OBTAINING E-R DIAGRAMS SEMI-AUTOMATICALLY FROM NATURAL LANGUAGE SPECIFICATIONS

Farid Meziane and Sunil Vadera
School of Computing, Science and Engineering
Salford University, Salford M5 4WT, UK.
{f.meziane, s.vadera}@salford.ac.uk

Keywords: Software engineering, Entity relationship models, Specifications, Natural language processing.

Abstract: Since their inception, entity relationship models have played a central role in systems specification, analysis and development. They have become an important part of several development methodologies and standards such as SSADM. Obtaining entity relationship models, can however, be a lengthy and time consuming task for all but the very smallest of specifications. This paper describes a semi-automatic approach for obtaining entity relationship models from natural language specifications. The approach begins by using natural language analysis techniques to translate sentences to a meaning representation language called logical form language. The logical forms of the sentences are used as a basis for identifying the entities and relationships. Heuristics are then used to suggest suitable degrees for the identified relationships. This paper describes and illustrates the main phases of the approach and presents a summary of the results obtained when it is applied to a case study.

1 INTRODUCTION

Since their inception, entity relationship models (ERMs) have played a central role in systems specification, analysis and development. They have become an important part of several development methodologies and standards such as SSADM (Ashworth and Goodland, 1990). Obtaining ERMs, can however, be a lengthy and time consuming task for all but the very smallest of specifications. This paper describes a semi-automatic approach for obtaining ERMs from natural language (NL) specifications.

The overall approach, summarised in Figure 1, is based on the view that nouns often denote entities and verbs often denote relationships. However, as we will see later, picking out just the nouns and verbs using string matching is not adequate for producing an ERM. We need also to identify the arguments and the degrees of the relationships. To enable this, the approach begins by using NL analysis techniques to translate sentences to a meaning representation language called *logical form language* (LFL). The logical forms (LFs) of the sentences are then used as a basis for identifying the entities, and relationships. The quantifiers in the LFs are then used to suggest suitable degrees for the identified relationships. The paper is organised in a manner that follows the main phases of the approach: section 2 describes the translation to LFL; section 3 illustrates the identification of the entities and relationships; and section 4 shows how the degrees can be identified. The paper concludes with a summary of the results obtained when the approach is applied to a case study.

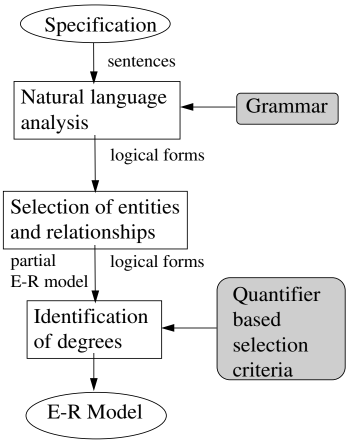
Specification
sentences
Natural language analysis
Grammar
logical forms
Selection of entities and relationships
partial E-R model
logical forms
Identification of degrees
Quantifier based selection criteria
E-R Model

Figure 1: Identifying E-R models semi-automatically

2 TRANSLATING SENTENCES TO LFL

As mentioned above, each sentence of a specification is first analysed and translated to a statement in LFL. In this section we summarise the syntax of LFL and the translation process. We refer the reader to (McCord, 1990; Meziane, 1994) for further details.

In general, each sentence can be translated to a LF statement of the form:

$$determiner(Base, Focus)$$

Typically, the base comes from the remainder of the noun phrase (NP) in which the determiner appears, and the focus comes from the sisters of the NP such as verb phrases (VP), and other NP.
In LF, nouns, verbs, adjective, and pronouns are represented as follows, where we use the Prolog convention that variables begin with a capital letter:

Nouns Nouns are usually represented as one place predicates (aircraft, $aircraft(X)$). Relational nouns take two arguments (mother, $mother(X,Y)$).

Verbs Depending on their category, verbs may be represented by predicates having nil, one, two or three arguments. Hence, the verbs *snow, crash, write* and *give* are represented by: $snow$, $crash(X)$, $write(X,Y)$ and $give(X,Y,Z)$.

Adjectives We distinguish two categories of adjectives, extensional adjectives and intensional adjectives. An adjective is intensional if it cannot be dissociated from the noun it modifies. An extensional adjective can be dissociated from the noun it modifies. The following two examples illustrate these two situations respectively:[1]

- The pilot uses a *moving* map display.
  the(pilot(X),the(moving(map(display(Y))), use(X,Y))).
- A *complex* aircraft uses a radar.
  ex(aircraft(X) & complex(X), ex(radar(Y), use(X,Y))).

