

Semantic Analysis of Natural Language Queries for an Object Oriented Database

Bentamar Hemerelain¹, Hafida Belbachir²

¹Oran University, Wilaya of Oran, Algeria; ²University of Science and Technologie of Oran (USTO), Wilaya of Oran, Algeria.
Email: hemerelain@yahoo.fr; hemerelain@hotmail.com

Received September 3rd, 2010; revised October 15th, 2010; accepted October 19th, 2010.

ABSTRACT

This paper presents the semantic analysis of queries written in natural language (French) and dedicated to the object oriented data bases. The studied queries include one or two nominal groups (NG) articulating around a verb. A NG consists of one or several keywords (application dependent noun or value). Simple semantic filters are defined for identifying these keywords which can be of semantic value: class, simple attribute, composed attribute, key value or not key value. Coherence rules and coherence constraints are introduced, to check the validity of the co-occurrence of two consecutive nouns in complex NG. If a query is constituted of a single NG, no further analysis is required. Otherwise, if a query covers two valid NG, it is a subject of studying the semantic coherence of the verb and both NG which are attached to it.

Keywords: Query, Nominal Group, Natural Language, Object Oriented Data Base, Semantic Validation

1. Introduction

On the earliest and most widely studied areas of natural language processing is the development of a Natural language interfaces to databases (NLIDB) [1-7]... A NLIDB allows users to input a query in a natural language such rather than in a formal query language and in conceptual terms particular to their understanding of the database application domain. In most works, the underlying database is assumed to be relational. In our work, we are interested in building a NLI for an object-oriented database (OODB). Unfortunately, the relational model and the object model are fundamentally different. Relational database (RDB) systems are based on two-dimensional tables in which each item appears as a row. Relationships among the data are expressed by comparing the values stored in these tables. The object model is based on the tight integration of code and data, flexible data types, hierarchical relationships among data types, and references [8].

This paper is concerned with the semantic validation of natural language query (NLQ) for an OODB. The natural language considered here is French.

A considerable amount of work has been published on the semantics of NLQ. Works, usually carried out by Computer Scientists, can be classified into two categories:

1) work based on established linguistic theories which are modified and/or extended such as [9-12]; 2) new ad hoc but efficient semantics that are not based on established linguistic theories. Examples are quoted by [2]. Some recent works are [13-15].

Our approach falls into category 2) and is designed on the principle of [13] who introduce the idea of semantically tractable (ST) questions. As their database is in the form of RDB, they define the ST questions as questions where the words correspond to relations (tables), attributes and values. The nature of the OODB introduces some new challenges, beyond these addresses for RDB. In this paper, we deal with the concept of class to model complex data, simple attributes, complex (composite) attributes with classes as their domains, and two special kinds of associations between classes: 1) inheritance which is the most powerful concept of object oriented. It is a mechanism of reusability: the news classes, known as sub-classes inherit attributes of the pre-existing classes, which are referred to as super classes; an inheritance relationship is created between the sub-class and the super-class. 2) reference: since a class C may have a complex attribute with domain C', a reference relationship can be established between C and C'.

Our center of interest is the analysis of NLQ consisting of one or two NGs articulating around a verb. At this

step of analysis, we consider the NG constituted of one or several keywords and we ignore the DB independent words (article, relative pronoun,...). A keyword can be an application dependent noun or a value. The former matches a class name, a simple attribute name or a composite attribute name in an OODB.

In a NLQ, the verb is created by the DB administrator. It expresses the link which exists between a class and another class or between a class and one of its attributes.

The application dependent nouns and the verbs with their semantic features are stored in a lexicon. This last includes inflection forms of nouns and conjugated forms of verbs. It is also augmented with relevant synonyms. This addition makes it possible to the user to write a query in various natural ways.

Heritage and reference relationships are added to lexicon allowing the safeguarding of the heritage and reference link of the OODB. These Heritage and reference relationships with the lexicon form the natural language database (NLDB).

The semantic analysis of a query first deals with identification of individual keywords in the NGs. It uses some simple semantic filters to determine their semantic value. If the NG is atomic and the single keyword identified, no further analysis is required. On the contrast, it is necessary to verify if the complex NG respects the conceptual constraints of the domain DB. The conceptual constraints mainly refer to inheritance and reference relationships and are function of the semantic value of the keywords and their combination in the NG. If the NLQ is constituted of a single NG, no supplementary treatment is necessary. If the NLQ covered two valid NGs, it is a subject of studying the semantic coherence of the verb and both NGs which are attached to him.

