

Association for Information Systems AIS Electronic Library (AISeL)

PACIS 2004 Proceedings

Pacific Asia Conference on Information Systems
(PACIS)

December 2004

Extending the UML Concepts to Transform Natural Language Queries with Fuzzy Semantics into SQL*

Frank Tseng

National Kaohsiung First University of Science and Technology

Chun-Ling Chen

Academia Sinica

Follow this and additional works at: <http://aisel.aisnet.org/pacis2004>

Recommended Citation

Tseng, Frank and Chen, Chun-Ling, "Extending the UML Concepts to Transform Natural Language Queries with Fuzzy Semantics into SQL*" (2004). *PACIS 2004 Proceedings*. 72.

<http://aisel.aisnet.org/pacis2004/72>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Extending the UML Concepts to Transform Natural Language Queries with Fuzzy Semantics into SQL^{*}

Frank S.C. Tseng
Dept. of Information Management
National Kaohsiung First University
of Science and Technology
Kaohsiung, Taiwan, 811 ROC
imfrank@ccms.nkfust.edu.tw

Chun-Ling Chen
Institute of Information Science
Academia Sinica
Taipei, Taiwan, 115 ROC
chunling@gate.sinica.edu.tw

Abstract

Database applications tend toward getting more versatile and broader to comply with the expansion of various organizations. However, naïve users usually suffer from accessing data arbitrarily by using formal query languages. Therefore, we believe that accessing databases using natural language constructs will become a popular interface in the future. The concept of Object-Oriented modeling makes the real world to be well represented or expressed in some kinds of logical form. Since the class diagram in UML is used to model the static relationships of databases, in this paper, we intend to study how to extend the class diagram of UML to capture natural language queries with fuzzy semantics. By referring to the conceptual schema throughout the class diagram representation, we propose a methodology to map natural language constructs into the corresponding class diagram and employ Structured Object Model (SOM) methodology to transform the natural language queries into SQL statements for query executions. Moreover, our approach can handle queries containing vague terms specified in fuzzy modifiers, like 'good' or 'bad'. By our approach, users obtain not only query answers but also the corresponding degree of vagueness, which can be regarded as the same way we are thinking.

Keywords : Natural Language Query, UML, Class Diagram, Object-Oriented Modeling, Fuzzy Set Theory.

1. Introduction

In today's highly challenging environments, knowledge is becoming an important organizational asset that enables sustainable competitive advantage. To share information and knowledge, it is inevitable to pose queries on database systems, as they are ubiquitous and popular in storing enterprise data for various applications. Although the rapid evolution of Internet enables people to share information everywhere at anytime by retrieving data from database systems, however, naïve users still suffer from issuing formal query statements arbitrary.

Since most of the human knowledge is recorded in linguistic form, systems that could understand natural languages could help to access various kinds of information. Therefore, one of the most natural ways to issue queries on databases is using natural languages. In particular, as we have shown in (Tseng et al. 1992), many retrieval tasks posed by complex formal queries can be expressed by using only simple natural language statements.

In this paper, we present an approach to map the natural language queries with fuzzy semantics into SQL statements through the UML class diagram representations. This work

^{*} This research was supported by National Science Council, ROC, under Contract No. NSC-91-2416-H-327-005.

stems from (Metais 2002), which pointed out that natural languages and the database conceptual schema represents the conceptualization aspects of the real world. The inter-relationship between the linguistic level and the database level is shown in *Fig. 1*. In the linguistic level, natural languages are analyzed by using the pre-defined grammar and the dictionary stored in order to reduce ambiguity and complexity of natural language themselves. On the other side, in the database level, a schema can be regarded as the blueprint of the conceptual design of a database. Therefore, we recommend organizing the conceptual schema by class diagrams for constructing a natural language interface.

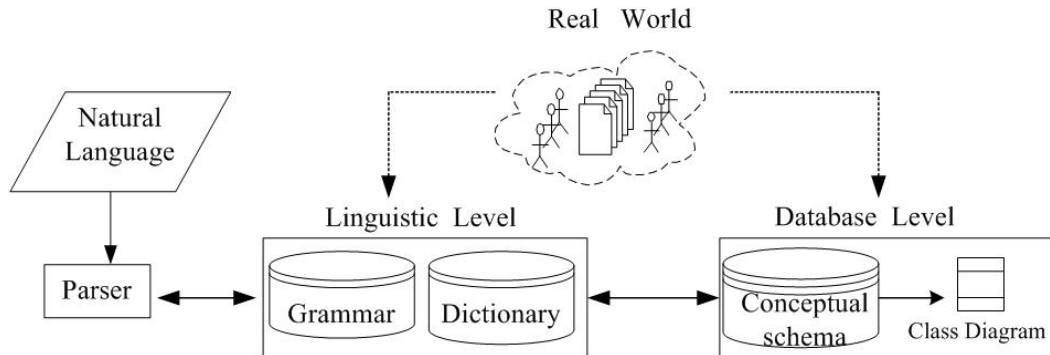


Figure 1: The inter-relationship between the linguistic level and the database level.

We explore the relationships between natural language constructs and the OO world. By referring to the database schema represented by the UML class diagram, our approach maps natural language constructs with fuzzy semantics into an extended class diagram representation, which will be further transformed into SQL statements for query executions according to the Structured Object Model (SOM) proposed in (Higa and Owei 1991). However, as pointed out in (Velardi 1986), a frequent criticism concern of NLI is that we cannot expect natural language interfaces to act appropriately for every input sentence. Therefore, we hope to limit our work in processing the following types of query sentences:

- Interrogative — “Does Smith supply monitors?”
- Imperative — “List all the suppliers.”
- Declarative — “Smith supplies monitors.” (Which is treated as a question.)

