Editor: Suzanne Robertson ■ The Atlantic Systems Guild ■ suzanne@systemsguild.com

Requirements and Psychology

Chris Rupp

"Well, I thought you meant..." "No, that's not what I said..." sounds familiar to any requirements engineer. Chris Rupp decided to borrow some ideas from the discipline of psychotherapy to help pin down what people really mean. Here's a good example of how we can use well-developed ideas from other disciplines to help improve our own. —Suzanne Robertson

Fulfilling your customers' interests determines your market success, but how do you find these requirements effectively and efficiently? As simple as this question sounds, answering it in daily practice is difficult. Often, stakeholders are interviewed about their requirements or asked to write them down, but this approach rarely uncovers the real requirements that reflect a customer's true interests or needs. We need a way of getting information about the customers' core desires—conscious, unconscious, even subconscious. The hottest sellers are products that fulfill these desires.

Linguistics

Communication between people is never easy, especially when it's about requirements for complex systems or products. To help overcome communication problems, I recommend a method from the area of neurolinguistic programming (NLP), which belongs to the field of psychotherapy. (For further research into this topic, please see R. Bandler and J. Grinder, *The Structure of Magic*, Science and Behavior Books, 1990.) This conversational method tries to decode what a person means when speaking.

According to NLP, mistakes in statements can happen in two places:

- perception of facts (reality)
- linguistic representation of one of these perceptions (personal reality)

The goal of semantic language analysis is to uncover formulations borne of subjective experience and replace them with clear, objective formulations.

Since 1995, our company, Sophist GmbH, has applied language models such as the Requirements Engineering (RE)-Metamodel to areas of computer science. The result has been simple, easy-to-use rules for prose requirements in project development. Using these adaptable rules, we can systematically review requirements formulated in natural language so that errors or ambiguities never see the light of day.

The RE-Metamodel

Figure 1 shows the RE-Metamodel, summarizing the most frequently observed language phenomena linguistically. Each of these phenomena indicates linguistic defects. I will explain an example of each type of defect and how you can recognize them in your own work.

Deletion

We all use the process of deletion to reduce the world to dimensions that we can handle. In requirements engineering, however, we must know exactly what information we've lost, or we run the risk of a mis-

0740-7459/02/$17.00 © 2002 IEEE

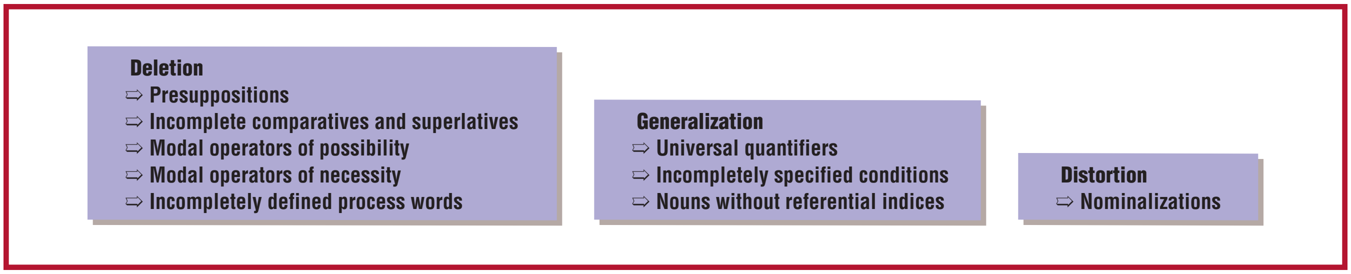
Deletion
⇨ Presuppositions
⇨ Incomplete comparatives and superlatives
⇨ Modal operators of possibility
⇨ Modal operators of necessity
⇨ Incompletely defined process words
Generalization
⇨ Universal quantifiers
⇨ Incompletely specified conditions
⇨ Nouns without referential indices
Distortion
⇨ Nominalizations

Figure 1. The RE-Metamodel, summarizing the most frequently observed linguistic defects.

understanding. One way to find deleted information is to examine a sentence's *process words*, or verbs.

To be complete, a process word usually requires an argument or a noun phrase. Consider the following: "The system shall report data loss." The process word "report" is completely defined only if the following questions are answered: Who reports what to whom? What is reported? When? How?

For each requirement, you must be sure that the process word in it is a satisfactory and clear-without-ambiguity definition of the actual process. Identifying and roughly specifying the processes that your system should carry out will guide you in asking questions that clarify meaning. If dynamics and complex processing rules play a deciding role in your system, pay special attention to process words. They can lead you to a clear definition of when and in what condition your system performs its processes.

Generalizations

In the generalization process, requirements seem to apply to the system as a whole, but in fact they usually apply to just a small piece of it. Typical indicators of generalization are *universal quantifiers*: parts of statements broadly applied to all of an occurrence's incidences.

Representatives of linguistic quantifiers include concepts such as "never," "always," "no," "every," and "all." The danger in using them is that the specified behavior does not always apply to all the referenced objects in a group or set. Consider the following: "Each signal shall be labeled with a time stamp." On the basis of the keyword "each," a question immediately arises: Are there one or more special cases in which the time stamp is not required?

You probably won't find many universal quantifiers when searching your text, which is not necessarily a good sign. Inspect sentences that don't contain explicit statements about the number of objects for which the specified reaction occurs. They often contain an implicit assumption that the specified reaction is valid for all relevant objects.

Sometimes knowing how much time and effort to spend on special cases and exceptions is hard. It all comes back to assessing the risk of not asking the questions. In cases where safety of life and limb is involved, you can't afford to miss exceptions.

