A New Quality Model for Natural Language Requirements Specifications

Daniel M. Berry[1], Antonio Bucchiarone[2], Stefania Gnesi[2], Giuseppe Lami[2], and Gianluca Trentanni[2]

[1] Cheriton School of Computer Science, University of Waterloo
Waterloo, Ontario N2L 3G1, Canada
dberry@uwaterloo.ca)

[2] Istituto di Scienze e Tecnologia dell'Informazione del C.N.R
I-56124 Pisa, Italy
{antonio.bucchiarone,stefania.gnesi,giuseppe.lami,
gianluca.trentanni}@isti.cnr.it)

Abstract. This paper describes an extension to the natural language requirements specification quality model that is the basis for the QuARS (Quality Analyzer of Requirements Specification) tool. The extension takes into account ambiguities that were not handled before.

1 Introduction

In the requirements elicitation (RE) phase of a project to develop a computer-based system (CBS), knowledge about the CBS under construction is acquired. The output of this phase is a document, called the *requirements specification* (RS), specifying the requirements for the system, that is written usually in a natural language (NL). The use of NL for specifying requirements has some advantages such as the ease with which the RS can be shared among the stakeholders of the CBS, including the customers and the developers. In fact, a NL RS can be used in different ways, and in different phases of the development of the CBS. It may be provided as input to the architects, the coders, the testers, and the user manual writers. It may be used also as a contract between the customer and the developer or as a source of information for the project managers [1].

Because of NL's universal accessibility, mostly NL is used in the software industry for specifying CBS requirements, despite NL's inherent ambiguity and informality that make determining a NL document's correctness difficult. There are other problems with NL RSs, such as

- the volatility of a CBS's requirements during the CBS's development, leading to many changes to the CBS's RS.
- the large variation in people's writing skills, leading to large variations in the linguistic quality of a RS,
- the large number of sources for a RS, leading to inconsistent linguistic styles within one RS [2]

These problems make writing a RS in NL a higly risky venture if quality of the RS is paramount.

There is certainly a need to evaluate and improve the quality of a NL RS. However, evaluating and improving—i.e., *proofing*—a RS are time consuming, tedious, and error prone when they are done by a human being. Automated tool support for the clerical parts of the proofing task would free up the human for the parts of the task requiring human intelligence. Such a tool will have to process the NL text of any input RS.

Two different approaches have been used to construct automatic NL text processors:

– the linguistic approach, based on a parse of the text into a structure of its linguistic components [3], and
– the statistical approach, based on frequencies of elements of the text, often determined by breaking the text into only lexical elements [4],

This paper provides a linguistic method for analysis of NL RSs, based on a quality model (QM) [5] developed by some of the authors of this paper and some others. This QM is hereafter called "QM1", to distinguish it from the QM, called "QM2", that results after adding the enhancements proposed in this paper. QM1 addresses a significant portion of expressiveness and structural consistency problems of NL RSs that can be dealt with at the linguistic level. The developers of QM1 have implemented QuARS (Quality Analyzer of Requirement Specification) [6], an automatic tool supporting proofing of NL RSs based on QM1. QM1 has been used successfully to proof several RSs obtained from industrial projects [7–9].

2 General QM for NL RSs

A general model for the quality of a NL RS is shown in Table 1. It shows that there are

Quality Characteristics		Manifestation			
Family	Subfamily	Lexical	Syntactic	Structural	Semantic
Understandability	Readability				
	Uniguity				
	Testability				
Consistency					
Completeness					
Corectness					

Table 1. General QM for NL RSs

three families of quality characteristics:

– understandability: the ease by which humans understand the meaning of the NL RS at hand,

- consistency: the lack of contradictions and incongruities in the NL RS at hand,
- completeness: the lack of missing information in the NL RS at hand, and
- correctness: (1) the lack of factual errors about the domain of the CBS under construction and (2) matching what the customer wants.

The understandability family has three subfamilies:

- readability: the ability for a human being to read the NL RS at hand,
- uniguity[1]: the lack of ambiguity in the NL RS at hand, and
- testability: the ability to test that a CBS implements what is specified in the NL RS at hand.

