AUTOMATIC DATABASE CONSTRUCTION FROM NATURAL LANGUAGE REQUIREMENTS SPECIFICATION TEXT

Geetha S.[1] and Anandha Mala G. S.[2]
[1]JNTU Hyderabad, Telangana, India
[2]St. Joseph's College of Engineering, Chennai, India
E-Mail: geethasoman@yahoo.com

ABSTRACT

Currently there is a growing interest in the automation of extracting the information from natural language text which occurs as a large part of domain knowledge. Software Requirements Specification (SRS) enlists all the user's requirements that can be analyzed through elicitation process from natural language text which has its own limitations. In the present study an attempt is been made to construct a schema for the tables and their inter relationships to all the other tables extracted from the natural language requirements specification text. Initially, the schema for the table is constructed by identification of the Primary Key (PK) attribute based on adjectives, prioritizing the preference of the attributes and hand crafted rules was trained from the statistical data. The Foreign Key (FK) attribute identification is done to construct a relational schema, which establishes the inter relationship between the table attributes from the extracted primary key. Finally, a database which can show the relationship among the tables is built after the identification of the foreign key attributes of a table using highly referenced primary key attribute. By constructing a validated real time automated database, the user can query and acquire domain knowledge.

Keywords: natural language processing, relational schema, key attributes, hand crafted rules.

INTRODUCTION

This paper presents a methodology to transform the Natural Language Text (NLT) into structured database representation. Translation of natural language SRS into usable software poses quite a challenge.

In many applications the information available is in the form of NLT rather than the structured one. Discovering knowledge from NLT is becoming an important aspect in Knowledge Discovery Database (KDD). This comprises the process of structuring the input, extracting patterns and interpretation of output.

The process of software development focuses on capturing the natural language requirements specification such as understanding the user needs and goals. Initially the software development starts with requirements capturing which is expected to provide the total understanding of the system and in turn describes the needs that are to be accomplished. In the requirements capture, when the requirements are not structured, it needs translation into structured one. Many algorithms have been reported so far to semi automate or automate the natural language SRS to a structured one. The job of software development is made easy as it is capable of representing the database and acquiring the knowledge by applying the queries. Having this perception, a methodology for converting natural language SRS text into the relational database finds its way in identifying the schema for the tables and relationship corresponding to all the tables extracted from the requirements specification proposed here.

A number of issues are there to be addressed in designing the relational schema of the database by identifying the referential integrity constraints enforced on a database. Automatically discovering the semantic associations between the table schema elements, namely PK's and FK's are challenging. This paper describes the technique of transforming the natural language requirements specification into the relational schema, which helps to bridge the gap between the informal natural language requirements specification and the formal language specification so as to offer effective software solutions. Based on this observation, the mapping rules and rule sets approach are used to identify PK and FK to construct the relational schema. The automatic construction of the relational schema from natural language SRS provides a platform to extract the information for specific domain.

The rest of the paper is organized as follows: Related work is discussed. Proposed methodology is explained. The procedure for results and evaluation of the different SRS are discussed. Conclusion and future work is discussed.

RELATED WORK

The construction of a schema for the database is a crucial step in data analysis. Many researchers have attempted several methods to achieve the above task (Rohit j. Kate and Raymond Mooney, 2010) extracted and presented the entities and their relationship in a sentence instead of text by which a schema cannot be constructed. (Yannis Sismanis *et al*., 2006) proposed an algorithm Gordian to discover composite key attribute from the large datasets.

