On Linguistic Quality of Natural Language Requirements

F. Fabbrini[1], M. Fusani[1], V. Gervasi[2], S. Gnesi[1], S. Ruggieri[2]

[1] *IEI-CNR, via S. Maria 46, 56125 Pisa Italy.*

[2] *Dipartimento di Informatica, Corso Italia 40, 56125 Pisa Italy.*

Abstract. We recognize the need for defining a quality model of linguistic properties of natural language requirements. We are convinced that the definition of a thorough linguistic quality model can be obtained as a meeting point between the specialization of high level quality frameworks for requirements engineering on one hand, and a systematic collection and classification of the different Natural Language Processing techniques proposed in the literature specifically for the analysis of requirements.

1 Motivations

Quality in software engineering is following, to some extent, the same historical path as quality in classical engineering has. Many methods and tools have been proposed in the past to achieve some specific quality objectives. However, the vast majority of them are the outcome of a painful trial-and-error process. Stories of projects ended in uttermost disaster have taught us what *not* to do in a software project, while experiences with successful projects, together with some common sense, have taught us what to do. Such an approach, although useful during the childhood of our discipline, has now reached its limits. Like classical engineering switched from quality control to total quality management, we need to switch from "best practices" to a systematic, comprehensive, and unifying approach in understanding and analyzing problems.

In this paper, we concentrate on quality of natural language (NL) requirements. Natural language properties deserve a particular attention since the main purpose of a requirements document is to convey information, gathered from the customer and other sources, to the developers, and this communication is mediated by the language used. In recent years, several attempts have been conducted in order to apply general Natural Language Processing (NLP) techniques to the purpose of analyzing and controlling software requirements. As an example, some works (e.g., [12]) investigated how the construction of a conceptual model of a problem is related to the linguistic mechanisms through which analysts abstract natural language requirements onto concepts. Unfortunately, the effectiveness of the proposed techniques and tools in affecting quality of requirements has not been demonstrated in most cases. The reason lies in the fact that no systematic classification of the relevant linguistic properties of requirements documents has been pursued. "Quality models" proposed in the past consisted of bread and butter lists of properties or, in jargon, *-ilities*. On the contrary, we argue in favor of the definition of a thorough quality model of linguistic properties of NL requirements documents.

2 Quality models

A generic quality model of a product consists of the definition of a set of well-identified product properties that *"bear on its ability to satisfy stated or implied needs"* [1] of its stakeholders. A quality model can be simply an enumeration of properties, or, better, be structured hierarchically in properties, sub-properties, etc. The properties defined in a quality model are called *factors*, and they are often indicated as *goal properties*. Usually, quality factors are defined aiming at a minimum overlap and thus maximum orthogonality. This allows each factor to be controlled separately. In addition, quality models are complemented by a set of support properties called *criteria*, which serve as indicators or predictors of factors.

The objective of measuring quality is to determine the extent to which a product exhibits the properties defined in the quality model. To this purpose, a generic quality model must be instantiated into a specific quality model for a given product. Quality models are also complemented by a collection of *activities* that influence criteria and hence factors. Performing an activity is then a means to control the overall quality of a product, hopefully improving it.

In particular, the advantages of defining a quality model of requirements include descriptive and prescriptive uses of the model in the analysis and control of requirements documents quality; planning, control and verification of the expected quality levels; establishment of a company quality improvement programme; classification, comparison, selection and aggregation of the most appropriate (in terms of effectiveness, budget and schedule) tools and techniques for writing, checking and evaluating requirements.

3 Our approach

The rationale of our approach is as follows. We are convinced that the definition of a thorough linguistic quality model can be obtained as a meeting point between two ways of looking at quality in requirements engineering.

On one hand, in recent years some high level quality frameworks of requirements and requirements engineering have been proposed, with the intention of capturing the basic products, agents, and relationships that are involved in the requirements engineering process. In this research area, we point out the proposals of Lindland et al. [6] and of Pohl [11], that have been further merged and refined by Krogstie et al. [5]. It is worth noting that a distinguishing feature of the last framework is that it can be closely linked to linguistic concepts.