Users should be aware that the system might be unable to provide an answer if their expectations exceed the actual database capabilities, since the information stored in a database is just a precise world subset. If the system cannot make a decision to get answers due to some ambiguity, then user are asked to answer some questions to clarify the ambiguity.

2. Related Work

2.1 Overview of Natural Language Processing

Our prior research (Tseng et al. 1992) has indicated that language-understanding process is commonly divided into three stages. First, the sentence is parsed according to the predefined grammar, then the semantic roles are built and finally these semantic roles are mapped into the specific objects in the real world.

Based on a set of predefined grammar rules, Augmented-Transition-Net (ATN) (Wood 1970) is traditionally used to parse a natural language sentence. By following a method for expression grammars in logic due to Kowalski (Kowalski 1979), Pereira and Warren (Pereira and Warren 1980) have developed a clearer and more powerful formalism named Define

Clause Grammar (DCG). McCord (McCord 1982, 1990) contributes to syntactic analysis and semantic interpretation of natural languages in the framework of logic programming.

For building the semantic roles, Winston (Winston 1992) described a variety of constraints to help establish semantic roles in sentences. These semantic roles reveal how the nouns are related to the verb.

The semantic roles of the sentence are shown in Fig. 2(a). It indicates that the *verb* is ‘supply’, the *subject* is ‘Smith’, the *object* is ‘red keyboards’, and the *with object* is ‘quantity 300’. Finally, these semantic roles can be mapped into specific objects in the real world as Fig. 2(b) illustrates.

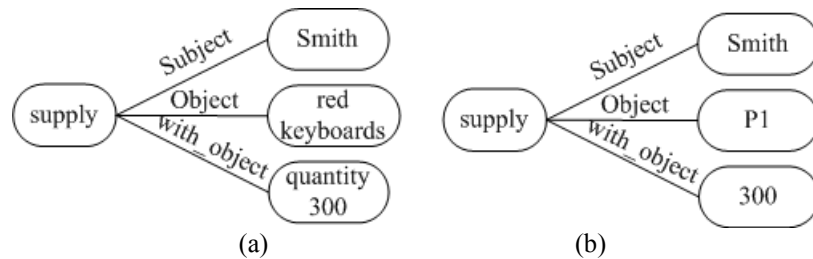


Figure 2: (a) The semantic roles for “Smith supplies red keyboards with quantity 300.”; (b) mapping semantic roles into specific objects.

Our approach follows these three stages and focuses on the mapping from semantic roles to a class diagram schema.

2.2 Review of UML and Class Diagram

The Unified Modeling Language (UML)(Booch et al. 1999; Oestereich 1999) is a graphical notation that provides the conceptual foundation for assembling a system out of components from the 4+1 views and nine diagrams. Our approach mainly employs class diagrams to provide a static view of application concepts, in terms of classes and their relationships including generalization and association. A class diagram (Booch et al. 1999; Muller 1999; Oestereich 1999) shows a set of classes and their relationships. For example, the class diagram of the Suppliers-and-Parts database as shown in Fig. 3 can be shown in Fig. 4. The terms with *italic style* in Fig. 4 indicates the concepts about class diagrams.

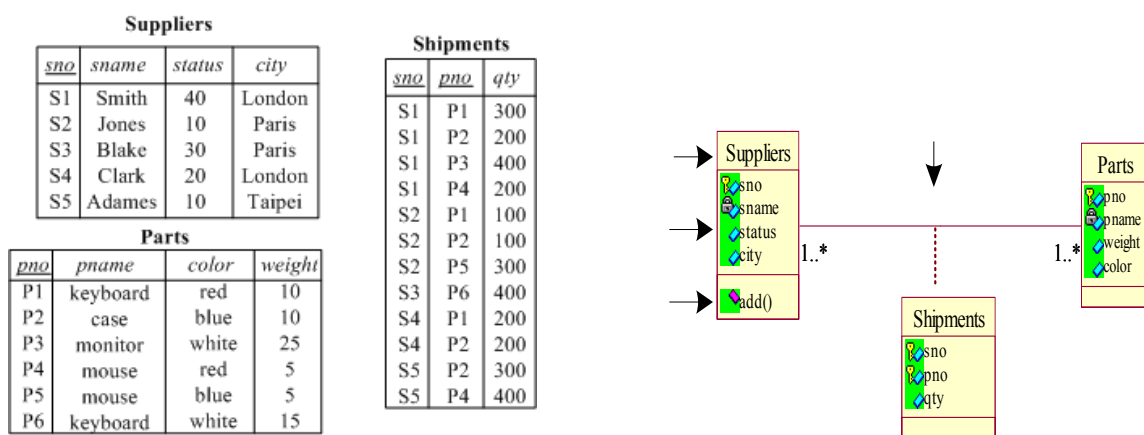


Figure 3: The Suppliers-and-Parts database. Figure 4: A class diagram for the Suppliers-and-Parts Database.

2.3 Traditional Database Model and Fuzzy Database Model

Traditional relational database systems have proved their usefulness in various domains. However, they cannot deal with vague and imprecise data, and that has led to the development of fuzzy databases. In (Umano 1986) and (Zemankova-Leech and Kandel 1984), researchers have pointed out that a fuzzy database can be defined as enhanced relational database that allow *fuzzy attribute values* and *fuzzy truth values* (Takahashi 1993) as shown in *Table 1*:

Table 1: Fuzzy attribute values and fuzzy truth values for *status*.

