FORMAL SPECIFICATIONS BUILDING FROM SPECIFICATIONS WRITTEN IN NATURAL LANGUAGE

Alain-Jérôme Fougères

FORMAL SPECIFICATIONS BUILDING FROM SPECIFICATIONS WRITTEN IN NATURAL LANGUAGE

Alain-Jérôme Fougères

LaRIS/UTBM
4, rue du Château Sévenans
90010 Belfort – France
Alain-Jerome.Fougeres@utbm.fr

Abstract :
Making specifications is taking more and more time; every day an enormous quantity of pages which, for the most part, is written in natural language. However the need to reduce the time needed in the development of the services is a priority. One method is to formalise the maximum number of specifications received. With this in mind, we will try to demonstrate the possibility of a certain automation in the passage from the informal to the formal, by means of methods and proven tools, available to assist an expert in specifications. For this end we propose a process of formalisation which relies on an intermediary representation of the specifications with the formalism of conceptual graphs before arriving at a formal description in Z of the initial specification.

Keywords: knowledge representation, natural language, formal specifications.

1 Presentation

In the first phases of complex system development, such as for telecommunications, software specifications, services or hardware, are accompanied by long documents usually written in natural language. For the persons in charge of specifications, there are two dimensions to solving the problem: when producing specifications, are their procedures which could facilitate the development and writing of specifications? when managing specifications, what sort of help could be found to help in the archiving of the masses of data and in maintaining coherence during their development and exploitation?

1.1 The context of study

This research is motivated by desire to find methods of reducing the time needed to develop new services in computer science or telecommunications. It serves as a means to shorten all the stages in the cycle of development of each service offered. In this context, mastering the stage of functional specifications becomes of prime importance, since it facilitates the realisation of the service. Thus, each step which facilitates the writing of the specifications, whilst keeping the expected quality [Sommerville, 1992], contributes to a reduction of the time, inherent in this stage and, consequently in subsequent stages. Our objective consists of helping specification writers to formalise their specifications, by concentrating our attentions on the semantic aspects [Toussaint, 1992] of an informal specification. The *point de départ* for this procedure of formalisation arrives from specifications written in natural language, being converted in the terminology of software science with regard to the target, that must be determined within the context of the not inconsiderable set of formal languages. There is a general

concensus which states that you cannot reasonably envisage passing directly from natural language to formal representation; in the main this being due to the problems inherent in the use of natural language and, more particularly, in its interpretation (ambiguities, context). This has led us to select for the construction of an intermediary semantic representation defined in [Sowa, 1984]: the model of *conceptual graphs* (CG).

1.2 The procedure of formalisation

For this move towards formalisation, we propose a sequence of processes from informal specifications, likely to provide us with a formal description (fig.1).

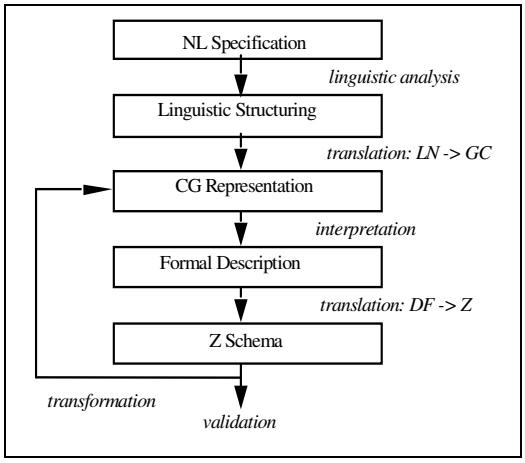

Figure 1: Different stages in the processing of specifications

1.3 Experimentation

Our experiments was based on the specification of *NEF*[1] and more specially on the tenth chapter detailing the *tarification*. The extensive nature of this specification and the linguistic complexities attached to it, have not permitted us, in this first approach to foresee a complete definitive path of formalisation. We settled, as an experimental protocol, on the realisation of the complete procedure in an incremental way.

2 Description of the work

2.1 Linguistic aspects

The processing of the language breaks down into two stages: a preliminary stage in which there is the acquisition of knowledge pertaining to the domain, and then the actual stage of linguistic analysis, itself. This second stage is generally sub-divided into five stages of analysis morphological, lexical, syntactic, semantic, pragmatic.

2.1.1 The acquisition of knowledge phase

This preliminary stage consists of extracting lexical information contained within the text, in order to determine the preferred links that the words have between them. A

[1] NEF : *Normes d'exploitation et de fonctionnement* (rules of conduct and operation) of France Telecom, worked out at CNET (Centre National d'Etudes des Télécommunications).

simple study of the co-occurrence of words, based on an analysis of lexical proximity, thus enables us to reveal the presence of compound words, of expression, of predicative relationship and of schemas of phrases peculiar to the domain. The united use of this frequential analysis with techniques of statistical filtrage, such as the *mutual information*[2] permits refinement and improvement in the pertinence of the results.

