

# Reducing Natural-Language Ambiguities in Requirements Engineering

BY LARS SCHNIEDER AND SUSANNE ARNDT

Interdisciplinary and interorganizational project collaboration is a challenge. One of the most essential tasks in big and heterogeneous projects is requirements engineering, which, done properly, helps master complexity and reduce misunderstanding. Requirements engineering is a communication process that involves understanding and coordinating terminologies from different disciplines. In order to obtain an unambiguous and precise specification of what is required, requirements engineers need to be aware of the potential pitfalls of drawing up requirements in natural language. Requirements engineering needs the support of tools that include models of domain-specific terminologies and conventions about their use to reduce the likely ambiguity and vagueness of requirements expressed in natural language.

re•quire•ment | ri'kwɪrəmənt |

re•quire•ment | ri'kwɪrəmənt |

a thing that is needed or wanted: *choose the type of window that suits your requirements best*

a thing that is compulsory; a necessary condition: *applicants must satisfy the normal*

noun

a thing that is needed or wanted: *choose the  
your requirements best.*

- a thing that is compulsory; a necessary condition: *satisfy the normal entry requirements.*

## Requirements Engineering in Complex Projects

Research and development projects inherently involve a lot of planning and modeling. In general, this work begins with the collection of requirements. These are discussed and modified over time to arrive as far as possible at a requirements specification that is testable, unambiguous, complete, and correct. The project manager of a complex research-and-development project needs to ensure that requirements are adequately administered, guided, controlled, and used throughout the project. For this reason a suitable requirements management process and an adequate tool such as IBM Rational DOORS or IBM RequisitePro need to be in place.

Failure to pay appropriate attention to requirements can have fatal consequences for project success: needs of the user not being met (development of required functions does not happen), resources of money and staff being wasted on unimportant work (for example, on the development of non-required functions), and the greater likelihood of bugs or errors.

## Collaboration

The more complex the research subjects or products, the more care and expertise are required to implement them professionally. This includes a rigorous collaborative approach toward requirements engineering.

First of all, researchers of different departments of one organization work together to specify required technical features of the system (intraorganizational collaboration). In addition, external suppliers need to understand the expected deliverable. Customer and supplier need to find a common language (interorganizational collaboration). Complex projects often require contributions from different domains. Since no single domain can completely capture all relevant requirements, experts from various disciplines need to get together—and they must understand each other (interdisciplinary collaboration). This is where the greatest problem lies.

As productive as different perspectives on collaborative requirements engineering can be in an ideal situation, it is difficult to get those benefits in reality. The key to making these

collaborations effective is language. Terminology that belongs to one knowledge system is often misunderstood by people who work with another knowledge system. Overcoming the linguistic barriers between these systems is essential.

## Ambiguity

Ambiguities are a problem because they can lead to two or more different interpretations of the same word. They are often part of the subconscious, so requirements writers will not necessarily recognize these potential sources of misunderstandings.

There are different kinds of ambiguity. *Lexical ambiguity* refers to single expressions that may be reasonably interpreted in more than one way. One simple example is the German adjective *einfach*, which implies two different meanings. The first meaning refers to the “ease of doing something.” A requirement containing *einfach* could be taken to mean, “The sensor shall be easy to install.” The second meaning of this adjective refers to the “frequency of an activity.” So a requirement containing *einfach* could also be read as, “The sensor shall be installed only once.” The second interpretation is not the intended meaning, as often sensors should feature a bracket to reattach them at some other location. Once an ambiguity like this is detected, the author should be guided toward a clear, alternative wording.

Misinterpretation can also result from *syntactic ambiguities*. The syntactical structure of a requirement—for example, when pronouns and relative clauses can refer back to two or more other elements of the preceding sentence—might also lead to different interpretations. In addition there might be *discourse ambiguities*, which are defined as incompatibility between several requirements.

## Vagueness

In cases of vagueness, it is difficult to form any interpretation at the desired level of specificity. This is in contrast to ambiguity, where more than one interpretation is evoked.

- Use of the passive voice can prevent a reader from understanding the meaning intended by the author of a



TERMINOLOGY THAT BELONGS TO ONE KNOWLEDGE SYSTEM IS OFTEN MISUNDERSTOOD BY PEOPLE WHO WORK WITH ANOTHER KNOWLEDGE SYSTEM. OVERCOMING THE LINGUISTIC BARRIERS BETWEEN THESE SYSTEMS IS ESSENTIAL.

requirement as it lacks explicit reference to the actor. In contrast to this, the active voice clearly identifies who is performing that action. A sample requirement that reads, “Video streams of critical vehicle movements have to be stored,” fails to indicate which technical or organizational entity is responsible for data storage and where the storage will be allocated.

- Use of participles to modify nouns also leaves open the agent of action. A sample requirement that reads, “Installed sensors must measure the weight of the vehicles,” leaves open the question of who will install the sensors.
- The use of unspecific adjectives is another likely source of vagueness. For instance, “Components installed in the public road space should have an unobtrusive appearance,” leaves open the possibility of interpreting “unobtrusive” in many ways. More detailed requirements that specify a maximum size and a desired color of the components avoid the problem.

### Improving the Quality of Natural-Language Requirements

These kinds of problems can be identified by means of strict detection routines. Requirements engineers can then remedy flaws and improve quality. At the DLR (German Aerospace Center), we have taken some steps in this direction:

- An in-depth initial linguistic analysis of the requirements repository identifies a list of critical and ambiguous terms that can serve as a basis for future requirements reviews.
- A prototype terminology management system plug-in for the requirements management tool is under development, which allows for modeling domain-specific terminologies. Using this plug-in, engineers can organize relevant terms according to the semantic relations between them. The plug-in includes ambiguous words and links to the associated different readings they can have. It indicates terms to avoid because of the ambiguity they create.

Future work will be directed to modeling rules for avoiding syntactically ambiguous sentence structures. Together with the detection of lexical ambiguities, this has the potential to significantly improve the linguistic clarity and quality of requirements. Future work will be directed toward the continuous improvement of the current prototype tool. The ultimate goal will be to use the tool for linguistic checks based on both the available terminology and the syntactical rule set.

Early experience has shown that especially the first steps of linguistic analysis and modeling need to be done by experts who are highly aware of potential natural-language problems. It also is clear that correcting faults requires close cooperation between linguists and domain experts whose knowledge is required to recognize and remedy linguistic faults. Terminology and existing syntactical rule sets need to be continually updated for new application contexts. In the future, we expect that requirements engineers will eventually have a tool that will allow them to model their own semantic networks and terminology systems to the benefit of their requirements engineering work. ●

**LARS SCHNIEDER** works at the German Aerospace Center, Institute of Transportation Systems.

**SUSANNE ARNDT** works at the Technische Universität Braunschweig, Institute for Traffic Safety and Automation Engineering.