These subfamilies can overlap. For example, an ambiguous sentence is not testable, and an unreadable sentence may be ambiguous.

Each family or subfamily can have up to four manifestations in the NL RS at hand.

- lexical: involving words in the NL RS at hand,
- syntactic: involving the parses of sentences in the NL RS at hand,
- structural: involving physical relationships between parts of the NL RS at hand, and
- semantic: involving meanings of parts of the NL RS at hand.

Sometimes, a characteristic can have more than one manifestation. For example, there is a potential *meaning problem* with the word "only" when it appears *immediately before the main verb of a sentence*.

The portion of this general QM that can be used in a purely linguistic tool must necessarily exclude the last, "Semantic", column of the table. Nevertheless, many a potential problem that can occur in a NL RS can be detected by purely linguistic means. We say "potential problem" because it may be necessary to take semantics into account to determine if the potential problem is indeed a real problem. For example, the potential meaning problem with the word "only" when it appears immediately before the main verb of a sentence is a real problem only if the writer intended to limit with "only" a word in the sentence other than the main verb. Fortunately, when a human being is shown a linguistically detected potential problem, he or she can look at it and usually determine by his or her understanding of meaning whether the potential problem is a real problem.

Even a characteristic that is traditionally thought of as semantic can have a non-semantic manifestation and thus be detectable by purely linguistic means. For example, the consistency family includes characteristics dealing with the lack of structural contradictions and incongruities in the NL RS at hand. Thus, the consistency family includes the following processes:

- Mutual referencing checking: detection and correction of missing or incorrect references in the NL RS;
- Clustering for discrepancy detection: grouping of portions of the NL RS that have similar contents in order to facilitate determination by a human of discrepancies, conflicts, and duplications.

[1] We have invented the word "uniguity", i.e., *one meaning*, to not have to use the heavy mouthful "nonambiguity".

Linguistic methods can deal also with some issues of correctness. For example, in some circumstances, it might be worth searching for specific phrases that make clearly incorrect statements, such as "never make mistakes" in "The authors of this paper never make mistakes."[2]

Of course, not all RS quality issues can be addressed with the same depth and the same ease. In fact, formally validating that the CBS defined by the RS is the CBS that the client wants is impossible. Verifying that the CBS matches its RS needs the power of rigorous formal methods [10, 11] and is beyond the power of either a linguistic or statistical approach.

3 QM2

The enhancement to QM1 to make QM2 is to include those ambiguities described by Berry, Kamsties, and Krieger in various documents [12–15] that are not already in QM1.

QM1, and thus QM2, consists of a set of indicators of low quality in NL RSs, each element of which can be detected linguistically, i.e., using lexical, syntactic, and structural processing of the NL RS at hand. Each indicator falls under one or more families of subfamilies of characteristics. QM2 specifies that both the single sentences of a RS and the whole RS are evaluated. Each indicator measures the quality of a sentence in the RS or the RS itself. In other words, each indicator has its own scope.

QM1 and QM2 are described simultaneously by a table, which has been split into page-sized subtables, Tables 2, 3, 4, 5, and 6. The portion of the table that describes the original QM1 is typeset with a sans serif typeface. The portion of the table that describes the extensions to QM1 that make QM2 is typeset with a serif typeface. Thus, the full table describes QM2.

For any row, the "Indicator / (Manifestations)" column names one possible indictor of low quality and gives its manifestations in parentheses after the name; the "Scope" column tells whether the scope of the indicator is a single sentence or the whole RS; the "Characteristics / (Sub)family" column entry for the row gives the characteristic family or subfamily to which the indicator belongs; the "Description" column entry for the row describes the nature of the indicated problem; the "Notes" column entry for the row gives some additional notes relevant to the description of the nature of the indicated problem; and the "Negative Examples" column entry gives one or more examples of a sentence with the indicated problem, with each instance of the indicator in bold face.[3] Note that the names for the indicators were chosen after analysis of several industrial RS documents [5, 16–19].