A schema proposed by (Bauckmann, *et al*., 2007) identifies Inclusion dependencies as specified pre condition for foreign keys. For the generation of relational database schema, he used open mms schema and parser to import the protein databank. Only Inclusion dependency between single attribute and sequences of attributes in a given database was considered in constructing the schema by (Marchi *et al*., 2009). Identification of relationship between the columns which explains the properties and characteristics of the values from the relational database was studied by (Meihui Zhang *et al*., 2011). Though ample work has been done in constructing the schema for the database analysis, in many ways building the schema for the database from NLT is still in the preliminary level. Significant work done by (Luis Tari *et al*., 2012) in proposing a method of information extraction from NLT and by (Xian-Yi Cheng *et al*., 2011) works on constructing the database based on whole, physical relation, generic relation etc from NLT, they did not explain the relational database schema from NLT. The author's (Divesh Srivastava, 2010 and Marco D. Adelfio, 2013) also have done extraordinary work in extracting the schema from complex database and metadata in tabular form, but not mentioned about the relational schema construction from the NLT. (G.S. Anandha Mala *et al*., 2011) proposed a method to extract Object Oriented Elements (OOE) from requirements specification text and (N. Mfourga, 1997) works on extracting the entity relationship from the schema of the relational database did not give much importance in constructing the relational schema for the database. An algorithm TGen proposed by (Michael J. Cafarella *et al*., 2007) discovers the schema automatically for the extracted data, but not specified the about the removal of data duplication and relationship among the columns.

Different NLP techniques to extract useful information from NLT were presented by (Jekub Piskorski and Roman Yangarber., 2013). So the proposed system is built to construct the database by extracting the schema information that describes the logical structure of the tables in an automated manner there by considering the natural language requirements specification.

PROPOSED METHODOLOGY

Identification of the PK / FK attribute is an elemental concern for different data management tasks such as extracting the relationship and queries. To build the database, discovering the keys and establishing the relationship between the tables using the keys to identify the relational schema is a crucial step in understanding and working with natural language SRS.

The proposed methodology deals with automatic construction of the relational schema by identifying the key attributes to build the database from SRS using rule based approach. The Figure-1 shows the system architecture of schema extractor.

Domain knowledge elicitor

The domain knowledge elicitor takes the natural language SRS text, which describes the user needs as the input. The problem statement SRS is split into sentences to reduce the processing overheads. The PoS tag designates each word of all sentences and classifies the words as nouns, verbs, adjectives, etc. Brill tagger is used for this purpose (Dan Garrette and Jason Baldridge, 2012; Brill, 1992). Tagging of the words is necessary to chunk the words that form a noun phrases or the verb phrases, then the noun and the verb phrases are classified based on simple phrasal grammar. When normalizing the sentence into Subject-Verb-Object (S-V-O) pattern, sometimes the subject and object happen to be a pronoun and they have to be resolved to their respective noun phrases (Mitkov, 1998). Then the sentence has to be interpreted into S-V-O pattern to map the words into OOE (Anandha Mala *et al.*, 2006).

Generation of class diagram

The OOE namely classes, attributes, methods and relationships are identified based on simple rule based approach from the S-V-O pattern (Anandha Mala *et al.*, 2006).

- Translating Nouns to classes.
- Translating Noun-Noun to class property according to the position.
- Translating the lexical verb of a non personal noun to a method of this noun.
- Translating S-V-O structure to a class diagram with the subject and object as classes both sharing the verb as a candidate method.
- A simple understand is used to conclude which nouns are classes and which from the attribute.