The paper is organized as follows: The Section 2 describes the lexicon and defines the inheritance and reference relationships in a NLDB. The Section 3 is divided into three parts. In the part 1, semantic filters are defined to identify keywords. With the introduction of the semantic coherence constraints of keywords, the part 2 discusses the process of NG's validation. Part 3 concerns the semantic coherence of the verb with its subject and its object. In Section 4, an example of illustration is treated to simulate the tools developed in preceding section. To understand this example and those that are quoted in the text; we give an extract of the database prototype which is used in our experimentation. The results of the experimentation are reported in Section 5. A conclusion is given in Section 6.

2. Natural Language Database

The natural language database (NLDB) is a translation of the object oriented database (OODB). The NLDB con-

tains a lexicon and inheritance and reference relationships.

2.1. Lexicon

A natural language query consists of two parts: an interrogative particle and a phrase. A phrase is a succession of nominal and verbal groups. A nominal group (NG) in the natural language query (NLQ) consists of one or several meaningful words. The latter can be an application dependent noun, a verb or a value.

2.1.1. Definitions

Application dependent noun

The former matches a class name, a simple attribute or a composite attribute name in an OODB.

Each application dependent noun is represented in the OODB as a variable whose name might be meaningful according to the naming conventions but not necessarily the same word in natural language (NL) (e.g. ENS, X). Thus, we generate, for each variable declared in an OODB, a word in NL (NW) (e.g. NW (ENS) → enseignant [teacher]; NW(X) → personne [person]).

Due to this, to each variable declared in the database, corresponds, a word family. It includes all of the inflection forms of the word (masculine singular (ms), masculine plural (mp), feminine singular (fs),...) (e.g. enseignant [teacher] → {enseignant (ms) [teacher], enseignants (mp) [teachers], enseignante (fs) [teacher], enseignantes (fp) [teachers]} and the set of its synonyms. (e.g. enseignant → {maitre,...} [teacher → {schoolmaster,...}])).

Verb

The verb of the query expresses the link between two classes or a class and one of its attribute. Then, the verb is transitive. To each created verb, corresponds conjugate forms (3rd person of the singular (s), 3rd person of the plural (p)) (e.g. habiter [to live] → {habite (s) [lives], habitant (p) [live]} and the set of its synonyms (e.g. (habiter) → {résider, demeurer,...} [to live → {to reside, to dwell,...}])).

All the nouns and verbs usually depend heavily on the application. They constitute the lexicon.

The initial set of synonyms is created with electronic dictionaries such that Wolf and Crisco and then proposed to expert individual who can modify it to reflect the specific senses of the DB application domain.

The sibling relation that exists between the words of the same family permits the database requester to use any of them without affecting the meaning of the query. Thus, a family of words is represented by a single term called the *stem* which can be the first generated word in natural language.

Value

On the other hand, a value in the query is an instance of a simple attribute. It is identified via a dialogue with the database requester. If an attribute is designated by the DB administrator as key then its value is said to be key and non-key otherwise. A key value can specify a class. For example, in the query “what is the salary of Linda”, Linda specifies the class “Employee” then it is a key value. However, in the query “what is the name of the module which has the coefficient 4”, the value ‘4’ is not key value because it specifies no class.

2.1.2. Description

Application dependent noun

Depending on the relation of the word to the database being handled, an application dependent noun is described, in the lexicon, by its stem and its semantic value which can be ‘C’ if it is about a class, ‘SA’ of a simple attribute or ‘CA’ of a composite attribute.

In the description of a simple attribute, we add the stem of the class containing it and in the case of a composite attribute, the stem of the referenced class by the composite attribute. For example, the composite attribute ‘matières’ [matters] is described by its stem ‘matière’ [matter], its semantic value ‘AC’ and the stem of its referenced class ‘module’ [module].

Verb

A verb is described by its subject which is the class that contains it and its object which is either a declared attribute in the subject class or a referenced class by a composite attribute of the subject class. It is formally described by its stem, the stem of the subject entity and the stem of the object entity. For example, the verb ‘habiter’ [to live] is described by its stem ‘habiter’ [to live], the stem of the subject entity ‘personne’ [person] and the stem of the object entity ‘ville’ [town].

2.2. Inheritance Relationship

The inheritance relationship in the NLDB is an extension of the traditional inheritance relationship ‘HERITE’ in an OODB and is given by the relation *Herite**. It allows handling the NLQ on a given class using the properties of the class itself or those of its super-classes and sub-classes.