<i>sno</i>	<i>status</i>	Fuzzy Attribute Values	Fuzzy Truth Values
S1	40	good	0.8
S2	10	bad	0.2
S3	30	good, middle	0.6
S4	20	middle, bad	0.4
S5	10	bad	0.2

2.4 Fuzzy Sets, Membership Function, Linguistic Variables and Fuzzy Terms

The theory of fuzzy sets (Zadeh 1965) provides a means for the representation of imprecise and vague data through membership functions. It has been used widely in various fields, like decision-making, and psychology.

The concept of *linguistic variable* (Zadeh 1975) is a variable whose values are not numbers but words or sentences in a natural language. The motivation for the use of words or sentences rather than numbers is that linguistic characterizations are, in general, less specific than numerical ones.

The use of such a linguistic variable (Schmucker 1984) allows a precise modeling of imprecise statements like ‘old’ and ‘young’. Linguistic variables allow for the easy and natural specification of values for imprecise concepts. For instance, if the values of “age” are linguistic (e.g., old, young, ...) rather than numerical (e.g. 20, 21, ...), then the “age” is a *linguistic variable*. In this case, “old” and “young” are simply called *fuzzy terms*. To take a simple example, an attribute value of *status* can be represented by the membership function of $\mu_{status}(u)$ as *Table 2* shows.

Table 2: the mapping between an attribute value status and μ_{status} .

<i>status</i>	0	10	20	30	40	50
μ_{status}	0	0.2	0.4	0.6	0.8	1

To present an example of $\mu_{status}(u)$, we define its corresponding fuzzy terms “good”, “middle” and “bad” in *Table 3*.

Table 3: the mapping between μ_{status} and its corresponding fuzzy terms.

μ_{status}	[1, 0.6]	[0.6, 0.4]	[0.4, 0]
Fuzzy Term	good	middle	bad

2.5 Structured Object Model (SOM)

Higa and Owei (Higa and Owei 1991) indicated a conceptual data model driven programming tool for database query processes. They developed Structured object Model (SOM) as an analysis, design and navigation tool for database applications.

Assume that the SOM diagram is as shown in *Fig. 5*. The root object in SOM is a dummy object, which represents the entire schema in the database. *Fig. 5* includes an added feature of

double-headed arrows to indicate direct logical connectivity between object pairs. Attached to these arrows are the attributes through which the objects are connected.

2.5.1 The Logical Adjacency Matrix and the Connectivity Matrix

The Logical Adjacency matrix (LAM) indicates direct connectivity between pairs of objects. Fig. 6 is the adjacency matrix, A , of Fig. 5, where the adjacency between pairs of objects is not taken in the sense of the physical connections shown in the figure. A zero element in the matrix implies that there is no direct connection, where the two objects have no relationship, or that an object is trivially connected to itself.

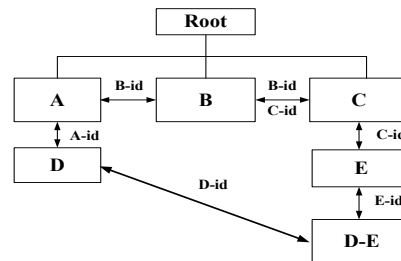


Figure 5: The SOM Diagram.

Based on the LAM, a connectivity matrix, R , for the SOM diagram can be determined. Matrix R captures the information on whether or not any two objects are connected, either directly or indirectly. The elements of R indicate a path length connecting the two corresponding objects.

Fig. 7 is the connectivity matrix, R , obtained from the A matrix in Fig. 6. By comparing these two matrices, it is easy to observe that A is a special case of R .

	A	B	C	D	E	D-E
A	0	1	0	1	0	0
B	1	0	1	0	0	0
C	0	1	0	0	1	0
D	1	0	0	0	0	1
E	0	0	1	0	0	1
D-E	0	0	0	1	1	0

Figure 6: LAM for the SOM diagram.

	A	B	C	D	E	D-E
A	0	1	2	1	3	2
B	1	0	1	2	2	3
C	2	1	0	3	1	2
D	1	2	3	0	2	1
E	3	2	1	2	0	1
D-E	2	3	2	1	1	0

Figure 7: Connectivity Matrix for the SOM Diagram.

2.5.2 The Mapping of Object Path and Query Statement

When end-users have a query request, the query processing can be executed through the following steps. First, the system extracts a subgraph (or sub-tree), called a validation sub-tree, of the SOM diagram. Then, the validation sub-tree consists solely of the relevant object connecting the *source* (the object or set of objects, with their association valued-attributes, which are stated in the query, and to which the object and attributes of interest are related) and *target* (the object of ultimate destination). Then, system internally maps the validated sub-tree to the connectivity matrix, to determine if a logical path exists between the source and target. Finally, if a logical path exists, the system internally maps the chosen logical path to an equivalent SQL-type statement, which is then executed.

We illustrate this process by referring the SOM diagram of Fig. 5. Suppose the query is such that we identify the source object as E, with primary attribute value 'E1', and the target object as D. The attribute of interest is the D-descriptor of object D. The query session

proceeds as outlined below.