Pronouns There is no general rule on how to interpret pronouns. Basically they are supposed to be replaced by the nouns they represent. However, resolving pronoun references is a very difficult problem. Our current implementation omits this aspect of NL understanding.

The translation process can take English sentences and produce LFs in the above representation. It does this in two main phases. First, a syntax analysis is performed to produce all possible parsings (syntax trees) of the sentence according to the defined grammar. Each syntax tree is then transformed into a unique LFL expression. This latter transformation forms the semantic interpretation of the English sentence and involves the use of logical operators to combine the different parts of the syntax tree to produce the desired LF (see (McCord, 1990) for details). This process has been implemented in Prolog and a suitable grammar has been developed. For example, the grammar rule used for a VP is:

```
vp(Infl,E,X) ==>
vhead(Infl,E,X,Slots):
postmods(Slots).
```

This rule defines a VP to be composed of a verb head and a list of postverbal modifiers. The job of `vhead` is to find a verb with an inflection `Infl`, subject marker `X`, verb type `E` and a list `Slots` of postverbal modifiers which contains verb modifiers such as objects, indirect objects and prepositional phrases. The grammar also includes prepositions [2] as well as the other categories mentioned above.

Sentences may be ambiguous, and may therefore have several meanings. In such cases, the NL analyser results in alternative LFs and an analyst will be required to select the intended meaning. Although, this makes the process less automatic, this is helpful since it enables ambiguities to be detected.

3 E-R MODELS FROM LOGICAL FORMS

The first task in identifying an ERM is to obtain a list of entities in the specification and the relationships between them. There is no clear definition of what constitutes an entity. In SSADM for example (Ashworth and Goodland, 1990), an entity is defined as something of importance to the system about which information can be held. The same definition is also used by Bowers (Bowers, 1988), who further suggests that entities can be objects (person, car, events (birth, scoring a goal), activities (production, playing) and associations (marriage).

Grammatically speaking, the above list gives types which define entities that are related. They all belong to the same grammar category of *nouns*. Likewise, a number of authors have reported that relations are mainly described by verbs (Ashworth and Goodland, 1990; Gane and Sarson, 1979). We therefore base our identification process on the view that entities are denoted by nouns and relationships by verbs. However, just scanning for nouns and verbs alone is not adequate. There are three significant problems that we now illustrate with the following sentences:

1. A company maintains a description for each item of stock.

[1] ex denotes the usual existential quantifier.

[2] Handling prepositions is non-trivial and details are given in (Meziane, 1994)

2. A computer-assisted flight planning system is used by a complex aircraft.
3. The system of a simple aircraft is considered to comprise the plan of the pilot.

The entities and relations can easily be picked out in the first sentence as the nouns: company, description, item, stock and verb maintains. However, we still need to find what entities are related by maintains and we don't know the degree of the relations. In the second sentence, selection of nouns alone as entities is inadequate since we need to identify the compound noun computer-assisted flight planning system. The third sentence has two verbs, comprise and consider. How can we identify that comprise is the one that relates the entities and that consider is only a subsidiary relation?

Fortunately, by first translating sentences to LFL, we are able to overcome these problems. So, for instance, the above three sentences result in the following LFs (where *all* and *ex* are the usual universal and existential quantifiers):

1. *all*(**item(X,stock)**,*the(company(Y),ex(description(Z), for(X,maintain(Y,Z)))))*
2. *ex(aircraft(X) & complex(X), ex(computer-assisted(flight(planning(Y))),use(X,Y)))*
3. *ex(aircraft(X)& simple(X),***the(system(Y,X),** *the(pilot(Z),* **the(plan(T,Z)**, *(comprise(Y,T)))))*

Given these LFs, the required information can be relatively easy to extract:

1. The term *maintain(Y,Z)*, in the first LF, gives the relationship between *Y* and *Z* which themselves are qualified in the focus as the company and the description;
2. the compound noun computer-assisted flight planning system is easily identified from the term *computer-assisted(flight(planning(system(Y))))*.
3. In the third LF, the relation 'comprise' is correctly identified by extracting the inner verb relationship *comprise(Y,T)* from the LF.

At this stage, its worth emphasising that this process produces only an initial list of entities and relationships. The model may well be incomplete since the informal description may be incomplete and may contain irrelevant entities and relationships.

4 USING QUANTIFIERS TO DETERMINE THE DEGREES

This section shows how the degrees of relationships can sometimes be identified from the quantifiers in the LFs of sentences. This approach to identifying the degrees is therefore highly dependent on the process of identifying the quantifiers in the English sentences. Hence, section 4.1 looks at the process of identifying the quantifiers in some detail. Section 4.2 then describes how the quantifiers help to identify the degrees.

