THE TWO FUNCTIONALITIES OF THE SAREL SYSTEM

Núria Castell and Àngels Hernández[1)]

Abstract

The specification phase is one of the most important and least supported part of the software development process. We have conceived SAREL (Assistance System for Writing Software Specification in Natural Language) as a knowledge-based tool to improve the specification phase. The purpose of SAREL is to assist engineers in the creation of software specifications written in natural language. The functionality of the system is twofold: vertical processing and horizontal processing. In vertical processing the input is a software specification written in natural language and the output is the conceptual representation associated. In this case the system validates every requirement taking into account the writing norms and the quality properties. The conceptual representation can be used to check the software quality factors. Working in horizontal processing the input is two different conceptual representations and the output is the information about the correspondence between them.

1. Introduction

The specification phase is one of the most important and least supported parts of the software development process. In this stage it is very important to control the writing of the Software Requirements Specifications because many more mistakes can be detected if the writing is clear and concise. A complete study of the problems arising from NL specifications can be found in [7]. In complex systems, documentation writing is guided by the norms which define the linguistic restrictions required and also by the software engineering constraints related to the quality factors of the software specifications.

[1] Universitat Politècnica de Catalunya, Dept. LSI, Jordi Girona 1-3. Mòdul C6, 08034 Barcelona, Spain, e-mail: {castell,ahernandez}@lsi.upc.es. This work is partially supported by CICYT (TIC96-1243-C03) and CIRIT (1997SGR51).

In order to obtain a high-quality software requirements specification we have designed the SAREL help-system (Assistance System for Writing Software Specifications in Natural Language). This is a knowledge-based system, where some linguistic engineering tools are applied. The controls contained within this system can be grouped into three modules: the Style Refinement Module, the Conceptual Refinement Module and the Software Quality Control Module. The last two modules use the Knowledge Base which contains the domain knowledge and the Requirements Base which contains the requirements representation.

2. The SAREL system and its functionalites

The main goal of SAREL is to assist an engineer in the creation of quality preliminar software specifications written in natural language. The assistance process validates every requirement introduced by the engineer taking into account the writing norms (for instance [2]) and the quality properties [4]: consistency, completeness, traceability, modifiability, and verifiability (among others stated by IEEE Standards). This process incrementally constructs a conceptual representation of the software specification. The controls in SAREL can be grouped into three modules:

- The *Style Refinement Module* controls the requirement according to the writing norms. This control is split into four steps: (1) the lexical analysis verifies that the words belong to the application domain lexicon; (2) the syntactic-semantic analysis is performed and a tree-like semantic representation is produced; (3) the ambiguity control helps the engineer to identify the correct representation between the possible interpretations; (4) the simplicity control detects whether the structure of the sentence is simple or compound.
- The *Conceptual Refinement Module* validates the requirement in relation to the Requirements Base. At first it obtains a conceptual representation using the Knowledge Base and after that it detects duplicated information.
- The *Software Quality Control Module* carries out a set of optional analyses to validate the global Requirements Base increased with the new requirement. The goal is to offer information about the above mentioned software quality properties.

The SAREL system has two different funcionalities depending on the user's goal:

- The *Vertical Processing* basically corresponds to the sequential application of the controls. The input is a software specification written in natural language and the output is its associated Conceptual Representation. The document is processed, requirement after requirement, by the Style Refinement Module and the Conceptual Refinement Module in order to obtain the Conceptual Representation that can be optionally validated by the engineer using the Software Quality Control Module. Once a requirement has been checked, its conceptual representation is added to the Requirements Base. Using this functionality, it is possible to obtain the Conceptual Representation associated with the preliminar software specification. This means that the information represented can be consulted in a collective or individual way by the engineers in a more reliable format. A more precise description of this functionality can be found in [3].

- In *Horizontal Processing* the input is of two different conceptual representations, and the goal here is to offer information about the correspondence between them. The correspondence analysis searches for every $requirement_1$ in $document_1$ its corresponding $requirement_2$ in $document_2$. Where $document_1$ corresponds to the user company that needs to develop a computer system and therefore, $document_2$ corresponds to the software company that will carry this out. The system will give a correspondence measure based on similarity analysis [8] applied to the components of the requirements. Depending on the value of this measure, the correspondence will be tagged as: Correct, Excess or Excess-Insufficient. See [5] for a more precise description.