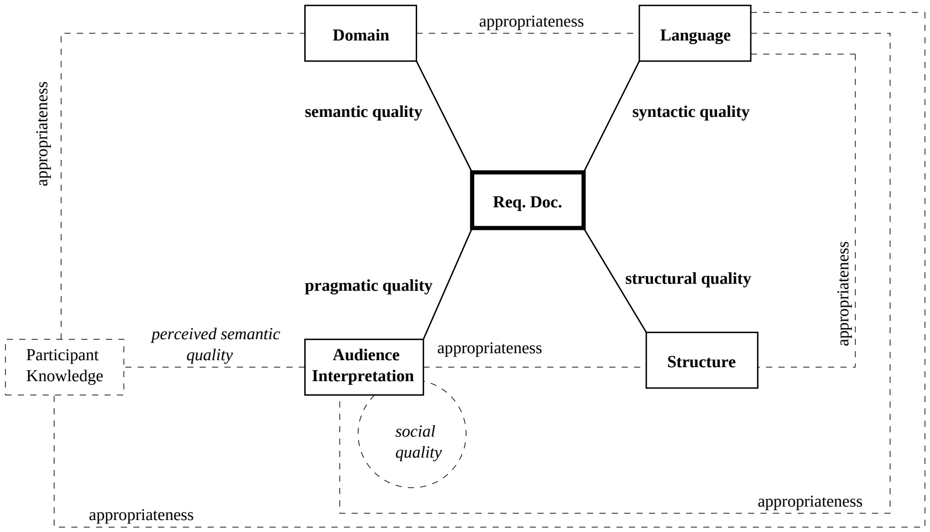

Fig. 1. A linguistic quality framework for NL requirements.

On the other hand, several specific NLP techniques have been tailored to the purpose of analyzing and controlling software requirements. Among the others, we cite Macias and Pulman [7, 8], Rolland and Proix [12], Ambriola and Gervasi [2], Fantechi et al. [3], Nelken and Francez [9]. The kind of techniques investigated is rather different, ranging from purely lexical analysis for the extraction of abstractions from natural language texts by Goldin and Berry [4], to speech act modeling by Nellborn and Holm [10]. Moreover, tools have been developed for most techniques.

Summing up, the state of the art is that on one hand a few high level quality frameworks have been proposed, and that on the other hand several techniques and tools are available for specific tasks. Starting from these considerations, we are investigating the definition of a linguistic quality model as the meeting point of those two approaches. We proceed along a top-down and a bottom-up way.

In our top-down analysis, we chose to consider a modification of the framework by Krogstie et al. [5], specializing it on purely linguistic aspects. The resulting linguistic framework consists of the relationships highlighted in Figure 1. Central to the framework is the *requirements document*, that is related to the *language* used, to the desired *structure* for the document, and to each participant's interpretation of the document (*audience interpretation*). Moreover, the requirement document is related to the *domain*, that is the ideal set of all correct and relevant statements pertaining to the problem at hand. Interestingly, those relations coincide with the *syntactic*, *structural*, *semantic* and *pragmatic* components of NL documents.

The bottom-up way starts from a collection of existing NLP techniques and best practices, and tries to abstract their common functionalities. Sources for

Top-Down		*Meeting*	*Bottom-Up*
Quality Types	**Factors**	**Criteria**	**Activities**
Syntactic	Syntactic Validity	Lexicon Correctness	Lexicon Checking Controlled Vocabulary
		Syntactic Correctness	Syntax Checking Syntax-Driven Writing
Structural	Structural Validity	Structure Correctness	Structure Checking Template Documents
	Structural Completeness	Structural Unit Presence	Checklists Template Documents
		Structure Modifiability	Hard-Reference Check. Tracing
Semantic	Semantic Validity	Domain-Independent Correctness	Ambiguity Checking Semantic-Driven Writing
		Domain-Dependent Correctness	Domain-based Modeling Semantic-Driven Writing
	Semantic Completeness	Domain-Independent Completeness	Consistency Checking Speech Act Modeling
		Domain-Dependent Completeness	Case Reasoning Intelligent Dialogue
Pragmatic	Valid Comprehension	Presentation Ordering	Fwd. Reference Checking Structural Modeling
		Readability	Readability Formulae Style Guidance
	Complete Comprehension	Navigability	Browsing Hypertext Editing
		Conciseness	Document Size Size Limitation

Table 1. Simplified linguistic quality model of NL requirements documents.

this collection included journals like ACM TOSEM, IEEE TSE, IEEE Software, together with more specialized ones like the Requirement Engineering Journal or the Knowledge Engineering Review, as well as project reports series, books, and a number of conference proceedings.