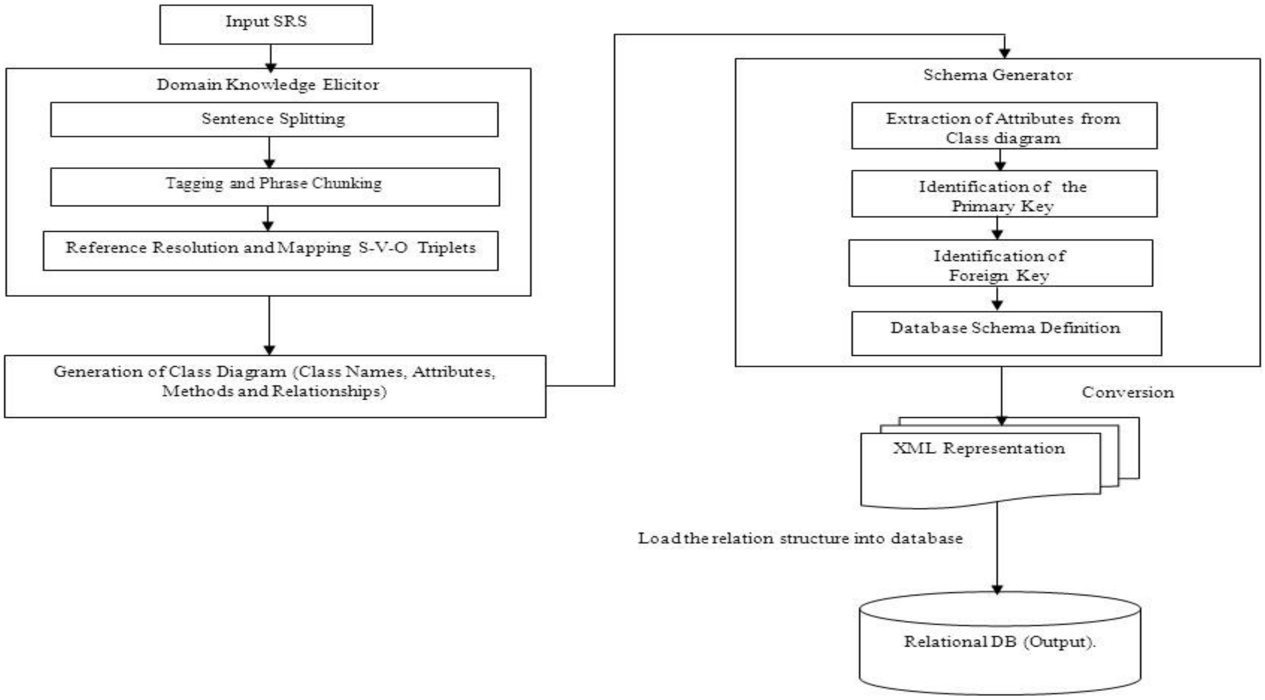

Input SRS
Domain Knowledge Elicitor
Sentence Splitting
Tagging and Phrase Chunking
Reference Resolution and Mapping S-V-O Triplets
Generation of Class Diagram (Class Names, Attributes, Methods and Relationships)
Schema Generator
Extraction of Attributes from Class diagram
Identification of the Primary Key
Identification of Foreign Key
Database Schema Definition
Conversion
XML Representation
Load the relation structure into database
Relational DB (Output).

Figure - 1. System architecture of schema generator.

Schema generator

This module extracts the schema by using the OOE identified by extracting the attributes from the class diagram, identifying PK and FK using rule based approach. Later the schema is represented as XML and stored in the database.

Extraction of attributes from class diagram

A relational schema represents the various properties of the tables and attributes (columns) within the tables. Attributes are the data members of the class and are related with each other in the class. Attribute shows the properties and characteristics of the values to be stored. In order to identify the attributes within the tables, attributes of the classes have to be considered. The list of attributes of all the classes from the result are extracted and to be stored in the table to construct the relational schema.

Identification of the primary key

Identification of the primary key attribute maintains the uniqueness of the table. Hence a rule based approach is proposed in this module to identify the primary key attribute from the attributes of all the tables. The attributes of the table plays a vital role in forming the metadata of the database. The PK attributes should have the following properties as

- The instance of an entity must have non null value
- Each instance must have an unique value
- No change in the values

This is achieved by applying the hand crafted rules for retrieving the PK attributes. The rules are as follows.

Rule-1: If the sentence is in the form of “Subject + Possessive verb + Adjective + Object”, then the object is a key attribute.

Rule-2: When the sentence is in the form of “Subject + Possessive verb + Object”, then if the object is prefixed or suffixed with set of predefined (training data set) words, then the object is a key attribute.