User Specifies *Source*:
 Object: E
 Attribute: E-id = 'E1'
 User Specifies *Target*:
 Object: D
 Attribute: D-descriptor

System returns the validation sub-tree as shown in Fig. 8. By matrix *R* as shown in Fig. 7, the system determines that logical path as:

$$E \rightarrow D-E \rightarrow D$$

Finally, the system performs the query processing as depicted in Fig. 9.

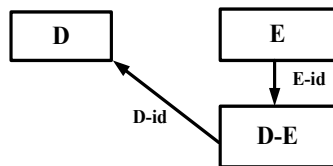


Figure 8: The validation sub-tree.

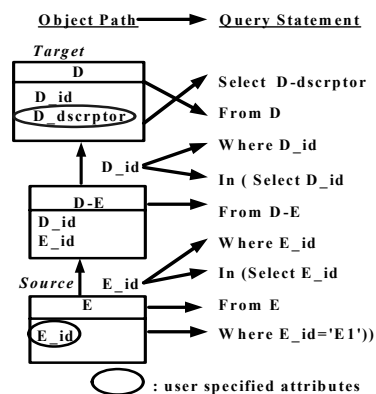


Figure 9: Object Path-Query Statement Mapping.

3. The Processing Model and Mapping Process

The natural language processing model is shown in Fig. 10. In this model, data can be distinguished into “crisp data” and “fuzzy data”:

1. Crisp data: Traditional databases store crisp data. For example, the attribute values of Age are precise numbers (e.g. Age can be 20, 30 or 40).
2. Fuzzy data: Fuzzy data are represented as fuzzy terms that the words or phrases in queries are the name of variables. For example, if “Age” is a linguistic variable, it may have fuzzy terms like “young” or “old”.

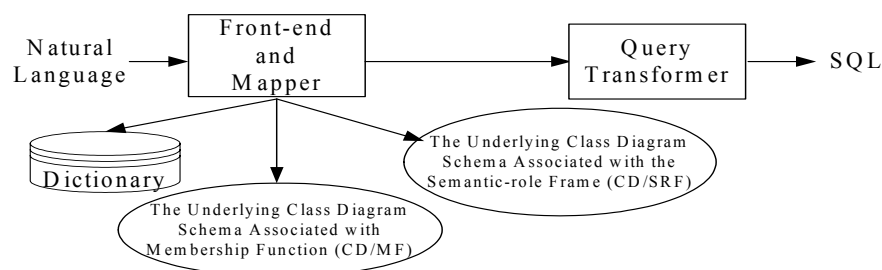


Figure 10: The processing model.

3.1 The Processing Model

In the processing model, there is a front-end parser/mapper to parse the English queries and map them into the underlying class diagram schema. The parsing and mapping process may refer to

1. The dictionary,
2. The underlying class diagram schema associated with the semantic-role frame (CD/SRF) as *Fig. 2* illustrates, and
3. The underlying class diagram schema associated with pre-defined membership functions (CD/MF). This database is used to allow users to express their queries with fuzzy predicates. It stores various membership functions and the description of the correspondence between fuzzy attribute values and fuzzy truth values.

After the parsing phase, the query is decomposed into:

1. *Semantic roles*, each of which is composed of a headnoun and some modifiers (a headnoun is the main noun in a noun phrase; for example, ‘the London suppliers’ has a headnoun ‘suppliers’ and the other noun ‘London’ is a modifier which modifies the headnoun).
2. The *verb* that relates these semantic roles.
3. *Fuzzy modifiers* and *fuzzy terms*, each fuzzy modifier is a linguistic variable and fuzzy term is an adjective to describe an object. For instance, a query containing ‘the suppliers with status good’ has a fuzzy modifiers (or a linguistic variable) ‘status’ to describe the headnoun ‘suppliers’ and a fuzzy term ‘good’.

By referring the dictionary (see Section 3.1.1), each of the semantic roles is mapped into a class and its headnoun and modifiers are mapped into the corresponding attributes of that classes based on CD/SRF (Section 3.1.2). The verb that relates these semantic roles is mapped into the relationship among these semantic roles. Each of fuzzy modifiers is mapped into a fuzzy class and its adjective is mapped into the corresponding fuzzy terms (FT) class of that classes based on CD/MF (Section 3.1.3).

In Table 4, we summarize the correspondence between English sentence structures and class diagram constructs, which simplifies the 11 translation rules addressed in (Chen 1983). Fig. 11 and Fig. 12 illustrate the mapping mechanisms for association relationships and fuzzy queries, respectively.

Our processing model makes the following assumptions:

1. Each class is augmented with an object identifier (OID), which is underlined in *Fig. 3*. Any reference to an OID is interpreted as a reference to the corresponding object.
2. Each association will be represented as an association class, where an OID is composed of the OIDs of the involved classes.
3. All fuzzy classes that modify the same class *C*, which corresponds to a semantic role, have the same OID attribute with *C*. Besides, all FT classes have the same OID attribute with the fuzzy class on which they are attached. That is, an instance of a fuzzy class and the associated FT classes use the same OID to relate to the class instance they modify.

Table 4: Correspondence between English sentence structures and class diagram constructs.

English sentence structures	Class diagram constructs
Common Noun	Class name/ Fuzzy class name
Proper Noun	Object name
Transitive Verb	Association name/ Association class name
Adjective	The attribute of class/object/FT class name
Adverb	The attribute of association class

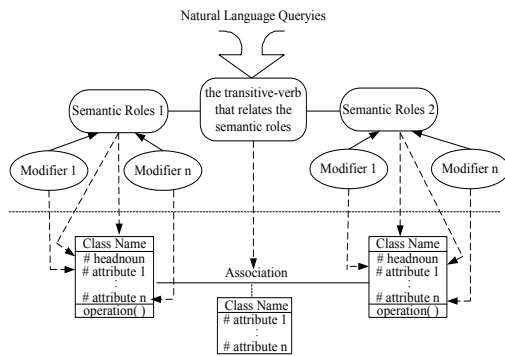


Figure 11: The mapping mechanism of association relationship.