A second phase of knowledge acquisition consists of extracting from the dictionary some definitions of terms retained as concepts, in order to describe them in a semantic dictionary in the form of conceptual graphs. In order to automate this task, we have adapted the algorithms of [Hernert, 1993] which allow us to detect the hyperonymic relationship contained in the dictionary definitions and to adjust them with terms modifying the definition. Once the content of the definition has been analysed, it is then possible to construct the corresponding CG and to include it in a canonical base.

2.1.2 The linguistic analysis phase

The morpho-lexical analysis. In the course of this phase, it is question of sequencing the analysed sentences in order to obtain a series of words after having identified the simple words, the compound words and the set expressions.

The syntactic analysis. Our aim here is not to set out the full array of numerous strategies for syntactic analysis but rather to set out clearly the formalism *Lexical Functional Grammars* (LFG) [Kaplan and Bresnan, 1982] that we have chosen. The LFG break down into three levels:

- the *c-structure* (analysis by components) described with rules of grammar out of context; it represents the syntactical structures in the form of a tree;
- the *f-structure* (functional description) comprises pairs of function-value, shows the grammatical functions such as subject, object, etc.;
- the *s-structure* (semantic structure), semantic projection based on the c-structure which only allows for predicate structure (predicates, arguments and modifiers).

LFG grammars add to the construction of the syntactic structure the formation of functional phrase structure (logic analysis), specified using patterns which are associated with grammar rules. But, the LFG analyser used does not have a sufficient linguistic range to analyse the more complex phrases frequently found in specification writing.

The semantic analysis. The selected formalism of representation of semantic knowledge, being the model of conceptual graphs, this analysis therefore consists of the semantic translation of the syntactical structure into the form of conceptual graph. For this, we have taken inspiration from the case grammars which determine the different thematic roles taken by the components of a phrase with the help of information acquired about word-order, about prepositions, verbs and context. The analyser determines the way in which the nominal groups of a phrase are bound to the verbs: the semantic role specifying how an object participates in the description of an action.

---

[2] When the number of couple of lexical unities observed becomes elevated, we estimates the probabilities of pertinent association by a method of likelihood: $I(x,y) = log\ (n_{x,y\ /\ nxny})$, with nx and ny the number of occurrences of x and y, and nx,y the number of occurrences of the couple (x,y).

2.2 The formalism of conceptual graphs

2.2.1 The elements of formalism

The world described by a formalism of representation of semantic knowledge is a collection of individuals and of relations between these individuals which specify a state which could be transformed. In the model of conceptual graphs [Sowa, 1984], the elementary objects are the *concepts* and *the relations*. Each proposition is represented by a CG built using oriented arcs connecting concepts and relations. Rules offering the possibility to join or disassociate are given. A formal correspondence with first order predicate logic is taken as a core. This basic description of CG shows the mixed qualities of this formalism: a graphic representation making it easy to read, and a system of mathematical proofs with solid axiomatic foundation which makes it formal.

2.2.2 Isomorphism CG and logic of the first order

Sowa defined the operator ϕ which makes a formula in the predicates logic of the first order correspond to every conceptual graph. In the following example, the sentence *The system consists of transmitter* will have for equivalent the formula $\phi(u)$:

u : [**system**: num1]->(**ConsistsOf**)->[**transmitter** : *].

$\phi(u) : \exists x,$ **system**(num1) $\wedge$ **ConsistsOf**(num1,x) $\wedge$ **transmitter**(x)

2.3 Towards a formalisation

2.3.1 Z language as the target language

A specification in Z [Spivey, 1992] is formed by a sequence of *paragraphs* comprising *schemas*, *variables* and *base types*. To every expression appearing in a specification in Z is associated a unique type. This type can be one of three sorts, a whole type, a cartesian produced type or still a schema type. The relations or the functions allow us to combine these three sorts of objects. A schema consists of a signature and of a property on this signature called *predicative part*. A signature is a collection of variables, each one possessing a type. They are created by the declarations and they provide the vocabulary necessary to the mathematical instructions expressed by the predicates. A predicate is the expression of a property which is characterised by the whole of the links for which it is true. The variables are of two sorts: the local variables which have a reduced scope on their schema of declaration and the global variable which form the object of a declaration outside the schema. Moreover, the formalism contains three standard *decorations* used in the description of the operations: "'" to label the final state of an operation, "?" to label its entries and "!" to label its exits.

2.3.2 Building of a formal description

To build a formal description which corresponds to an informal specification, we worked by analogy with the construction mechanism developed by a human expert (fig.2). We begin by extracting the elements of the formal description, next we identify and insert the indispensable elements which are not included in the module to be specified, then we establish the logic formulas (pre-conditions, post-conditions) corresponding to the collected elements and their links defined in natural language. The final phase is built by modelling in Z the CG based on the informal specification.

init : 1 CG by analyzed phrase.
Step 1 : join the graphs of one section : Joint(u1, ..., un) -> u.
Step 2 : find the external references.
Step 3 : find the parameters, variables and relation of the description :
. list the individual referent (object instances),
. list the concepts in function of the referent type,
. list the relations in function of the arity.
Step 4 : launch CG -> Z algorithm which gives Schema U.