Rule-3: When the sentence is in the form of “Subject + Possessive verb + Object”, if the object is in the trained data set, then the object is considered as a key attribute.

Rule-4: By prioritizing the priority assigned to the objects of the form “Subject + Possessive verb + Object1”, “Subject + Possessive verb + Object2”, based on the relevance between the subject and object1, object2. Then the highest priority object nominated for the key attribute.

Identification of the foreign key

As PK attribute of all the tables are identified using the rule sets, the next step is to identify the FK attributes. FK attribute can be the single attribute or combination of the attributes which forms relationships across the database tables. To identify the FK attribute of the database tables, the attribute must satisfy the following:

- The FK/PK attribute name must be similar
- The FK should cover almost all the PK

- The FK should have cardinality 1:1, 1: N, M: N.

The system checks the PK attribute of all the tables. Then the occurrence of the primary key attribute of all the tables matches with the attributes of other database tables and assign that as a foreign key attribute.

Database schema definition

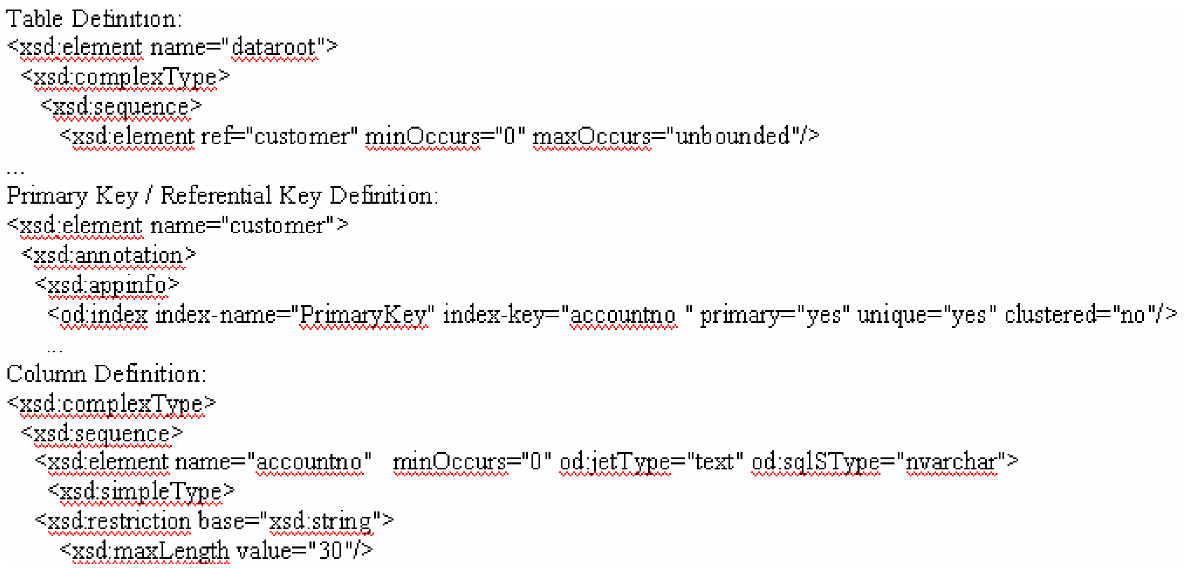
Defining the schema of the database is important for data analysis. The database schema is designed to store information about the tables extracted from SRS. The table includes a set of attributes and each attribute is associated with data type. The type of attribute is determined from which the attribute derives its properties. The data types considered to assign for the attributes of the table are number, character, date, etc. After distinguishing the key attributes (PK/FK) from all the tables, data type has to be assigned without violating the referential constraints. The following procedure is interpreted in order to assign the data type for each attributes.

- Assign date as data type for the attribute whose name has date as substring
- If not the date as data type, the attributes are considered either as the character or number as the data type. In order to resolve these issues, set of predefined data sets words used in the rule 2 for identifying the primary key attributes are utilized.

Once type is assigned to the attribute, the database schema shows all the tables, the attributes in all the tables and the relationship between the attributes of one table with other.