4.1 Identifying Implicit Quantifiers

The English language has two articles: the definite article "the" and the indefinite article "a" ("an") It has been assumed for a long time that both articles can be interpreted as existential quantifiers. Some authors have shown that this is not always true, and these articles can sometimes lead to the universal quantifier(Hess, 1985). In the following subsections we will identify some cases where the quantifiers can be identified from the articles.

The Definite Article "the"

The definite article is often translated into the unique existensial quantifier (ie., there exists one and only one). It is, for instance, correct to assume that in the sentence:

The student passed the exam.

we are talking about a particular student who passed a particular exam. However, in the sentence:

The students passed the exam.

we cannot assume the unique existence for the first definite article. McCord (McCord, 1990) also recognises that interpreting the definite article only as the unique existence is not adequate but does not suggest any alternatives. In our approach we use the singularity or plurality of the noun to determine if it should be interpreted as the unique existence or normal existence. It is interpreted as the unique existence only if the noun quantified is in the singular form.

The Indefinite Article "a"

There seems to be general agreement that the use of the indefinite article is always a source of ambiguities (Allen, 1987). The indefinite article can sometimes be translated to the existential quantifier and sometimes to the universal quantifier. According to Hess (Hess, 1985) the most important way to determine the quantification of a sentence is through the choice of the verb. For example, consider the following sentences:

1. A text editor *makes*modifications to a text file.
2. A text editor *is making* modifications to a text file.
3. A text editor *made* modifications to a text file.
4. A text editor *has made* modifications to a text file.

The present tense is used in example (1) to say that a text editor makes modifications to a text file in general. The main use of the present tense is to express habitual actions. In examples (2) to (4) we say that there is, or was, a case of a text editor making modifications to a text file. Therefore, Hess suggested that because the present tense is used in the first sentence, text editor must be universally quantified. Likewise, because of the tenses used in the other sentences, text editor must be existentially quantified in the remaining sentences.

In some cases the future is preferred over the present tense for general statements as in the following example:

A man who loves a woman will stroke her.

Dynamic verbs, such as to stroke, seem to call for the future tense, whereas static verbs such as to respect seem to go better with the present tense. Hence, Hess formulated the following rules:

- **Rule 1:**
  The subject of a sentence is existentially quantified if the VP is in the past tense, in the progressive aspect, or in the perfective aspect.
- **Rule 2:**
  Otherwise the subject is universally quantified, in particular if it is in the present tense or in the future tense.

Once we have determined the quantification of the subject of the sentence, we have to do the same thing to the other components of the sentence. Let us consider the following examples:

1. A man who loves a woman is happy.
2. A man that loves a woman respects her.

Intuitively, we can see that woman should be existentially quantified in the first sentence and universally quantified in the second sentence. To observe the difference, Let us consider the LFs of these sentences:

1. *all(man(X),ex(woman(Y)&love(X,Y),happy(X)))*
2. *all(man(X),all(woman(Y)&love(X,Y),respect(X,Y)))*

The main verb of the first sentence is happy and does not refer to the NP woman. In the second sentence the main verb respects refers to the NP woman. This is the reason why the NP "woman" should be existentially quantified in the first sentence and universally quantified in the second. Hence, Hess suggested a third rule which is:

- **Rule 3:**
  In a restrictive NP those arguments that are referred to by the main verb are universally quantified and those that are not referred to by the main verb are existentially quantified.

This rule now enables the correct interpretation of the above sentences. However, it does not hold for non-restrictive NPs. In particular, when a NP appears at the right of a verb, the kind of sentences we have encountered suggest that the indefinite article should be interpreted as an existential quantifier. For example in the sentence:

A complex aircraft uses a radar.

The second indefinite article is interpreted as the existential quantifier and not as the universal quantifier. There are two exceptions to the above rules which are analysed in the following cases:

- As an exception to rule 2, the past tense can express a universally quantified assertion, as in the following example:

  A student read books *when* I was young.

  This universal quantification is possible because the main verb (read) requires a spatial or temporal post modifier(when).
- As an exception to rule 1, the progressive aspect can express universal quantification as in:

  John is always coming late

  This is only possible when the verb is modified by expressions such as "always", "in general", "regularly".

To cover these exceptions, we can suggest the following fourth rule which takes precedence over rules 1 and 2.