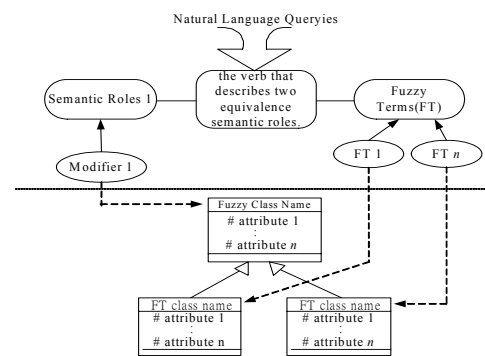


Figure 12: The mapping mechanism of fuzzy queries.

3.1.1 The Dictionary

The linguistic knowledge that enables the semantic roles of a verb to be mapped into the correct attributes of the corresponding relationship is stored in the dictionary. For the Suppliers-and-Parts database, the knowledge for a verb occurred in a natural language query can be shown in *Table 5*. For example, according to *Fig. 4*, the transitive verb ‘supply’ (and its synonyms) could be mapped into the association **supply**, which corresponds to the class **Shipments**, and its corresponding semantic roles can be mapped in the schema are a subject, an object, and a with_object (which is regarded as an adverb phrase that modifies the verb). The subject and object associate the classes **Suppliers** and **Parts**, respectively, and are mapped into the corresponding OIDs.

Besides, the linguistic variable that enables the fuzzy modifier to be mapped into the fuzzy class of database of various membership functions is stored in the fuzzy dictionary. For the Suppliers-and-Parts database, we have shown in *Table 6* that the fuzzy modifier ‘status’ occurred in a natural language query can be mapped into a fuzzy classes ‘Status’ and three fuzzy term classes ‘**Good**’, ‘**Middle**’, and ‘**Bad**’ in the fuzzy dictionary.

Table 5: The dictionary structures of Suppliers-and-Parts database.

Verbs	Class	Semantic Roles	Corresponding Attribute(s)
supply (and its synonyms)	Shipments	Subject Object With_object	<i>sno</i> (the OID of Suppliers) <i>pno</i> (the OID of Parts) <i>qty</i>

Table 6: The fuzzy dictionary structures of Suppliers-and-Parts database.

Fuzzy Attribute	Corresponding Fuzzy Classes	Corresponding FT Classes
<i>status</i>	Status	Good Middle Bad

3.1.2 The Schema and the Semantic-Role Frame Represented in Class Diagram

For each class, attributes are identified as the *headnoun* and the *modifiers* that modify the

headnoun. For example, the attributes corresponding to the headnouns of **Suppliers** and **Parts** are *sname* and *pname*, respectively. Other attributes are identified to be the corresponding modifiers. The above information will be encoded into the original class diagram. For example, the schema and the semantic-role frame representing the Suppliers-and-Parts database are shown in Fig. 13.

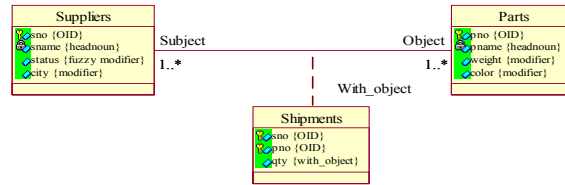


Figure 13: The schema and the semantic-role frame of the Suppliers-and-Parts database.

3.1.3 The Schema and Membership Function Represented in Class Diagram

Each fuzzy modifier can be mapped into a membership function to define its corresponding fuzzy class. We assume the name of a fuzzy modifier and its corresponding fuzzy class is the same. For instance, a fuzzy modifier *status* in the class **Suppliers** of the Suppliers-and-Part database can be mapped into the fuzzy class **Status**.

In our example, we regard ‘status’ as a linguistic variable and define its fuzzy term classes as **Good**, **Middle**, and **Bad**. The class diagram of fuzzy class **Status** and its fuzzy term classes are shown in Fig. 14 (please also refer to Table 1) and Fig. 15, respectively.

Status	
sno	MFV
S1	0.8
S2	0.2
S3	0.6
S4	0.4
S5	0.2

Good	Middle	Bad
sno	sno	sno
S1	S3	S2
S3	S4	S4
		S5

Figure 14: The fuzzy class **Status** and its term classes.

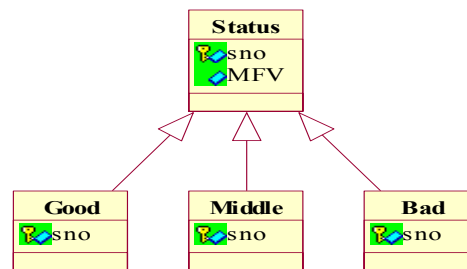


Figure 15 : The class diagram schema of the fuzzy class **Status** and its fuzzy term classes.

