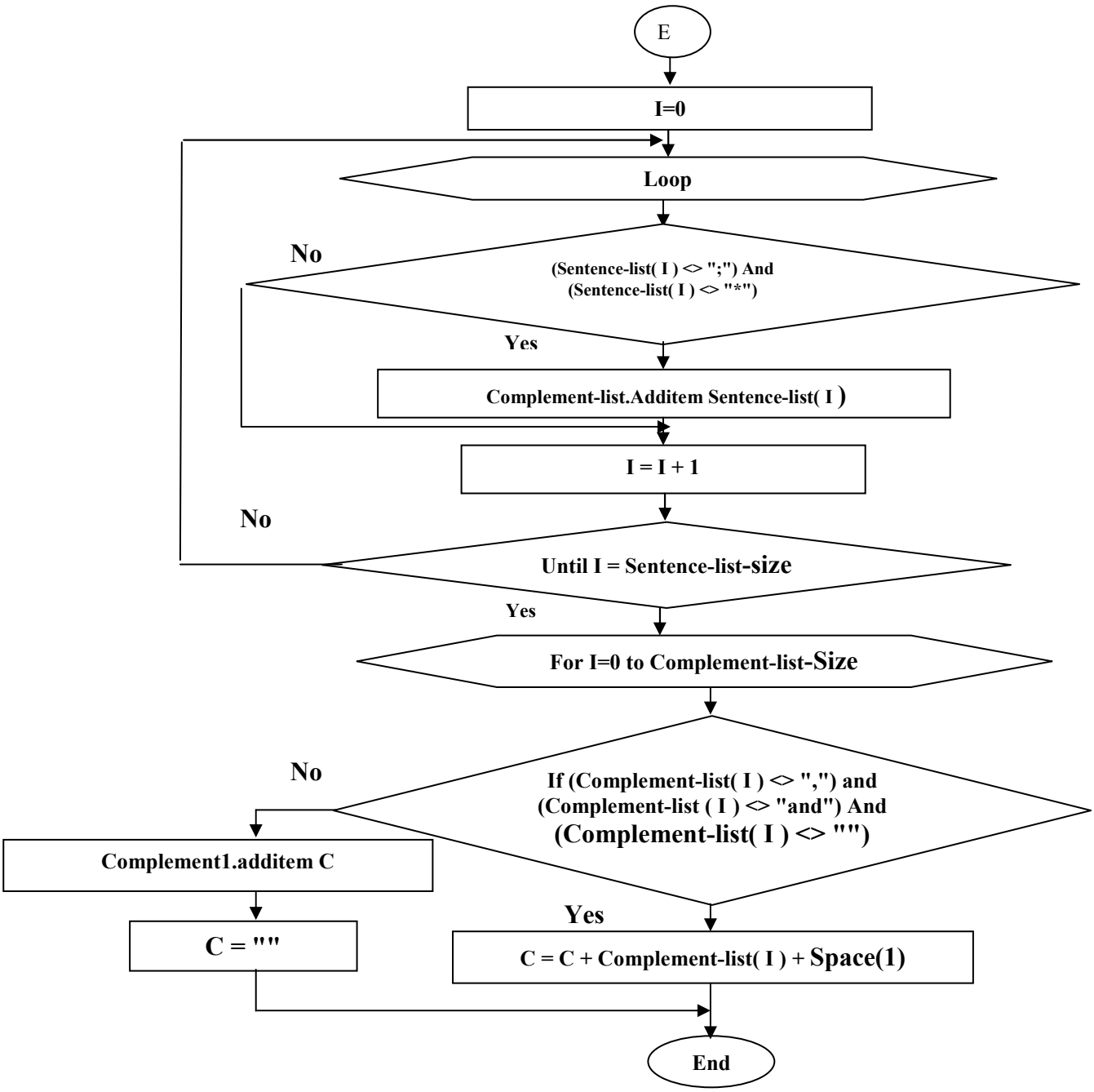

Figure (6) Complement Group Extraction

The Implementation:

The user should write the sentence according to the syntax that have seen in page 2, the sentence is written in the blank text box then click "Sentence Analysis" button, the program will match each word in sentence with verb dictionary to find the verb; if the verb does not found the system asks the user to save the verb in dictionary then clicking "Sentence Analysis" button to repeat the process of parsing. After the verb group has been found the Sentence list is divided into Subject list and Complement list. The complement list is taken as an input to the next process that separate each item in single row in a new Complement list, the same process will be done to the Subject list as shown in Figure (7). The "New Sentence" button is used to clear the text box from previous sentence and all other lists and variables will be cleared from data of previous execution. The last button is "Exit" to end program execution.