QM2 does not cover all possible defects in RS quality, but it is sufficiently specific that it can be applied, with the support of an automatic tool, to verify the quality of NL RSs, perhaps on a comparative basis. Furthermore, the set of properties in QM2 is

[2] The description of QM2, in Section 3, has some better examples, but we do not want to reveal them now!

[3] Space limitations precludes describing any indicator both in the text and in the table. Since the tables are intended to be portable to any paper, they must be complete. Hence, all descriptions of indicators are in the tables.

sufficient to include a large fraction of the syntactic defects of a RS along with a number of interesting semantic defects.

Only those cases of an indicator that are more likely than not to be erroneous are detected. For example, consider the “only” variant of the “misplaced limiter” indicator. The only time an instance of “only” is to be reported as a potential problem is when it immediately precedes the main verb of its sentence. If an “only” appears elsewhere it its sentence, it is assumed to be correctly placed. Since it is so unusual in English for an “only” not to be immediately preceding the main verb of a sentence, an “only” elsewhere has probably been placed where it is consciously and correctly after a bit of careful thought. It would not be worth reporting such an “only”. Of course, the fairness of this assumption must be checked during the validation of the tool suggested for future work.

The tool QuARS is applied to a document which has been broken into sentences. Whenever QuARS finds at least one potential problem categorized in QM1 in a sentence, QuARS highlights the sentence and gives the indicator for each potential problem QuARS has found in the sentence. It is left to the user to determine if a potential problem is a real problem and if necessary to correct it.

4 Future Work

The tool QuARS must be extended to search for the new indicators that have been added to QM1 to make QM2. The resulting new QuARS has to be tested for effectiveness and usefulness on real-life RSs.

The current version of QuARS only[4] highlights each so-called offending sentence that contains at least one instance of an indicator of low quality. Since QuARS uses a parser to identify the parts of a sentence before searching for indicators in it, a possible extension for QuARS is for QuARS to construct something that can alert the user as to the exact difficulty with an offending sentence.

Perhaps for an ambiguity, QuARS could display the alternative constructions. For example, with “You get a soup or a salad and a vegetable.”, QuARS could show, “You get (1) a soup or (2) a salad and a vegetable. AND You get (1) a soup or a salad and (2) a vegetable.” or “You get (a soup) or ((a salad) and (a vegetable)). AND You get ((a soup) or (a salad)) and (a vegetable).”

Perhaps for an arbitrary offending sentence, QuARS could formulate a question or clarifying sentence that makes it clear what the offending sentence actually says. For example, with the offending sentence, “I only nap after lunch.”, QuARS could ask, “Are you sure that: The only thing I do after lunch is nap?” or “Are you sure that: I ONLY nap and not do something else after lunch?”

These possible extensions to QuARS need to be explored.

As a test of the effectiveness of QuARS, each NL document written about either QM or QuARS should be submitted to QuARS for an evaluation of its writing quality.

[4] Notice that this “only”, though preceding the main verb, is in the right place, since the author’s intent is to limit the verb, to say that the only thing that is done to an offending sentence is to highlight it!

Doing so would amount to a good self test. Also, it would help eliminate the potential of embarrassment coming from not following our own advice.