3.2 Description of the mapping process

The natural language mapping process can be divided into the following stages:

1. Mapping the only transitive verb into an association relationship. Since a query sentence has only one transitive verb, by referring to the dictionary, we can map the transitive verb of a query sentence into the corresponding association and its semantic roles into the corresponding attributes. Besides, an adverb (or adverb phrase) that modifies the verb can be mapped into the corresponding attribute of the association class. For instance, the query “Does Smith *supply* keyboards with quantity 300?” has the transitive verb *supply*, and by referring to the dictionary shown in Table 5, the semantic roles can be mapped into the corresponding attributes as depicted in Fig. 16.

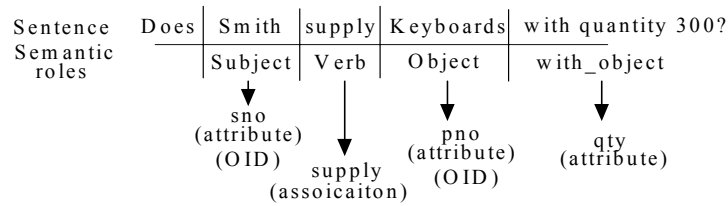


Figure 16: The dictionary mapping for “Does Smith supply nuts with quantity 300?”

- Mapping noun phrases into classes and attributes. The semantic roles of a verb in a sentence can be mapped into those OIDs of involved classes. The noun phrases may consist of more than one noun modifier (which can be adjectives or nouns) or relative clauses that modify the headnoun. For example, “the London supplier” and “the red parts” are two noun phrases and have the noun modifiers ‘London’ and ‘red’, respectively. According to the dictionary, the interpretation of these noun modifiers described above can be mapped as *Fig. 17* illustrates.

Noun Phrase	Corresponding Class	Interpretation of the Noun Modifier
the London suppliers	Suppliers	City = ‘London’
The red parts	Parts	Color = ‘red’

Figure 17: Mapping noun modifiers into attributes.

- Mapping fuzzy noun phrases into fuzzy classes and fuzzy term classes. If a noun phrase consists of one fuzzy modifier, this fuzzy modifier can be mapped into corresponding fuzzy classes and fuzzy term classes. For example, a noun phrases as “the suppliers with status good” (or “the good supplier”) have the fuzzy modifier ‘status’. This noun fuzzy modifier can be interpreted into fuzzy classes **Status**, and the adjective ‘good’ that modifies this fuzzy modifier can be mapped into fuzzy term class **Good**. According to the fuzzy dictionary, the interpretation of these noun modifiers described above can be mapped as *Fig. 18* illustrates.

Noun Phrase	Corresponding Fuzzy Modifier	Corresponding Fuzzy Classes	Corresponding FT Classes
good status	status	Status	Good

Figure 18: Mapping fuzzy noun modifiers into fuzzy classes and fuzzy term classes.

Example 3.1: For the query $Q =$ “Is Smith a good supplier who supplies red parts?” the corresponding mapped class diagram of this query is shown in *Fig. 19*.

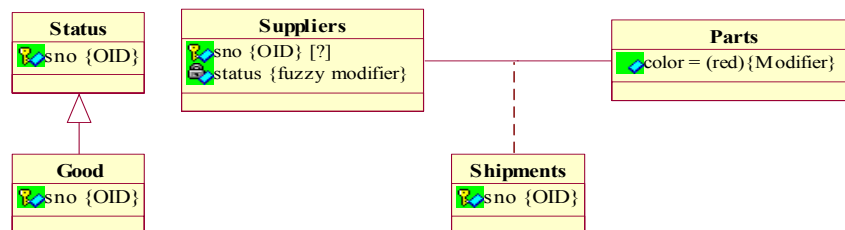


Figure 19: Mapping “Is Smith a good supplier who supplies red parts?” to class diagram.

4. Query Transformer

In this section, we employ SOM to transform English queries into SQL statements through the corresponding mapped class diagrams. The database schema of Suppliers-and-Parts can be mapped into the SOM schema as shown in Fig. 20.

We follow *Example 3.1* to illustrate how to transform the corresponding class diagram into SQL statements. By referring to Fig. 20, Fig. 19 can be converted to an SOM diagram as depicted in Fig. 21, which have the corresponding LAM as shown in Fig. 22. According to Fig. 21, the connectivity matrix for this SOM diagram can be determined in Fig. 23, which can be employed to realize whether or not any two objects are connected, either directly or indirectly.

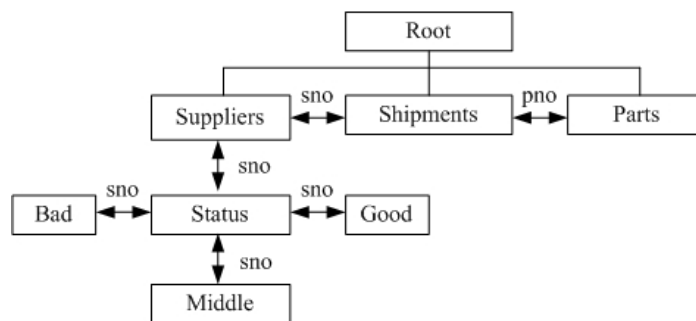


Figure 20: The SOM diagram of the Suppliers-and-Parts database.

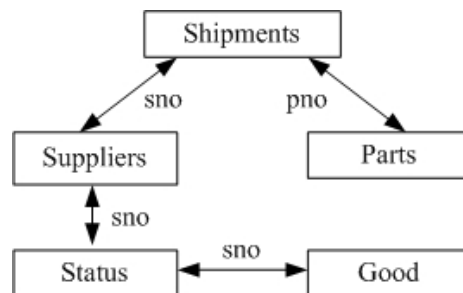


Figure 21: the SOM diagram of Example 3.1.