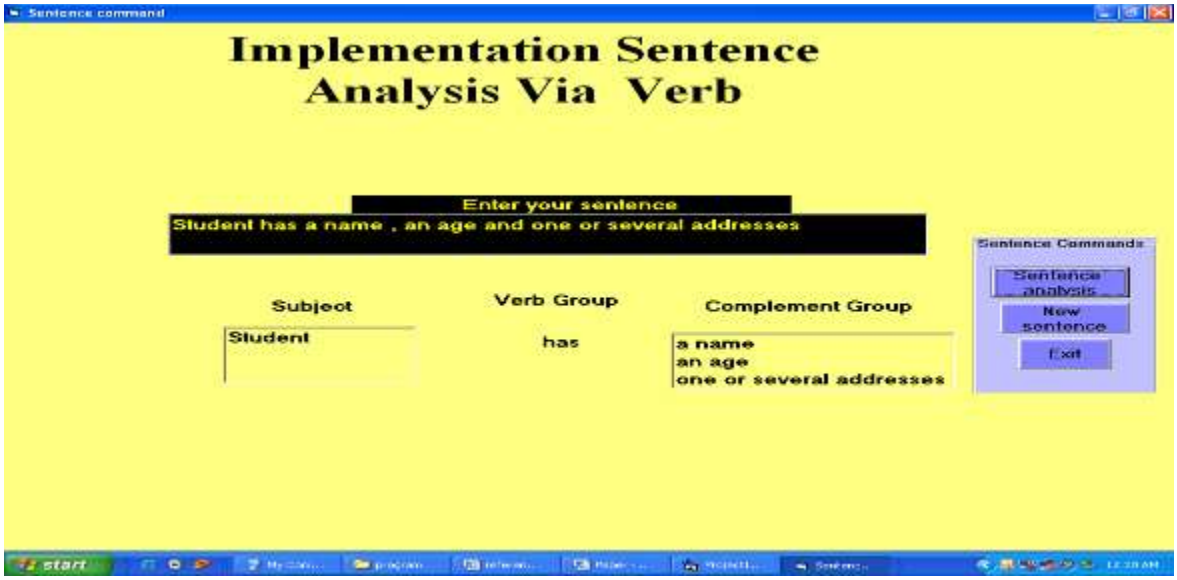

Figure (7) Sentence Analysis Window

Conclusion:

The sentence analysis is the first step of analysis, designing and built accurate database system. The system can be developed by split the relationship from the subject and complement groups in order to create Entity-Relationship Table (E-R-T). The E-R-T is very useful to apply Inference Rule (IR) for Function Dependencies (FD). The FD process makes the operation (Adding, Deleting and Editing) of the information without any mistake or redundant.

References:

1- S. Bing Yao; "Principles of Database Design volume-1(Logic Organizations)"; Printice – Hall; 1985.
2- C.J. Date; "An Introduction to Database system"; Addison – Wesley; 1999.
3- "Logic and Semantic Networks"; On line document available at www.doc.ic.ac.uk.
4-"Using Natural Language for Database Design" ; On line document available at citeseerx.ist.psu.edu

5- Salwa A. Taha; "Tools to Aid for Database Design and Development"; Ph. D. Thesis, Cairo University; 1994.
6- Virginia Andersen; "How to Do Everything with Access 2002"; McGraw-Hill Edition; 2001.
7- John Carter; "Database Design and Programming with Access, SQL and Visual Basic"; McGraw-Hill; 2000
8- "Natural Language Processing Artificial Intelligence" on line document available at www.myreaders.info