The extended inheritance relationship allows moving in both directions up and down the hierarchy of the classes.

The direct inheritance relationship ‘herite’ is defined as follows:

Let CLASSE be classes stems of a NLDB.
 herite: CLASSExCLASSE → {0,1}
 (x,y) → herite (x,y)

{

herite (x, y): = 1 if $NW^{-1}(x) \text{ HERITE } NW^{-1}(y)/*$
 $NW(v) = x/'v'$ is a variable in the OODB and ‘x’ is the word corresponding in natural language */

herite (x, y):= 0 otherwise;

}

The relation *Herite** is defined as follows:

Herite*: CLASSExCLASSE → {0,1}
 (x,y) → Herite* (x,y)

{

Herite*: = 1

if (x = y) or (x ≠ y and (herite (x, y)

or herite (y, x))

or (x ≠ y and ∃ C1, C2,... Cn ∈ CLASSE/

herite (x, C1)

and (∀i ∈ [1, n-1] herite (Ci, Ci+1))

and herite (Cn, y)

or (x ≠ y and ∃ C1, C2,... et Cn ∈ CLASSE/

herite (y, Cn)

and (∀i ∈ [1, n-1] herite (Ci+1, Ci)

and herite (C1, x))

Herite*: = 0 otherwise;

}

2.3. Reference Relationship

Let M1 be a class, M2 its composite attribute, M3 the referenced class and $R_{1,2}$ a relation connecting M1 and M2.

The relation $R_{1,2}$ is also applicable between the class of definition M1 and the referenced class M3. It represents a direct reference link between these two classes. This relation is called the reference relationship in a NLDB. It is an extension of the traditional reference relationship in an OODB. It allows generating the verbs of the NLDB.

We notice that the existence of a reference relationship, does not base itself on a criterion according to these links, but that it is random, thus the solutions in that case relying on a linguistic analysis which indicates for all the roads of (direct) reference, the existence or the non-existence of a verb of relation.

They translate direct reference links between the classes. In that case, the existence of the verb can create a indirect reference link (e.g. travailler (chercheur, projet) [to work (searcher, project)] and appartenir (projet, laboratoire) [to belong (project, Laboratory)] ⇒ travailler (chercheur, laboratoire) [to work (searcher, Laboratory)]).

NOTE. - The deducted relations can be calculated by combining the reference links and the links of inheritance (e.g. assurer (enseignant, module) [to ensure (teacher, module)] and Herite* (enseignant, personne) [Herite* (teacher, person)] ⇒ assurer (personne, module)).

Let Ref be an extension of the defined reference relationship previously. This relation is defined as follows:

Let 'ac' be a composite attribute.

Ref: CLASSExCLASSE $\rightarrow \{0,1\}$
 $(x, y) \rightarrow \text{Ref}(x, y)$
 $\{$
 Ref $(x, y) = 1$ if/* exists 'ac'/'x' is the class
 where 'ac' is defined or its sub or super-class and the
 class 'y' is the domain of 'ac' or a indirect reference link
 between the classes 'x' and 'y' */
 Ref $(x, y) = 0$ else;
 $\}$

3. Query Semantic Validation

A NLQ consists of one or two NGs articulating around a verb. We consider a NG constituted by one or several keywords. A keyword is an application dependent noun or a value. So, the process of the validation of the query consists in:

- *keyword's identification* in the NG; In the case of a NG formed by an identified single keyword, no supplementary treatment is necessary since only one component is used: the NG is valid. If the NG is constituted by more than a keyword and that one of them is not identified, the NG is considered as erroneous. Should the opposite occur, it is necessary to carry out the *NG's semantic validation*.
- If the query is constituted by a valid NG, no further-analysis is required. If the query covered two valid NGs with an identified verb, it is a subject of studying the *semantic coherence of the verb with its arguments*.

3.1. Keyword's Identification

The first step of the analysis is to determine if a word of the query is a keyword. For this, we define a simple semantic filter denoted $SF_{\langle w \rangle}$ as the set of triplets $\{(*, *, \omega)\}$ associated with the semantic value ω . The first star in the triplet represents the lexical information while the second designates the syntactic information associated with the word.

For an application dependent noun, a semantic filter is defined by:

$SF_{\langle \omega \rangle} = \{(*, \text{Noun}, \omega) / \omega \in \{C (\text{class}), SA (\text{simple attribute}), CA (\text{composite attribute})\}\}$ where the star represents the word's stem.