	Suppliers	Status	Good	Shipments	Part
Suppliers	0	1	0	1	0
Status	1	0	1	0	0
Good	0	1	0	0	0
Shipments	1	0	0	0	1
Parts	0	0	0	1	0

Figure 22: LAM for the SOM Diagram of Fig. 21.

	Suppliers	Status	Good	Shipments	Parts
Suppliers	0	1	2	1	2
Status	1	0	1	2	3
Good	2	1	0	3	4
Shipments	1	2	3	0	1
Parts	2	3	4	1	0

Figure 23: Connectivity Matrix for the SOM Diagram of Fig. 21.

Now, we explain the query transformation process by the following steps:

1. According to the class diagram obtained from natural language query, the query transformer extracts a valid subgraph from the SOM diagram, such that the subgraph consists of the relevant objects connecting the *source* and *target*. For our example, according to *Fig. 19*, *Fig. 20* is the corresponding SOM diagram and *Fig. 21* is a valid subgraph of *Fig. 20*.
2. The query transformer derives the connectivity matrix from the validated subgraph to determine if a logical path exists between the source and target. If the path does not exist then it proceeds to Step 6. For example, the validated subgraph, *Fig. 21*, can be mapped into *Fig. 23*.
3. If a logical path exists, the query transformer maps the chosen logical path to equivalent SQL statements for execution. For instance, the source object of the query specified in *Example 3.1* can be identified as **Suppliers**, **Status**, **Good**, and **Parts** (with attribute value *color* = 'red'). Besides, the target object can be recognized as **Shipments**. The attribute of interest is the attribute *sno* of **Shipments**. The query session thus proceeds as below.

User Specifies Source:
Object: **Suppliers**, **Status**, **Good**, and **Parts**
Attribute: *color* = 'red', *sname* = 'Smith'
User Specifies Target:
Object: **Shipments**
Attribute: *sno*

The query transformer returns the valid subgraph (see *Fig. 24*). By *Fig. 23*, the query transformer determines the logical path as:

Shipments ← Suppliers ← Status ← Good ← Parts

4. Based on the result derived in Step 3, the query transformer can derive the query statement from the object path as *Fig. 25* illustrates.
5. Then, the system presents the execution result to the end-user. For our example, the system will show 'YES' (refer to *Fig. 3* and *Fig. 14*) to the end-user.
6. If the system cannot make a decision to get answers, the question will be resolved by the following two principles. First, when some logical path exists between the source and the target, users are asked to answer some questions to clarify the ambiguity. Otherwise, if there is no logical path exists between the source and the target, an error message will be presented to the end-user.

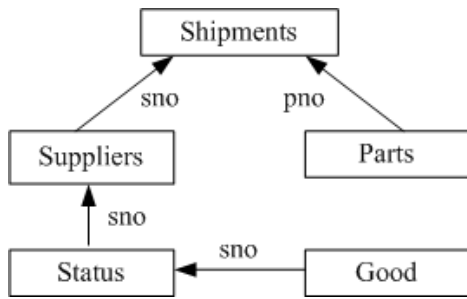


Figure 24: The validation subgraph.

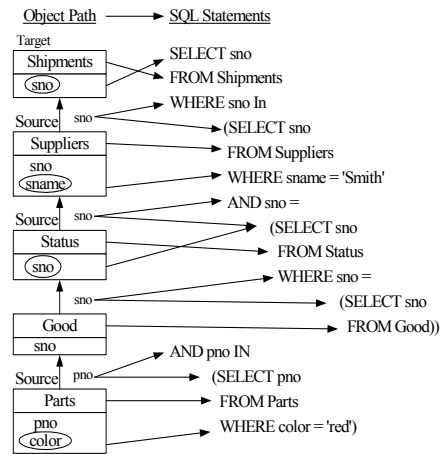


Figure 25: the mapping process

5. Conclusions and Future Directions

5.1 Conclusions

Natural language interfaces are promising for today's database applications. This can be realized from some database vendors are eager to empower their products with the ability of issuing English queries for database access. For example, Microsoft SQL Server 2000 is equipped with an English Query module to do this. However, there are still limitations of using the E-R diagram to be the logical form when mapping natural language constructs into database schemas. In this paper, we propose to use the class diagram of UML instead of E-R diagram to capture more natural language semantics for querying databases. We employ SOM methodology by extending the class diagram representations to capture natural language semantics and describe a processing model for the query transformation process. In this processing model, when the target database is changed we only have to change the dictionary, the CD/SRF, and the CD/MF in *Fig. 10*. The front-end parser/mapper and the query transformer remain unchanged.

Furthermore, we have studied the inter-relationship between natural language constructs and the class diagram conceptual schema. The basic parts of English sentences can be mapped into class diagram schemas in a natural way. Our approach allows users to intermingle fuzzy queries with crisp queries and provide a natural way to retrieve data from databases.

Finally, if the underlying schema of a database was pre-existing but not structured by class diagram approach, then the schema can be restructured by class diagram through defining the design views by the UML approach to achieve our work.

5.2 Future Directions

We will extend our future research into the following aspects:

1. Capture the semantics of natural language queries with fuzzy hedges: By combining present proposed fuzzy database frameworks, for example (Zadeh 1978), (Takahashi 1993) and (Bosc et al. 1988), our work can be extended to process a natural language query involving a modifier like 'almost', 'very', or 'nearly'. This combination is served as a step toward analyzing the use of modifiers, which are fuzzy in natural, to communicate with fuzzy databases.

