

Improving Dependability through the Application of Rigorous Theoretical Linguistics to Requirements Capture

Kimberly S. Hanks, John C. Knight
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
{ksh4q | jck}@cs.virginia.edu

The primary measure of success of a software system is the degree to which it meets the purpose for which it was intended.

--Nuseibeh and Easterbrook

The obvious implication of the above is that it is necessary to *know* the purpose for which a given system is intended if the system is to be dependable. However, many have recognized that this problem is formidable in and of itself (Nuseibeh, Easterbrook, Potts, Goguen, Jackson).

It is often the case that the requirements are incomplete, inconsistent, or just plain incorrect. A software system meeting such requirements does *not* meet the purpose for which it was intended. This implicates the capture process in many system failures.

A fundamental issue in doing requirements is achieving the conversion from an informal to a formal representation of information. *Every* problem to be solved begins, by nature, as an informal cognitive concept in the mind of a client. *Every* implementation of a solution, by virtue of the well-definedness of programming languages and the underlying switches they manipulate, is by definition a formal specification of machine behavior. A boundary is thus defined between the non-implementable informal semantic domain and the refined formal one. This boundary *has to be* traversed in producing *any* piece of software.

However, natural language and formal language have fundamentally different characters. The semantics of formal languages are well understood because of their precise, mathematical nature. The semantics of natural languages will not be completely understood in our lifetimes, even by linguists. This notwithstanding, linguistics has assembled a large body of accepted belief on much of the semantic structure present in natural language. There are several elements of natural language semantic structure which differ fundamentally from their corresponding formal elements. It is these elements which are arguably the most difficult to convert appropriately and

usefully, and they are the cause of many of the issues in capturing valid requirements. The validity of requirements takes on heightened importance in systems for which the consequences of failure are high, so we would like to better understand the fundamental differences between natural and formal representations of information in order to more effectively traverse this boundary in capturing requirements for dependable systems.

Among the structural elements in natural language with a differing formal analog is the notion of a *conceptual category*, formalized as a *type*. Formal types are discrete sets with flat internal structure. Membership is binary, and all members have equal status as representatives of the type. Conceptual categories are fuzzy sets with hierarchical internal structure. Members may be judged as such inconsistently by natural language users and some members are better representatives of a category than others. These fundamental differences in the nature of classification are not accounted for within a direct mapping of category to type, and thus a wholesale formal flattening of a client's notion of a real world category has hidden perils, such as the loss of critical information. A far more complex mapping procedure is required.

A more considered conversion of a category to a type might include a probe of the semantics the client associates with the category and extraction of several judgements of membership, as well as a description of a prototypical member. Such information can enable the construction of a type which better encapsulates the most central and pervasive features of a given conceptual category. Such information can also illuminate cases of overloaded terminology for which experts have uncommon semantics. More importantly, such information might suggest an entirely different mechanism for conversion.

Our work in progress currently consists of identifying the critical elements of natural language structure and their formal analogs, and analyzing the structural differences between them. From the resulting analysis, we intend to develop strategies for approaching these difficult conversions and to devise a methodology for the

application of these strategies with sound theoretical foundations. We expect that the application of rigorous theoretical linguistics will provide insight and reduce significant sources of error in producing valid requirements for dependable systems.

References

- [1]Goguen, J., "Formality and informality in requirements engineering", *Proceedings, 1996 IEEE International Conference on Requirements Engineering*, pp. 102-108.
- [2]Jackson, M., "The meaning of requirements", *Annals of Software Engineering*, 3: pp. 5-21.
- [3]Nuseibeh, B., and Easterbrook, S., "Requirements engineering: a roadmap", in A. Finkelstein, editor, *The Future of Software Engineering*, Special Volume published in conjunction with ICSE 2000.
- [4]Potts, C., "Naturalistic inquiry and requirements engineering: reconciling their theoretical foundations", *Proceedings, 3rd IEEE International Symposium on Requirements Engineering*, pp. 118-127.