XML (eXtensible Markup Language) representation and relational DB

The tool has been developed to convert the relational schema extracted from the SRS into XML, an intermediate representation to be imported into the database. The data which is getting stored in the XML has to follow the structure as XSD (XML Schema Definition). The XML database can be constructed by considering the attributes of all the tables and its constraints. The code snippets for the different representation of the elements in a DB are shown in Figure-2. Then a simple java library called Jackcess, used to write the XML schema definition into MS Access database tables includes key attributes, non key attributes, data type and its size.



```
Table Definition:
<xsd:element name="dataroot">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="customer" minOccurs="0" maxOccurs="unbounded"/>
...
Primary Key / Referential Key Definition:
<xsd:element name="customer">
  <xsd:annotation>
    <xsd:appinfo>
     <od:index index-name="PrimaryKey" index-key="accountno " primary="yes" unique="yes" clustered="no"/>
     ...
Column Definition:
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="accountno"   minOccurs="0" od:jetType="text" od:sqlSType="nvarchar">
     <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="30"/>
```

Figure-2. Representation of information related to the table.

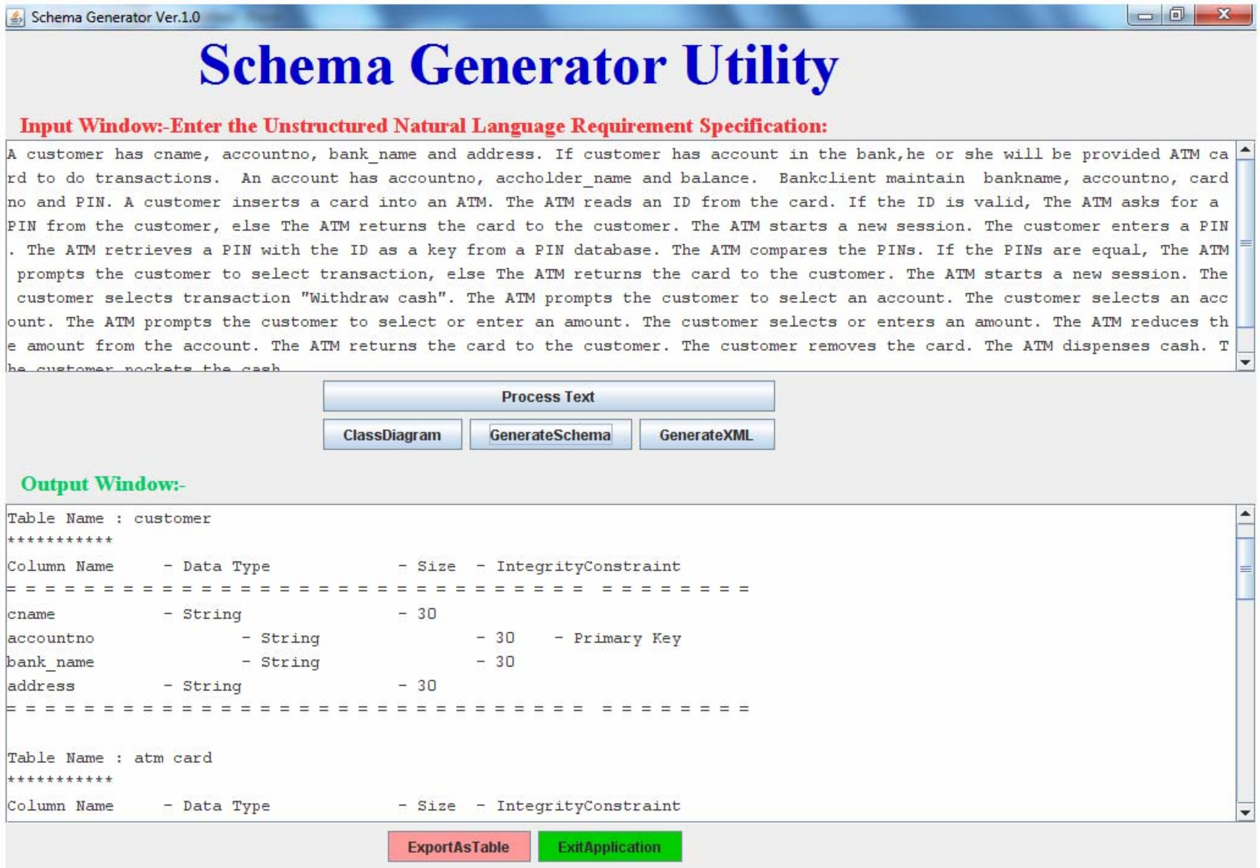
RESULTS AND DISCUSSIONS