For instance, to represent the word 'students' that matches the class 'Student'; we would have the semantic filter $SF_{\langle C \rangle}$ including the triplet (student, Noun, C).

Likewise, the semantic filter, for a value, has the same structure except that the category ω takes values from the set:

$\omega = \{VK (\text{Value_as_Key}), VNK (\text{Value_None_Key})\}$ and the tagging information is the value in the first case, hence $SF_{\langle \omega \rangle} = \{(*, \text{value}, \omega)\}$. For The number '4' in the NG "coefficient 4", we have the triplet (4, value,

VNK).

From the query "what is the salary of the teacher Linda", $\text{salary} \in SF_{\langle SA \rangle}$, $\text{teacher} \in SF_{\langle C \rangle}$ and $\text{Linda} \in SF_{\langle VK \rangle}$ then all words are semantically correct.

The process of identification of a keyword ' m_i ' consists mainly in determining the corresponding simple semantic filter. It is defined by the following function:

Function Ident (m_i): boolean
 $\{$
 ident: = 1 if $\exists \omega \in \{C, SA, CA, VK, VNK\} /$
 $(m_i, *, *) \in SF_{\langle \omega \rangle} \cup SF_{\langle \omega \rangle}$
 ident:= 0 otherwise;
 $\}$

3.2. NG's Semantic Validation

In the case of a NG formed by several identified keywords, it is necessary to verify if the complex NG respects the conceptual constraints (semantic coherence constraints) of the domain database which amounts to associate semantic relations between every two consecutive keywords in the NG. These relations mainly refer to inheritance or reference relationships. They are based on the semantic values of the keywords and their combination in the NG.

For example, the NG 'salaire de l'enseignant' [salary of the teacher] is valid because the class 'enseignant' uses the attribute 'salaire' of its super class 'employé' [employee]. Thus, at the combination 'simple attribute-class' (SA-C), we must have an inheritance relationship between the class that contains the attribute (*i.e.* 'employé') and the specified class in the NG (*i.e.* 'enseignant').

Before giving the coherence constraints, we need to introduce some notations and notions:

- A part of a NG constituted by the succession of two keywords x and y is noted $x - y$.
- if $(m, *, *) \in FS_X(w) / w \in \{SA, C, CA, VK, VNK\}$ and $x \in \{N, V\}$, the keyword 'm' is respectively noted SA', C', CA', VK', VNK'.
- A class can be manipulated through its name, the name of the composite attribute which references it or a key value which is the value of its key attribute (e.g. 'module', 'matter' and 'Compilation' specify the same class 'module'). For that, 'class' will indicate these three information.
- In what follows we take:
 $X_i = \text{Classe}(m_i)$ if $m_i = C'$
 $X_i = \text{Class-ref}(m_i)$ if $m_i = CA'$
 $X_i = \text{Class}(\text{Attrib}(m_i))$ and $Y_i = \text{Attrib}(m_i)$
 if $m_i \in \{SA', VK', VNK'\}$

where *Classe* is a function which returns, for a class, the stem of this one and for a simple attribute that of the

class where it is defined. *Attrib* gets, for a simple attribute or for a value, the stem of this attribute and *Class-ref*, for a composite attribute, the stem of its referenced class.

3.2.1. Semantic Coherence Constraints

The combination (COMB) of two keywords according to their semantic value is governed by a set of rules called *coherence rules* (CR). Based on the patterns of combination, we have formulated coherence constraints (CC) that express the semantic relation between these words.

CR₁: *a class can be specified by either its super class or subclass. A class indicated by a key value cannot be specified by a class indicated by another key value. A key value can specify at most one class. The semantic relation is an inheritance relationship.*

COMB1: $m_i \in \{C', CA', VK'\}$ and $m_{i+1} \in \{C', CA', VK'\}$
 /if $m_i = VK'$ then $m_{i+1} \neq VK'$
 CC₁ ($m_i - m_{i+1}$): **Herite*** (X_i, X_{i+1})

CR₂: *a simple attribute can be specified by its value. The attribute corresponding to the value must be identical to the specified attribute in the query and their class respective too.*

COMB2: $m_i \in \{SA'\}$ and $m_{i+1} \in \{VK', VNK'\}$ or $m_i \in \{VK', VNK'\}$ and $m_{i+1} \in \{SA'\}$,
 CC₂ ($m_i - m_{i+1}$): $Y_i = Y_{i+1}$; $X_i = X_{i+1}$

CR₃: *a simple attribute of a class can be used by its class, its super-classes or sub-classes. The semantic relation is an inheritance relationship.*