References

1. Power, N.: Variety and quality in requirements documentation. In: Proceedings of the Seventh International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ). (2001)
2. Regnell, B., Beremark, P., Eklundh, O.: A market-driven requirements engineering process: Results from an industrial process improvement programme. Requirements Engineering **3** (1998) 121–129
3. Allen, J.: Natural Language Understanding. second edn. Addison-Wesley (1995)
4. och Dag, J.N., Regnell, B., Carlshamre, P., Andersson, M., Karlsson, J.: Evaluating automated support for requirements similarity analysis in market-driven development. In: Proceedings of the Seventh International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ). (2001)
5. Fabbrini, F., Fusani, M., Gnesi, S., Lami, G.: Quality evaluation of software requirement specifications. In: Proceedings of the Software and Internet Quality Week 2000 Conference. (2000)
6. Gnesi, S., Lami, G., Trentanni, G., Fabbrini, F., Fusani, M.: An automatic tool for the analysis of natural language requirements. International Journal of Computer Systems Science and Engineering **20** (2005)
7. Bucchiarone, A., Gnesi, S., Pierini, P.: Quality analysis of nl requirements: An industrial case study. In: Proceedings of the Thirteenth IEEE International Conference on Requirements Engineering (RE'05). (2005) 390–394
8. Bucchiarone, A., Gnesi, S., Winzen, A., Fantechi, A.: A quality evaluation process for collaborative requirements: An industrial case study. Technical report, ISTI-CNR, Pisa, Italy (2006)
9. Lami, G., Ferguson, R., Goldenson, D., Fusani, M., Fabbrini, F., Gnesi, S.: Automated natural language analysis of requirements and specifications. In: Proceedings of the INCOSE (International Council on System Engineering) International Symposium. (2004)
10. Meyer, B.: On formalism in specifications. IEEE Software **2** (1985) 6–26
11. Wing, J., Woodcock, J., Davies, J.: FM'99—Formal Methods, Volumes I & II. LNCS 1708. Springer (1999)
12. Berry, D., Kamsties, E.: The dangerous "all" in specifications. In: Proceedings of 10th International Workshop on Software Specification & Design (IWSSD-10), IEEE CS (2000) 191–194
13. Berry, D., Kamsties, E., Krieger, M.: From contract drafting to software specification: Linguistic sources of ambiguity. Technical report, University of Waterloo, Waterloo, ON, Canada (2003) `http://se.uwaterloo.ca/˜dberry/handbook/ambiguityHandbook.pdf`.
14. Berry, D., Kamsties, E.: Ambiguity in requirements specification. In: Perspectives on Requirements Engineering, Kluwer (2004) 7–44
15. Berry, D., Kamsties, E.: The syntactically dangerous *all* and plural in specifications. IEEE Software **22** (2005) 55–57
16. Fabbrini, F., Fusani, M., Gervasi, V., Gnesi, S., Ruggieri, S.: On linguistic quality of natural language requirements. In: Proceedings of the Fourth International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ'98). (1998)

17. Fantechi, A., Fusani, M., Gnesi, S., Ristori, G.: Expressing properties of software requirements through syntactical rules. Technical report, IEI-CNR, Pisa, Italy (1997)
18. Fantechi, A., Gnesi, S., Ristori, G., Carenini, M., Vanocchi, M., Moreschini, P.: Assisting requirement formalization by means of natural language translation. Formal Methods in System Design **4** (1994) 243–263
19. Lami, G.: Towards an automatic quality evaluation of natural language software specifications. Technical Report B4-25-11-99, IEI-CNR, Pisa, Italy (1999)
20. Schwertel, U.: Controlling plural ambiguities in Attempto Controlled English. In: Proceedings of the Third Internaltional Workshop on Controlled Language Applications (CLAW), Seattle, WA, USA (2000)