3. The Knowledge Base

The Conceptual Refinement Module uses a domain representation to generate the Conceptual Representation of the requirement that is being processed. Our Knowledge Base contains this domain representation using a frame-based formalism [1].

As with other systems we take into account the three essential parts in a Software Requirements Specification [2]. We can distinguish the *introduction* and the *overall description* as parts that should be used in the Knowledge Base construction. Upto a point these two parts contain all the background information needed to understand the problem as a whole. By use of information extraction techniques [9] the entities involved in the application domain can be obtained.

The information contained in the *specific requirements section* will correspond to the information represented in the Requirements Base. All the entities and all the activities contained in a given requirement must correspond to entities and activities represented in the Knowledge Base.

4. The Requirements Base

As set out above, the Requirements Base contains the requirements representation. The creation of this Base is undertaken requirement by requirement, this process incrementally constructs a conceptual representation of the software specification. Taking into account the two functionalities of the SAREL system, the Requirements Base can be used in two different ways.

From the Vertical Processing point of view, the Requirements Base is used by the analysis tasks [4] contained in the Software Quality Control Module. For example the traceability analysis provides the engineer with information about the traceability links of one fixed requirement. In this case SAREL activates a set of existing algorithms which control the traceability taking into account the relationships between the introduced requirement and a subset of requirements of the Requirements Base. The relationships are based on the entities and the relations stated in the requirements. From this information the engineer can evaluate the level of traceability of a set of requirements.

From the Horizontal Processing point of view, the correspondence analysis searches in the Requirements Base for every requirement$_1$ in document$_1$ its corresponding requirement$_2$ in document$_2$. During this search it is useful to take into account information about what kind of requirement requirement$_1$ is (Functional, Performance or Interface). The measuring mechanism and the software requirements modelling used in SAREL can be found in [6]. This corresponding analysis will be improved giving consideration to the concepts contained in both Requirements Bases instead of the individual requirements.

5. The SAREL system at present.

For the Style Refinement Module, we have adapted some linguistic tools developed in our department : a morphological analyzer (*Maco+*), a POS tagger (*Relax*), and a syntactic chart-based parser (*Tacat*). Public access to our tools: http://www.lsi.upc.es/~acquilex/nlrg.html. In order to construct the Knowledge Base we are adapting the *Turbio* system [9] output to the frame formalism *Yaya* [1]. For the Software Quality Control Module, SAREL adopts the results of previous works [4] related to the modifiability and traceability analysis.

References

[1]Alvarez, J., Yet Another Yet Another (YAYA). Technical report, LSI-96-15-T Dept. of LSI, Universitat Politècnica de Catalunya, 1996.

[2]IEEE Std 830-1993: IEEE Recommended Practice for Software Requirements Specifications, 1993.

[3]Castell, N. & Hernández, A., Filtering Software Specifications Written in Natural Language, in: Proceedings of the 7th Portuguese Conference on Artificial Intelligence (EPIA'95), LNAI 990, pp 447-455, Funchal, Madeira Island, Portugal, 1995.

[4]Castell, N. & Slavkova, O. & Toussaint, Y. & Tuells, A., Quality Control of Software Specifications written in Natural Language, in: Proceedings of the 7th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE'94), pp 37-44, Austin, Texas, USA, 1994.

[5]Castell, N. & Hernández, A., The use of SAREL to control the correspondence between Specification Documents, in: Proceedings of VII Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA'97), pp 529-539, Málaga, Spain, 1997.

[6]Castell, N. & Hernández, A., The Software Requirements Modelling in SAREL, in: Proceedings of the 4th International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ'98), pp 49-56, Pisa, Italy, 1998.

[7]MELCHISEDECH, R., Investigation of Requirements Documents Written in Natural Language, in: Requirements Engineering (1998) 3:2, pp 91-97.

[8] ROMESBURG, H.C., Cluster analysis for researchers. Belmont, Calif.: Lifetime Learning Publications, 1984.

[9]TURMO, J. & Català, N. & Rodríguez, H. TURBIO: A System for Extracting Information from Restricted Domain Texts. (IEA/AIE'98), LNAI 1415, pp 708-721, Benicàssim, Spain, 1998.