The entire work (tool) was implemented using java and it was validated using around 200 samples of 100 to 150 lines approximately. The proposed approach creates a tool to generate relational schema automatically from the input SRS. The core idea is to identify the dependency of the attributes between the tables based on rule based approach. The process of finding the relationship between the attributes starts with the requirements extractor which interprets the SRS into S-V-O to map the word into OOE. The identified S-V-O patterns are used to classify OOE as class, attributes, methods and relationships using hand crafted rules. The crafted rules, priority and training data set are used to identify the PK attributes. With the use of PK attributes, FK attributes are identified based on the similarity of the attributes between the tables and the data type is assigned to all the attributes. Table-1 shows some sample schema retrieved from the SRS.

Table-1. Sample PK and FK attribute identification - ✓ says yes and X says no.

Customer				Account				Bankclient			
Attribute list	**PK**	**FK**	**Data type**	**Attribute list**	**PK**	**FK**	**Data type**	**Attribute list**	**PK**	**FK**	**Data type**
cname	X	X	string	accountno	X	✓	string	bankname	X	X	string
accontno	✓	X	string	accholder_name	X	X	string	accontno	X	✓	String
bank_name	X	X	string	balance	X	X	Number	cardno	✓	X	string
Address	X	X	string					Pin	X	X	string

The construction of the database schema is shown in Figure-3. The database schema to be converted into XML is imported into relational database without violating the integrity constraints shown in Figure-4.