Indicator (Manifestations)	**Scope**	**Characteristic (Sub)-families**	**Description**	**Notes**	**Negative Examples**
Vagueness (Lexical)	Sentence	Testability	The sentence includes an inherently vague word, that is a word having a non uniquely quantifiable meaning.	Vagueness-revealing words: *adequate*, *bad*, *clear*, *close*, *easy*, *efficient*, *far*, *fast*, *good*, *in front*, *near*, *recent*, *significant*, *slow*, *strong*, *useful*, . . .	The C code shall be **clearly** commented.
Subjectivity (Lexical)	Sentence	Testability	The sentence refers to personal opinions or feelings.	Subjectivity-revealing words: *as* [adjective] *as possible*, *best*, *better*, *having in mind*, *similarly*, *similar*, *take into account*, *take into consideration*, *worse*, *worst*, . . .	**To the largest extent possible**, the system shall be built from commercially available software products.
Optionality (Lexical)	Sentence	Testability	The sentence contains an optional part, i.e., a part that may but does not need to be considered.	Optionality-revealing words: *possibly*, *eventually*, *in case*, *if possible*, *if appropriate*, *if needed*, . . .	The system shall be such that the mission can be pursued, **possibly** without performance degredation.
Weakness (Syntactic)	Sentence	Testability	The sentence contains a weak main verb.	Weakness-revealing words: *can*, *could*, *may*, . . .	The results of the initialization checks **may be** reported in a special file.
Multiplicity (Lexical & Syntactic)	Sentence	Uniguity & Testability	The sentence has more than one main verb or more than one direct or indirect complement that describes the sentence's subject.	Multiplicity-revealing words: *and*, *and/or*, *or*, . . .	The mean time needed to remove a faulty board and restore service shall be less than 30 minutes.
Implicity (Lexical & Syntactic)	Sentence	Uniguity	The sentence's subject is generic rather than specific.	The subject is expressed by means of: demonstrative adjective (*that*, *these*, *this*, *those*, . . .), pronoun (*he*, *it*, *she*, *they*, . . .), positional adjective (*following*, *last*, *next*, *previous*, . . .), positional preposition (*above*, *below*, . . .), . . .	The **above** requirements shall be verified by test.

Table 2. Quality Model QM2 in Tabular Form, Part 1

Indicator (Manifestations)	Scope	Characteristic (Sub)-families	Description	Notes	Negative Examples
Coordination (Lexical & Syntactic)	Sentence	Uniguity	The sentence includes more than one occurrence of conjuctions *and*, *or*, or both.	The precedences of the conjunctions are unclear.	You get a soup or a salad and a vegetable. I saw Peter and Paul and Mary saw me.
Coordination (Lexical & Syntactic)	Sentence	Uniguity	The sentence includes an adjective preceding a phrase containing at least one conjuction, *and* or *or*.	The scopes of the adjective and the conjunction are unclear.	young man and woman
Coordination (Lexical & Syntactic)	Sentence	Uniguity	The sentence includes a *not* preceding a phrase containing at least one conjuction, *and* or *or*.	The scopes of the *not* and the conjunction are unclear.	The system shall not give out secrets and open files.
Negation of Causality (Lexical & Syntactic)	Sentence	Uniguity	The sentence includes a *not* preceding a phrase containing a *because*.	The scopes of the *not* and the *because* are unclear. One way to read the example is that the case was not brought before committee and the incident the night before is what caused the case not to be brought. However, is it definite that the case was not brought before committee at all? One cannot be sure. Another way to read the sentence is that the incident the night before did not cause the case not to be brought, but in fact, the case was brought before committee anyway.	The witness said that the case was not brought before committee because of the incident the night before.

Table 3. Quality Model QM2 in Tabular Form, Part 2

Indicator (Manifestations)	**Scope**	**Characteristic (Sub)-families**	**Description**	**Notes**	**Negative Examples**
Misplaced Limiter (Lexical & Syntactic)	Sentence	Uniguity	The sentence includes a limiter word immediately preceeding the main verb of the sentence.	Limiter words include: *almost*, *also*, *even*, *just*, *mainly*, *merely*, *nearly*, *only*, *really*, *usually*, ... A limiter must precede the word that the limiter limits. However, in English, the norm is to put a limiter immediately preceding the main verb of the sentence, even if the verb is not to be limited. Does the example *I only nap after lunch.* mean *Only I nap after lunch.*, *I only nap after lunch.*, *I nap only after lunch.*, or *I nap after only lunch.*? When a limiter is put before a word other than the main verb, it is highly likely that the limiter was put there correctly, because putting the limiter there goes against the norm.	I only nap after lunch. I also nap after lunch.
Dangerous Plural (Syntactic)	Sentence	Uniguity	The sentence includes a plural subject and plural main verb.	The correspondence between members of the subject set and the main verb's completer is unclear. In the first example, does each of the three girls lift a table or do all three girls together lift a table?	Three girls lift a table. [20] Each term, students take 4 or 5 courses. Each term, students take hundreds of courses.
Number Error (Syntactic)	Sentence	Uniguity	The sentence includes plural pronoun for singular referent.	Either the sentence has a grammatical error or the pronoun refers to a plural noun in a previous sentence. In the example, if *their* refers to *Everybody*, the grammatical error is a plural pronoun referring to a singular subject noun, and a correct version of the sentence is *Everybody knows his or her home address.* If *their* refers to a plural noun in a previous sentence, e.g., as in . . . *the leaders* . . . . *Everybody knows their home address.*, there is no grammatical error.	Everybody knows their home address.