COMB3: $m_i \in \{C', CA', VK'\}$ and $m_{i+1} \in \{SA'\}$ or $m_i \in \{SA'\}$ and $m_{i+1} \in \{C', CA', VK'\}$,
 CC₃ ($m_i - m_{i+1}$): **Herite*** (X_i, X_{i+1})

CR₄: *a class is associated with another class if there exists a reference link between these classes. The semantic relation is a reference relationship.*

COMB4: $m_i \in \{C', CA', VK'\}$ and $m_{i+1} \in \{C', CA', VK'\}$,
 CC₄ ($m_i - m_{i+1}$): **Ref** (X_i, X_{i+1})

3.2.2. Algorithm of NG Validation

The operation of validation of a NG constituted by several identified keywords $m_1 - m_2 - \dots - m_n$ consists in finding a pattern of the combination for every two consecutive words m_i and m_{i+1} for $i \in [1, n-1]$ and verifying the correspondent semantic relation $CC_j(m_i - m_{i+1})$.

The algorithm of validation is defined as follows:

```
{
i = 1;
if i = n then
/* NG  $\equiv$  1 keyword */
```

```
if Ident ( $m_i$ ) then 'valid NG' else 'Erroneous NG'
else
/* NG  $\equiv$  several keywords */
while ( $i < n \wedge$  Ident ( $m_i$ )  $\wedge$  Ident ( $m_{i+1}$ )  $\wedge \exists j \in [1,4]/$ 
CCj ( $m_i - m_{i+1}$ )) do
i = i + 1; end;
if i = n then 'valid NG' else 'Erroneous NG'
}
```

3.3. Semantic Coherence of the Verb with Its Arguments

In the lexicon, a verb is described with its real subject (RS) and its real object (RO). In a query, the subject of a verb is one of keywords of the subject NG which really carries out the action. It is noted (QS). The object is generally the first keyword of object NG which undergoes the action. It is noted (QO).

A verb is compatible with its subject NG (QS) and its complement (QO) if:

- QS is identical to RS or it is its super or sub class because the subject is always a class;
- 1) The object is a class, then QO must be identical to QS or be its super or sub class;
 2) The object is an attribute (or a value of an attribute) then the attribute QO must be identical to QS.

4. Example of Query's Validation

To understand the following example and the examples of illustration quoted in the text, we give an extract of a NLDB on the University of Oran, in French and its equivalent in English. The adopted syntax is the following one:

```
<class> ({<mother class>}, <attribute1>, <attribute2>, ...
<attributen>, verb1, verb2, ...)
Personne ({}, nom, ville, habiter)
Employé ({Personne}, salaire)
Enseignant ({Employé}, matière: Module)
Chercheur ({Enseignant}, projet: Projet)
Module ({}, nom, coefficient)
Projet ({}, laboratoire: laboratoire)
Laboratoire ({} ...)
```

```
Person ({}, name, town, to live)
Employee ({Person}, salary)
Teacher ({Employee}, matter: Module)
Searcher ({Teacher}, project: Project)
Module ({}, name, coefficient)
Project ({}, laboratory: laboratory)
Laboratory({} ...)
```

Example: 'Donner le salaire des enseignants de Compilation qui habitent Oran'

[Give the salary of teachers of Compilation who live in

Oran]

Verb = habiter;

NG1 = $m_1 - m_2 - m_3$ = noms – enseignants – Compilation;

$m_1 \in SF_N\langle SA \rangle$, $m_2 \in SF_N\langle C \rangle$, $m_3 \in SF_V\langle VK \rangle$,

$m_1 - m_2 = COMB_3$;

CC₃ = Herite* (Classe (Attrib (nom)), Classe (enseignants))

= Hérite* (employée, enseignant): = true;

$m_2 - m_3 = COMB_4 = Ref$ (Classe (enseignants), Classe (Attrib (Compilation))) = Ref (enseignant, module): = true;

NG2 = Oran; $m \in SF_V\langle VNK \rangle$,

In the lexicon, we have: habiter (personne, ville) and in the query habiter (enseignant, Oran);

Herite* (personne, enseignant): = true, Attrib (Oran) = ville.

The keywords are identified, the CC_j between them validated and the verb is coherent with its arguments, and then *the query is valid*.