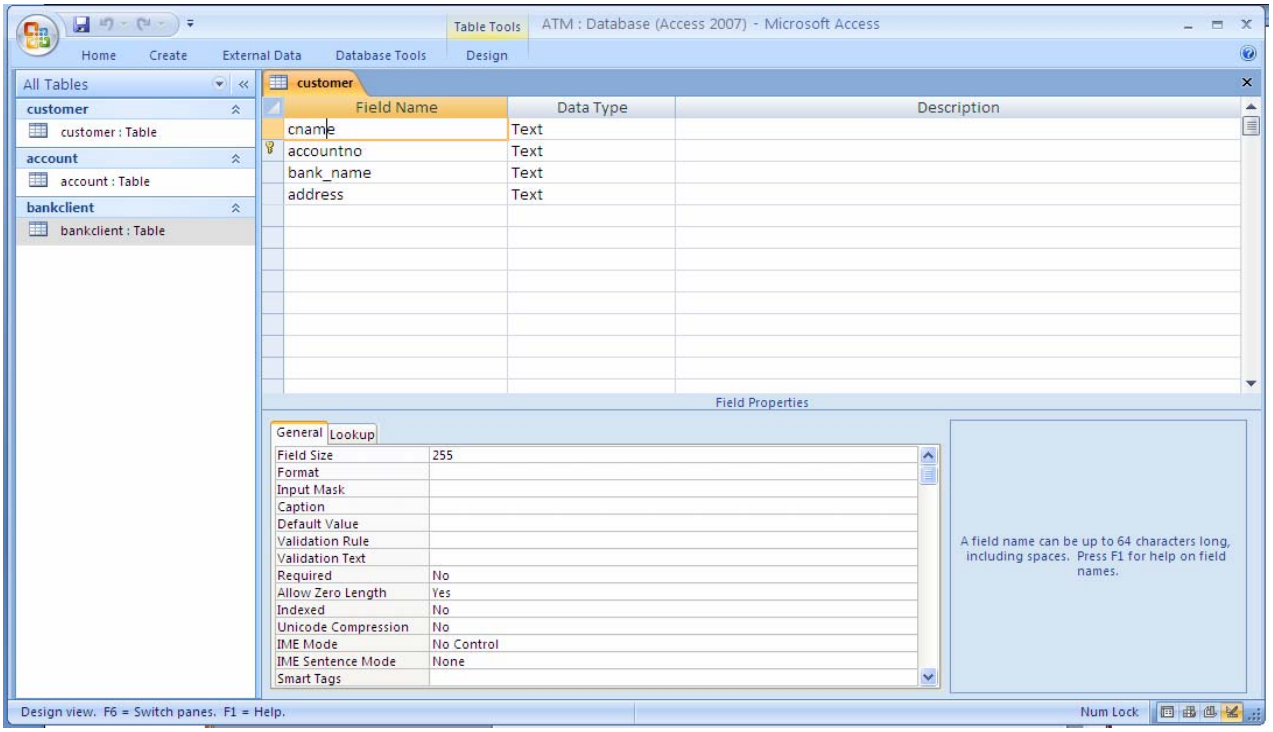
```
A customer has cname, accountno, bank_name and address. If customer has account in the bank,he or she will be provided ATM ca
rd to do transactions.  An account has accountno, accholder_name and balance.  Bankclient maintain  bankname, accountno, card
no and PIN. A customer inserts a card into an ATM. The ATM reads an ID from the card. If the ID is valid, The ATM asks for a
PIN from the customer, else The ATM returns the card to the customer. The ATM starts a new session. The customer enters a PIN
. The ATM retrieves a PIN with the ID as a key from a PIN database. The ATM compares the PINs. If the PINs are equal, The ATM
 prompts the customer to select transaction, else The ATM returns the card to the customer. The ATM starts a new session. The
 customer selects transaction "Withdraw cash". The ATM prompts the customer to select an account. The customer selects an acc
ount. The ATM prompts the customer to select or enter an amount. The customer selects or enters an amount. The ATM reduces th
e amount from the account. The ATM returns the card to the customer. The customer removes the card. The ATM dispenses cash. T
he customer pockets the cash
```

```
Table Name : customer
***********
Column Name     - Data Type            - Size  - IntegrityConstraint
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =  = = = = = = = =
cname           - String               - 30
accountno               - String               - 30    - Primary Key
bank_name               - String               - 30
address         - String               - 30
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =  = = = = = = = =


Table Name : atm card
***********
Column Name     - Data Type            - Size  - IntegrityConstraint
```

Figure-3. Automatic generation of database schema.

The tool does not miss to identify the PK and FK attributes. But approximately 12.92 % and 10.61 % additional primary key and foreign key attribute are identified by the schema generator based on the true positives and false positives which computes the accuracy of the tool. The results were compared with the PK & FK attributes identified by the human intervention. The tool is tested for the following domains like Airline ticket reservation, banking system, retailing system and so on.



Field Name	Data Type	Description
cname	Text	
accountno	Text	
bank_name	Text	
address	Text	

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Figure-4. Automatic generation of table schema in the relational database MS - access.

Accuracy is measured as

Accuracy = (Number of True Positives + Number of True Negatives) / (Number of True Positives + Number of False Positives + Number of False Negatives + Number of True Negatives)

- Number of true positives = Correct identification of the PK and FK attributes.
- Number of true negatives = Correct rejection of the PK and FK attributes.
- Number of false positives = Incorrect (Extra) identification of the PK and FK attributes.
- Number of false negatives = Incorrect rejection of the PK and FK attributes.