Figure 2: Formal description construction methodology based on CG

The algorithmic elements of the translation of a conceptual graph in Z having been presented in [Fougères, 1997], we will essentially retain the two successive processes done on the concepts and on the relations. The referent of a concept becomes an element of the whole, represented by the type label; as for the relation, this is the object of a functional definition. The following figures illustrate the process of formalisation by presenting the three levels of representation of the specification of a simple transmission of messages between a transmitter and a receiver via a channel of transmission. We first applied the methodology on the phrases (1, 4 and 7) in figure 3 to obtain a unique graph *u* (fig.4, left). Then we calculated the corresponding logic formula, thus deducing the formal description and then we derived schema Z corresponding to *u* (fig.4, right).

(1) System consists of a transmitter.
(2) System consists of a channel.
(3) System is consists of a receiver.
(4) Transmitter sends a message.
(5) Transmitter receives indication of loss p in channel.
(6) Channel receives message.
(7) Channel transmits message to receiver.
(8) Channel returns indication of loss to transmitter.
(9) Receiver receives message.

Figure 3: Simplified wording for "message transmission"

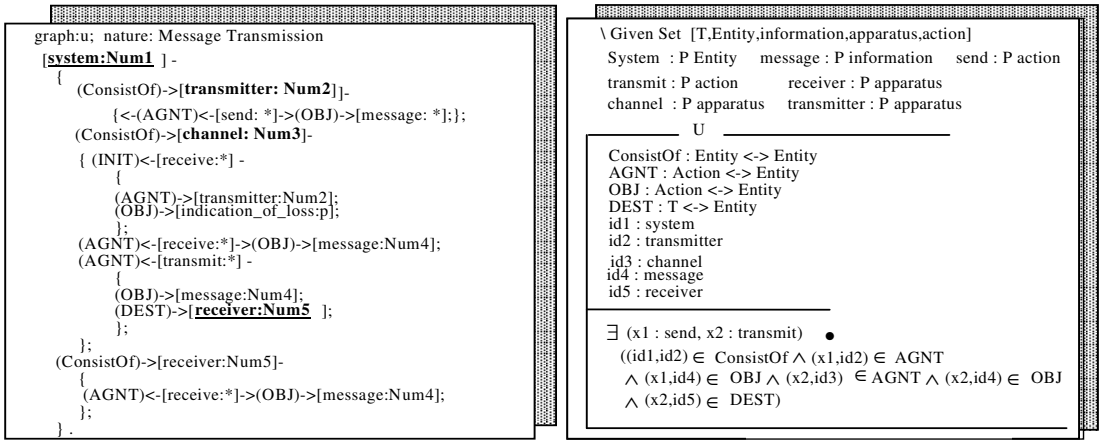

Figure 4: Fragments of the representation in the form of CG and specification in Z

3 Conclusion

We have proposed in this article an overall presentation of the procedure of formalisation of informal specifications expressed in natural language. It would be tempting, after having described a formalisation process, to postulate on perspectives of making an "automatic specifier" [Balzer, 1985]. This however is not our point of view. We have not distanced ourselves from perspectives of assisted specification, even if we have only briefly mentioned the problematic of assistance. Automation of translating of NL specification to formal specification contains three problems:

- managing the intrinsic semantic irregularities of natural language;
- finding the quantity of expert information, of which abstraction is made in the source specifications – characterisation of the explicit and the implicit;
- compensating for the irreducible part of the passage from informal to formal – Newell clearly underlines the existence of a specific level of knowledge located outside of any formal system as well as the problem of transferring this informal knowledge into formal knowledge ("symbol level").

References

R. Balzer (1985). A 15 Year Perspective on Automatic Programming, in *IEEE Transactions on Software Engineering*, vol. SE-11, No 11, pages 1257-1268.

E. Davis (1990). Representations of Commonsense Knowledge, Morgan Kaufmann Publishers, San Mateo CA.

A.-J. Fougères and P. Trigano (1996). The formalisation of specifications frome specifications written in natural language, in Proceedings of *Expersys'96*, Paris-Marne La Vallée, October 21-22.

A.-J. Fougères (1997). *Aide à la rédaction de spécifications formelles à partir de spécifications rédigéees en langage naturel. Application aux spécifications de services de France Telecom*, PhD thesis of the University of Technology of Compiègne.

P. Hernert (1993). *Un système d'acquisition de définitions basé sur le modèle des graphes conceptuels*, Phd thesis of the University Paris XIII.

R. M. Kaplan, J. Bresnan (1982). Lexical-Functional Grammar: A Formal System for Grammatical Representation, in J. Bresnan ed., The Mental Representation of Grammatical Relations, MIT Press.

I. Sommerville (1992). Software engineering, Addison-Wesley, Reading, MA.

J. F. Sowa (1984). Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley Publishing Company, Reading, MA.

J.M. Spivey (1992). The Z Notation - A Reference Manual, Prentice Hall International (UK) Ltd.

Y. Toussaint (1992). *Méthodes informatiques et linguistiques pour l'aide à la spécification de logiciel*, PhD thesis of the University Toulouse III.