We obtained the meeting point of these two approaches by using the linguistic quality framework as a superordinate rationale supporting the classification of NLP techniques into homogeneous groups, by refining Krogstie et al.'s model into finer elements and by extending it according to the results of the study of existing NLP techniques. In other words, the meeting point is a quality model that aims to be comprehensive enough to embrace all properties of requirements documents that affect their linguistic quality, and sufficiently specific for being applied to classify existing and future techniques, tools, and practices. A simplified version of the quality model is shown in Table 1. Quality types are derived from the relations of Figure 1, while factors further refine those relations. Activities correspond to techniques proposed in the literature, while criteria are defined as the meeting point of specializing factors and abstracting activities. For the sake of the example, we only cited two activities for each criteria, namely one aimed at evaluating quality and one aimed at inducing it during the requirement production process.

4 Open issues

While the applications and the uses of the linguistic quality model are clear, some issues needs further investigation. First, the proposed model must be validated on real case studies both for descriptive and prescriptive uses of the model. In this sense, the availability of automatic tools seems of primary importance. It is worth noting, however, that specific tools for basic techniques already exist. What we need, is then some glue layer. Moreover, we observe that further guidelines are necessary for applying the quality model in practice. In particular, additional classifications of activities could be provided that are orthogonal to the criteria classification: for example, activities might be grouped according to the process phase in which they are performed, or to the agents performing them, or to the objects they are performed on, or to the preconditions required for their applicability, or to their effects over criteria, and so on. Second, a measurement theory must be developed both on the theoretical side and on the experimental validation side, providing a collection of metrics for measuring criteria and a fortiori quality. Along this direction, some works from computational linguistic (e.g., readability formulae) might be a basic starting point. Third, methods for the assessment and certification of quality must be defined starting from the proposed quality model.

References

1. ISO/IEC 8402. *Quality management and quality assurance – Vocabulary.* International Organization for Standardization, 1994. Edition 2.
2. V. Ambriola and V. Gervasi. Processing natural language requirements. In *Proceedings of ASE 1997*, pages 36–45. IEEE Press, 1997.
3. A. Fantechi, S. Gnesi, G. Ristori, M. Carenini, M. Vanocchi, and P. Moreschini. Assisting requirement formalization by means of natural language translation. *Formal Methods in System Design*, 4(3), 1994.
4. L. Goldin and D. M. Berry. Abstfinder, a prototype abstraction finder for natural language text for use in requirements elicitation: Design, methodology, and evaluation. In *Proceedings of the First International Conference on Requirements Engineering*, 1994.
5. J. Krogstie, O. I. Lindland, and G. Sindre. Towards a deeper understanding of quality in requirements engineering. In *7th International CAiSE Conference*, volume 932 of *Lecture Notes in Computer Science*, pages 82–95, 1995.
6. O. I. Lindland, G. Sindre, and A. Sølvberg. Understanding quality in conceptual modeling. *IEEE Software*, 11(2):42–49, March 1994.
7. B. Macias and S. G. Pulman. Natural language processing for requirements specifications. In Redmill and Anderson, editors, *Safety-critical Systems*, pages 57–89. Chapman and Hall, 1993.

8. B. Macias and S. G. Pulman. A method for controlling the production of specifications in natural language. *The Computer Journal*, 38(4):310–318, 1995.
9. R. Nelken and N. Francez. Automatic translation of natural-language system specifications into temporal logic. In *Proceedings of CAV'96*, number 1102 in Lecture Notes in Computer Science, pages 360–371, 1996.
10. C. Nellborn and P. Holm. Capturing information systems requirements through enterprise and speech act modeling. In *6th International CAiSE Conference*, volume 811 of *Lecture Notes in Computer Science*, pages 172–185, 1995.
11. K. Pohl. The three dimensions of requirements engineering: a framework and its applications. *Information Systems*, 19(3):243–258, 1994.
12. C. Rolland and C. Proix. A natural language approach for requirements engineering. In *4th International CAiSE Conference*, volume 593 of *Lecture Notes in Computer Science*, pages 257–277, 1992.