CONCLUSION

This approach presents an idea to restructure the natural language text into structured information. Natural language processing techniques and set of rules are used to extract the domain knowledge from the natural language SRS. The system compactly constructs the relational schema by identifying the class diagram, schema for the tables and relationship between the tables using primary key / foreign key attributes. Thus the set of hand crafted rules are presented to identify the key attributes to construct the relational schema. Then the relational database automatically constructs from relational schema which is converted into XML.

The user can query and acquire the domain knowledge from the relational database built from natural language SRS. The developed tool is found to extract the schema efficiently which is indicated by the performance measures like True Positives and False Positives. This method could be extended to identify the functional dependencies exist in the table.

ACKNOWLEDGEMENT

The authors wish to thank the management of Dr. MGR Educational and Research Institute-University, Chennai for providing all the computational facilities to carry out this work.

REFERENCES

Anandha Mala G.S, Jayaradika J and Uma G.V. 2006. Object Oriented Visualization of Natural Language Requirement Specification and NFR Preference Elicitation. IJCSNS International Journal of Computer Science and Network Security. 6(8).

Bauckmann J, Leser U, Naumann F and Tietz V. 2007. Efficiently detecting inclusion dependencies. In ICDE. pp. 1448-1450.

Divesh Srivastava J. 2010. Schema Extraction. Proceedings of the CIKM ’10 of the 19th ACM international conference on information and knowledge management.

Brill J E. 1992. A simple rule-based part -of-speech tagger. Proceedings of 3rd conference on Applied Natural Language Processing.

Dan Garrette and Jason Baldridge. 2012. Type-Supervised Hidden Markov Models for Part-of-Speech Tagging with

Incomplete Tag Dictionaries. Proceedings of the 2012 joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 821-83.

Jekub Piskorski and Roman Yangarber. 2013. Information Extraction: Past, Present and Future. Theory and Applications of Natural Language Processing. Springer - Verlag Berlin Heidelberg.

Luis Tari, Phan Huy Tu, Jorg Hakenberg, Yi Chen, Tran Cao Son, Graciela Gonzalez and Chitta Baral. 2012. Incremental Information Extraction Using Relational Databases. IEEE Transactions on Data and Knowledge Engineering. 24(1): 86-99.

Marchi F.D, Lopes S and J.-M. Petit. 2009. Unary and n-ary inclusion dependency discovery in relational databases. Journal of Intelligent, Information Systems. 32(1): 53-73.

Marco D. Adelfio Hanan Samet. 2013. Schema Extraction for Tabular Data on the Web. Proceedings of the VLDB Endowment. 6(6).

Meihui Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, Cecilia M. Procopiuc and Divesh Srivastava. 2011. Automatic discovery of attributes in relational databases. Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. pp. 109-120.

Mfourga N. 1997. Extracting Entity-Relationship Schemas from Relational Databases: A Form- Driven Approach. IEEE, Proceedings of the 4th Working Conference on Reverse Engineering (WCRE '97).

Michael J. Cafarella,Dan Suciu and Oren Etzioni. 2007. Navigating Extracted Data with Schema Discovery. Proceedings of the 10th International Workshop on Web and Databases.

Mitkov, R.1998. Robust pronoun resolution with limited knowledge. Proceedings of 18th International Conference on Computational Linguistics (COLING'98) / Conference Montreal. pp. 869-875.

Rohit j. Kate and Raymond Mooney. 2010 Joint Entity Relation Extraction. Proceedings of the14th conference on Computational Natural Language Learning (CoNLL-2010). pp. 203-212.

Xian-Yi Cheng and Xiao-hong Chen, Jin Hua. 2011. The Overview of Entity Relation Extraction Methods. Intelligent Computing and Information Science, communications in Computer and Information Science, Springer. 134: 749-754.

Yannis Sismanis, Paul Brown, Peter J. Haas and Berthold Reinwald. Gordian. 2006. Efficient and scalable discovery of composite keys. ACM DL. Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB). pp. 691-702.