Distortion

The problem of distortion appears almost exclusively in the form of *nominalization*, which occurs when a process is reformulated into an event.

A nominalization can change a statement's meaning and cause important information regarding a process to be lost. Linguistically, nominalization is a process word (verb or predicate) molded into an event word (noun or argument). Consider the following: "After a system breakdown, a restart shall automatically be initiated." The processes behind the nouns "system," "breakdown," and "restart" actually consist of a system breaking down and a system restarting. But, how is the restart being performed? Who initializes it? What ends it?

Using nominalized terms for a complex process is fine—if the process is clearly defined. This definition should not leave any margin for interpretation of the process and should clarify its progression as well as all input and output parameters. The goal is not necessarily to avoid nominalizations but to use them only if the process lying behind them is clear. Nominalizations often appear in domains with an extensive technical language, so check your field's terminology. Certainly you will identify most specialist terms as nominalizations that hide a large amount of specialized knowledge.

Applying the metamodel

Many companies have successfully used the RE-Metamodel in various project areas: Deutsche Flugsicherung, Swisscontrol, and Eurocontrol (air traffic control systems); Deutsche Post (logistic systems); Kreditwerk and Bausparkasse Schwäbisch Hall (societal systems); and Stoll GmbH (knitting machines).

In all these different areas, the RE-Metamodel helped systematize the engineering of stakeholder needs. Stakeholders used additional templates to formulate requirements, which in turn clearly defined how to finalize them (see www.sophist.de for more information). Clear preconditions on how to elicit and accurately document requirements is especially helpful in cases where they are not noted in the stakeholders' primary language. Most analysts and stakeholders love to receive precise working instructions because specifying a new system can be a complex and nonintuitive process.

You can apply the RE-Metamodel in three ways:

- *Dialogue*. The analyst immediately checks each stakeholder statement for linguistic effects and studies missing facts that appear to be important. Thus, interviews quickly get down to the nitty gritty, efficiently eliciting the required knowledge and promptly revealing knowledge gaps. This technique requires an experienced analyst. The interviewee should not notice that the analyst is systematically inspecting all statements for linguistic effects. To do this, the analyst must be experienced enough to identify and ask essential questions.
- *Inspection of written requirements*. Using linguistic rules, the analyst inspects already existing requirements for omissions, lack of clarity, and so forth according to the metamodel's rules.
- *Combination*. Of course, you can combine both approaches.

Ultimately, the analyst decides how to use the RE-Metamodel, adapting it to each particular situation on a case-by-case basis. With some concessions, the RE-Metamodel can help represent a complete, faultless set of requirements using the informal notation of natural language. Thus, the door is opened for the analyst and stakeholder to work together to review and criticize the problem's working description.

Usually, the NLP approach is imparted at a two- or three-day training course. Afterward, you can successfully apply it immediately because you don't have to learn a new way of thinking, you just have to explore your knowledge of language. You're learning a model to make that knowledge more accessible and some refinements to help you apply it to RE.

For the first application of the method after training, we recommend using it to analyze already existing documents. On the basis of existing requirements, you can test each statement for linguistic effects and scrutinize it before interviewing stakeholders. A few days of practice will integrate the RE-Metamodel into your thinking enough for you to examine and scrutinize your customers' statements in real time.

What makes a good analyst?

Knowing that perfect communications are nearly impossible relieves you of trying to achieve a perfection you never could reach. Managing the incompleteness of communication is core to mastering the RE process. The point of developing approaches such as the RE-Metamodel is to help systems analysts be more effective in doing a complex job.

Good analysts mediate between worlds. They investigate the stakeholders' needs while discovering, formulating, and inventing requirements. They then translate and transmit this information to the world of system development and informatics.

Analysts do much more than translate stakeholders' wishes into requirements, however. Excellent communicative and therapeutic skills are expected. In cases of diverging opinions between stakeholders, analysts must work out a common essence and develop acceptable solutions for all people involved. They must overcome the stakeholders' hindrances and fears; every new system carries the potential to create fear by scrutinizing previous positions and habits. Analysts should have experience in group dynamics, conflict management, and team management apart from sound IT and domain know-how.

Analysts do much more than translate stakeholders' wishes into requirements, however. Excellent communicative and therapeutic skills are expected.

Good systems analysts have a complete suitcase of elicitation techniques from various sectors. This suitcase should contain therapeutic-sector approaches (NLP, family therapy) as well as creativity techniques (brainstorming), behavioral science methods (observation techniques), and computer science approaches (use-case-driven modeling, prototyping techniques).

A good analyst's working method is never really predictable. He or she starts the conversation with a stakeholder and decides, based on the situation, which technique to use. Factors influencing that decision include

- Power and trust relationships within the team
- Whether the demanded information is lodged in the conscious, unconscious, or subconscious
- Whether the stakeholder is good at explaining knowledge (prefers to work with concrete examples or abstract concepts)
- Whether the system is concerned with dynamic–active procedures or static information

The analyst should develop a flair for using the right technique in the right situation. A mixture of many different techniques can bring revolutionary success (see www.systemsguild.com).

To be a successful requirements engineer, be prepared to always be in a learning situation, accept change, and never do things by rote. The variety of techniques, stakeholders, and domains make the job of RE an increasingly challenging and enjoyable one.

Chris Rupp is the managing director of Sophist GmbH and Sophist Technologies GmbH. Contact her at Sophist Gesellschaft für innovative Softwareentwicklung mbH, Vordere Cramergasse 11-13, 90478 Nuremberg, Germany; chris.rupp@sophist.de; www.sophist.de.