- **Rule 4:**
  1. The past tense can express a universally quantified assertion if the main verb requires a spatial or a temporal post modifier.
  2. The progressive aspect can express a universal quantification if the verb is modified by expressions such as "always", "in general" and "regularly".

4.2 Identifying the Degrees from the Quantifiers

This section illustrates how we can make use of the quantifiers identified in the previous section to identify the degrees of some relations. Consider the following examples and their LFs:

- A complex aircraft uses a radar.
  all(aircraft(X) & complex(X), ex(radar(Y), use(X,Y)))

- The students passed the exam.
  all(student(X), the(exam(Y), pass(X,Y)))

In the first example, the first entity in the relation is quantified by the universal quantifier and the second by the existential quantifier. Based on our current experience, and the examples encountered, usually only one occurrence of the variable quantified by the existential quantifier is involved in the relation. In such cases the LF quantifier "ex" is interpreted as a unique existential quantifier. We are therefore in a case were many occurrences of the first variable are related to one occurrence of the second variable. By definition, this is a many-to-one relationship. In the second example, the interpretation is much more stronger since we have a unique existence interpretation for the second "the". We have again a case of a many-to-one relationship. Let us now consider the following set of sentences and their LFs:

- The company maintains a description for each item of stock.
  all(item(X,stock), the(company(Y), ex(description(Z), for(X,maintain(Y,Z)))))
- The student passed all exams.
  the(student(X), all(exam(Y), pass(X,Y)))

The NP each item of stock, in the first sentence, suggests that we are talking about a particular stock that contains many items. Therefore wa have a one-to-many relation between the entity "stock" and the entity "item". In general, sentences where the first entity is singular and quantified by the definite article and when the second entity is quantified by the universal quantifier define one-to-many relationships. A typical example is the second sentence. Let us consider now the following sentence and its LF:

- The student passed the exams.
  the(student(X), the(exam(Y), pass(X,Y)))

The previous rule does not apply because the second entity is itself singular and quantified by the unique existential quantifier. As this example suggest, we are talking about a particular student who passed a particular exam. In this case, we infer a one-to-one relationship between the entities.

These are the main cases where our approach can help in identifying the degrees of the relations from the LFs of the sentences. In other cases, when it is difficult to predict the degree of a relation, we let the user determine it.

5 CONCLUSION AND FUTURE WORK

This paper has presented a novel approach that can help an analyst produce an initial ERM from specifications written in NL. The approach makes use of NL analysis techniques to translate sentence to LFs which are then used as a basis for identifying the entities and relationships. The quantifiers in the LFs also enable the identification of the degrees of relationships in some common cases.

The approach has been implemented in Prolog and tested on some examples. To date, the most interesting application has been to a specification of a flight planning system that was written independently of our work (Hepworth, 1988; Vadera and Meziane, 1994). In that case study, the approach worked well in that:

- The majority of entities and relationships were correctly identified. The system identified 55 entities of which only 1 was thought to be spurious and none had been missed. It identified 52 relationships of which were incorrect and none overlooked.
- Most of the degrees were correctly identified. The degrees for 49 of the 52 identified relations were correctly predicted.

Future research aims to develop the techniques so that a wider range of sentences and more structured objects, like tables, can be handled by the NL processing phase. This should enable a broader evaluation of the approach on larger specifications. The results obtained with our current implementation are encouraging and suggest that further research may lead to an invaluable practical aid for producing ERMs.

REFERENCES

Allen, J. (1987). *Natural language understanding*. The Benjamin/Cummings Publishing Company, Inc.

Ashworth, C. and Goodland, M. (1990). *SSADM: A practical approach*. McGraw-Hill Book company.

Bowers, D. (1988). *From data to data base*. Van Nostrand reinhold (U.K) Co. Ltd.

Gane, C. and Sarson, T. (1979). *Structured System Analysis*. Prentice-hall Software series.

Hepworth, B. (1988). An introduction to Z. Technical Report BAe-WIT-RP-GEN-SWE-152, Systems Computing Department, British Aerospace Ltd.

Hess, M. (1985). How does natural language quantify? In *Second Conference of the European Chapter of the association for Computational Linguistics*, pages 8–15.

McCord, M. (1990). Natural language processing in Prolog. In Adrian, W., editor, *A logical approach to expert systems and natural language processing Knowledge systems and PROLOG*, pages 391–402. Addison-Wesley Publishing company.

Meziane, F. (1994). *From English to Formal Specifications*. PhD thesis, University of Salford.

Vadera, S. and Meziane, F. (1994). From English To Formal Specifications. *The Computer Journal*, 37(9):753–763.