5. Experimental Results

To evaluate the performance of the system, we have tested our system on 120 significant queries collected from the students of computer department of university Oran. 7 queries were disregarded because they did not cover the BD application domain. Of the remaining queries, 102 queries were correctly analyzed (*i.e.* success rate 90, 26%). The errors were mainly due to unregistered keywords in the lexicon. Our lexicon contains nouns, verbs and their various synonyms that typically are used in a particular domain. Their semi-automatic generation involves inevitably false results. A better solution for the interface is the integration of a learning system to enrich the lexicon.

6. Conclusions

Our work proposes a method of query's validation for OODB. Define a query key information to be analyzed semantically: its structure from one part and its sense from another part, express the link which connects them and the method of handling them were the strong problems which we tried to solve.

There is not dependency between our system and the DB application domain. When changing one DB to another, the only change occurs at the lexicon.

Nevertheless, the study present limits relating to selected requests model (a request includes only one subject, only one complement and only one verb). These limits carried out knowingly, into a qualitative aim, make that, the developed tools remain incomplete but can be wide to more complex queries.

Also note that, the presented information is necessary in NLDB, to treat the second aspect of the semantics which is the translation of the NLQ to formal language

query such as OQL [16].

REFERENCES

- [1] A.Copestake and K. S. Jones, "Natural Language Interfaces to Databases," *The Knowledge Engineering Review*, Vol. 5, No. 4, 1989, pp. 225-249.
- [2] I. Androutsopoulos, G. Ritchie and P. Thanisch, "Natural Language Interfaces to Databases – An Introduction," *Journal of Language Engineering*, Vol. 1, No. 1, 1995, pp. 29-81.
- [3] J. Chae and S. Lee, "Natural Language Query Processing in Korean Interface for Object-Oriented Databases," *Proceedings of First International Workshop on Applications of Natural Language to Databases (NLDB'95)*, Versailles, 1995, pp.81-94.
- [4] P. Reis, J. Matias and N. Mamede, "Edite – A Natural Language Interface to Databases: A New Dimension for an Old Approach," *Proceedings of the Fourth International Conference on Information and Communication Technology in Tourism (ENTER'97)*, Edinburgh, 1997.
- [5] Y. Chandra and R. Mihalcea, "Natural Language Interfaces to Databases," Thesis (M. S.), University of North Texas, Denton, 2006.
- [6] M. Owda, Z. Bandar and K. Crockett, "Conversation-Based Natural Language Interface to Relational Databases," *International Conference on Web Intelligence/IAT Workshops*, 2007, pp. 363-367.
- [7] M. Minock, P. Olofsson and A. N. Aslund, "Towards Building Robust Natural Language Interfaces to Databases," *Proceedings of the 13th International Conference on Natural Language and Information Systems*, Berlin, 2008, pp 187-198.
- [8] J. Robie and D. Bartels, "A Comparison between Relational and Object Oriented Databases for Object Oriented Application Development," POET Software Corporation 800-950-8845, 1994.
- [9] J. Clifford and D. S. Warren, "Formal Semantics for Time in Databases," *ACM Transactions on Database Systems*, Vol. 8, No. 2, June 1983, pp. 215-254.
- [10] R. A. Frost and P. Boulos, "An Efficient Compositional Semantics for Natural Language Database Queries with Arbitrarily-Nested Quantification and Negation," *Lecture Notes in Computer Science*, Vol. 2338, 2002, pp. 252-267.
- [11] H. D. Lee and J. C. Park, "Interpretation of Natural Language Queries for Relational Database Access with Combinatory Categorical Grammar," *International Journal of Computer Processing of Oriental Languages*, Vol. 15, No. 3, 2002, pp. 281-304.
- [12] R. A. Frost and R. J. Fortier, "An Efficient Denotational Semantics for Natural Language Database Queries," *Natural Language Processing and Information Systems*, Vol. 4592, 2007, pp. 12-24.
- [13] A. M. Popescu, O. Etzioni and H. Kautz, "Towards a Theory of Natural Language Interfaces to Databases," *Proceedings of the 8th International Conference on Intelligent User Interfaces*, Miami, 2003, pp. 149-157.

- [14] J. Little, M. Ga, T. Ozyer and R. Alhajj, "Query Builder: A Natural Language Interface for Structured Databases," *Lecture Notes in Computer Science*, Vol. 3280, 2004, pp. 470-479.
- [15] V. Boonjing and C. Hsu, "A New Feasible Natural Language Query Method," *International Journal on AI Tools*, Vol. 15, No. 2, 2006, pp. 323-330.
- [16] G. G. R. Cattell, "ODMG-93 Standard des Bases de Données Objet," Addison-Wesley, Boston, 1995.