2. Consider to output query results in XML formats: XML is rapidly becoming the standard scheme for storing and exchanging structured and semi-structured data and information between organizations. Therefore, we will further stretch our work to transform the query results into XML documents for sending or interchanging information across the Internet.

6. References

- Alavi, M. and Leidner, D. E. "Knowledge Management Systems: Issues, challenges, and benefits," *Comm. of the Association for Information Systems*, 1, article 7, 1999.
- Alavi, M. and Leidner, D.E. "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues," *MIS Quarterly* (25:1), 2001, pp. 107-136.
- Booch, G., *Object Solutions: Managing the Object-Oriented Project*, Benjamin/Cummings Publishing Company, Inc., 1996.
- Booch, G., Rumbaugh, J. and Jacobson, I. *The Unified Modeling Language User Guide*, Addison-Wesley, 1999.
- Bosc, P., Galibourg, M. and Hamon, G. "Fuzzy Querying with SQL: Extensions and Implementation Aspects," *Fuzzy Sets and Systems* (28), 1988, pp. 333-349.
- Chen, P.P. "English sentence structure and Entity-Relationship Diagrams," *Information Sciences* (29), 1983, pp. 127-149.
- Date, C.J., *An Introduction to Database Systems 7th ed.*, Addison-Wesley, 2000.
- Higa, K. and Owei, V. "A data model driven database query tool" *Proc. the 24th Annual Hawaii Int'l Conf. on System Sciences* (3), 1991, pp. 53 –62.
- Isoda, S. "Object-Oriented Real-World Modeling Revisited," *Journal of Systems and Software* (59:2), 2001, pp. 153-162.
- Jacobson, I. *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992.
- Kowalski, R.A. *Logical for Problem Solving*, North-Holland, Amsterdam, 1979.
- McCord, M.C. "Using Slots and Modifiers in Logic Grammars for Natural Language," *Artificial Intelligence* (18:3), 1982, pp.327-367.
- McCord, M.C. "Slot grammar: A system for simpler construction of practical natural language grammars," *Natural Language and Logic*, 1990, pp. 118-145.
- Métais, E. "Enhancing information systems management with natural language processing technique," *Data & Knowledge Eng.* (41:2-3), 2002, pp. 247-272.
- Moreno, A.M. and van de Riet, R.P. "Justification of the Equivalence between Linguistic and Conceptual Patterns for the Object Model," *Proc. Int'l Workshop on Applications of Natural Language to Information Systems*, June 27-29, 1997.
- Muller, R.J. *Database Design for Smarties: Using UML for Data Modeling*, Morgan Kaufmann, 1999.
- Oestereich, B. *Developing Software with UML Object-Oriented Analysis and Design in Practice*, Addison-Wesley, 1999.
- Pereira, F.C.N. and Warren, D.H.D. "Definite Clause Grammars for Language Analysis—A Survey of the Formalism and a Comparison with Augmented Transition Networks," *Artificial Intelligence* (13:3), 1980, pp 231-278.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorenzen, W. *Object-Oriented Modeling and Design*, Prentice-Hall, Inc., 1991.
- Schmucker, K. J. *Fuzzy sets, natural language computations, and risk analysis*, Computer Science Press, Rockville MD, 1984.
- Snodgrass, R.T. *Developing Time-Oriented Database Applications in SQL*, Morgan Kaufmann Publishers, Inc., San Francisco, July 1999.
- Takahashi, Y. "Fuzzy database query languages and their relational completeness theorem," *IEEE*

- Trans. on Knowledge and Data Eng.* (5:1), 1993, pp. 122 –125.
- Tseng, F.S.C, Chen, A.L.P and Yang, W.P. “On mapping natural language constructs into relational algebra through E-R representation,” *Data & Knowledge Engineering* (9:1), 1992/93, pp. 97-118.
- Tseng, F.S.C. and Chen, C.L. Extending the UML Concepts to Capture Natural Language Semantics for Database Access, submitted for publication.
- Umano, M. “Relational Algebra in fuzzy database,” *IEICE Tech. Rep. DE86-4* (86:192), 1986, pp. 1-8.
- Velardi, P. “Natural Language Interfaces to Databases: Features and Limitations,” *Proc. the 7th International Conference on Entity-Relationship Approach*, 1986.
- Winston, P.H., *Artificial Intelligence*, Addison-Wesley, 1992.
- Wood, W.A. “Transition Network Grammars for Natural Language Analysis,” *Communications of the ACM* (13:10), Oct. 1970, pp. 591-606.
- Yazici, A. and Cibiceli, D. “An access structure for similarity-based fuzzy databases” *Information Sciences*, Vol. 115, No. 1-4, Apr. 1999, pp. 137-163.
- Zadeh L. A. “Fuzzy Sets,” *Information and control* (8), 1965, pp. 338-353.
- Zadeh, L.A. “The concept of a linguistic variable and its application to approximate reasoning (I),” *Information Sciences* (8:3), 1975, pp. 199–249.
- Zadeh, L.A. “PRUF-a meaning representation language for natural language,” *International Journal on Man-Machine Studies*, Vol. 10, 1978, pp.395-460.
- Zemankova-Leech, M. and Kandel, A. *Fuzzy relational database-A key to expert system*, Verlag TUV Rheinland, Cologne (Germany), 1984.

Acknowledgement: This research was supported by the National Science Council, ROC, under contract No. NSC 92-2416-H-327-005.