Table 4. Quality Model QM2 in Tabular Form, Part 3

Indicator (Manifestations)	**Scope**	**Characteristic (Sub)-families**	**Description**	**Notes**	**Negative Examples**
Special Implicity (Lexical & Syntactic)	Sentence	Uniguity	The sentence has *this* used as a noun, either as the subject or as an object.	The referent of *this* is unclear. *This* can refer to any amount of text preceding the sentence containing *this*.	This is the reason for the error. The reader can understand this.
Unclear Scope of Quantifier (Lexical & Syntactic)	Sentence	Uniguity	The sentence includes more than one universal or existential quantifier equivalent.	The scope of the quantifiers is unclear. Universal quantifier equivalents include: *all*, *each*, *every*, . . . Existential quantifier equivalents include: *a*, *at least one*, *some*, . . . In the example, does each linguist prefer his or her own theory or is there one theory that each linguist prefers?	Each linguist prefers a theory.
Dangerous Universal (Lexical & Syntactic)	Sentence	Correctness	The sentence includes a universally quantified statement.	An indicative universally quantified statement that is probably false in real life. Indicators of universally quantified statements include: *all*, *each*, *every*, *no one*, *none*, *always*, *never*, . . . Software depending on the correctness of the statement, and not checking that it is true, will not be robust. In the example, a given person might have more than one or no national identity number.	Each person has a unique national identity number.
Nonparenthetical Parentheses (Lexical)	Sentence	Uniguity	The sentence includes at least one parenthesis.	The meaning of the parenthesized expression is unclear.	When there is (not) an alarm, turn the power off (on). In an emergency (fire, earthquake, hurricane), shut off the power.
Dangerous Slash (Lexical)	Sentence	Uniguity	The sentence includes at least one slash, "/".	The meaning of the slash is unclear, particularly in *and/or*.	When the input is incorrect and/or unreasonable, do X. The customers/clients want X.

Table 5. Quality Model QM2 in Tabular Form, Part 4

Indicator (Manifestations)	Scope	Characteristic (Sub)-families	Description	Notes	Negative Examples
Under-specification (Syntactic)	Sentence	Completeness	The subject of the sentence contains a word identifying a class of objects without the word's having a modifier specifying an instance of the class.	A word needing to be more specified: *flow (control flow, data flow, . . .)*, *access (authorized access, read access, remote access, write access, . . .)*, *testing (acceptance testing, functional testing, structural testing, unit testing, . . .)*, . . .	The system shall be able to run also in case of **attack**.
Under-reference (Lexical & Structural)	Whole RS	Completeness	The sentence contains an undefined reference.	Undefined references include: an acronym not defined in the acronym dictionary, a bibliographical citation to an item not in the bibliography, a sentence number not numbering any sentence, a technical term not defined in the glossary, . . .	The software shall be designed **according to the rules of object-oriented design**. The handling of any received valid **TC** shall be started in less than one **CUT**.
Comment Frequency (Structural)	Whole RS	Readability	The document's comment frequency index, CFI = NC / NR, where NC is the total number of sentences in the document having one or more comments, and NR, is the total number of sentences in the document.		
Readability Index (Lexical & Structural)	Whole RS	Readability	The documents's automated readability index, ARI = WS + 9$\times$SW, where WS is the average number of words per sentence in the document, SW is the average number of letters per word in the document		
Directives Frequency (Lexical & Structural)	Whole RS	Readability	The document's directives frequency is the number of instances of a reference, e.g., a line number, a figure number, a table number, a footnote, to another part of the document		

Table 6. Quality Model QM2 in Tabular Form, Part 5