**Erik Kamsties**: **Surfacing Ambiguity in Natural Language Requirements**

**Promotion:** Universität Kaiserslautern
**Erstgutachter:** Prof. Dr. H. Dieter Rombach (Universität Kaiserslautern)
**Zweitgutachter:** Prof. Dr. D. M. Berry (University of Waterloo, Kanada)
**Datum der Prüfung:** 16.03.2001
**Veröffentlichung:** Stuttgart: Fraunhofer IRB Verlag, 2001 (PhD Theses in Experimental Software Engineering 5). Zugl.: Kaiserslautern, Univ., Diss., 2001; ISBN 3-8167-5903-3

**Kurzfassung:**

In industrial requirements engineering (RE), natural language is the most frequently used representation to state the requirements that are to be met by information technology products or services. Natural language is universal, flexible, and wide-spread, but unfortunately also inherently ambiguous. Even worse, often neither customers nor software developers recognize an ambiguity, and each derives an interpretation that differs from that of others without noticing this difference. Consequently, software developers design and implement a system that does not behave as intended by the customers. Semi-formal and formal representation techniques, such as UML or SCR, have been proposed to overcome the weaknesses of natural language. However, they only shift the problem; an ambiguity simply becomes an unambiguously wrong specification statement, which must be detected by the customers in reviews or simulations of the specification.

This thesis addresses the problem of detecting ambiguities in natural language requirements. We distinguish between linguistic and RE-specific ambiguities. Linguistic ambiguities are the commonly known ambiguities of natural language, such as ambiguous pronoun references. RE-specific ambiguities arise from the RE context, which comprises the application, system, and development domain. The requirement "The system injects coolant to [...] when the pressure falls below a 'low' threshold" is ambiguous because of the system domain, i.e., it is not clear whether injecting coolant is a discrete action or continuous activity. An experiment was performed to understand the problems that the different types of ambiguity cause during the formalization of informal requirements.

This thesis presents the ADTD approach, which allows developing techniques for detecting ambiguities in natural language requirements on the basis of existing and industrially proven techniques, namely checklists, scenario-based reading, and agendas. A technique for detecting ambiguities must be tailored to a particular RE context to be effective, because RE-specific ambiguities depend heavily on the context in which an RE process takes place. For this purpose, the ADTD approach provides heuristics to investigate metamodels, which capture information about an RE context. These heuristics help identify the RE-specific types of ambiguities that are typical for the particular RE context.

A set of ambiguity detection techniques for the domain of embedded systems was developed using the ADTD approach to show its feasibility. To quantify the benefits of the ADTD approach, the resulting techniques were empirically validated in five experiments. The goals were (1) to validate that the investigation of metamodels pays off in terms of higher efficiency of the resulting technique and (2) to show that our ambiguity detection techniques are superior to existing ones.

From a practitioner's perspective, a set of validated techniques for detecting ambiguities is provided, which allow improving current RE processes of different maturity. The ADTD approach is available for developing additional techniques if the provided ones do not fit the situation at hand. From a researcher's perspective, new insights are provided on the problems that defects, and ambiguities in particular, cause when semi-formal or formal specifications are developed from an informal statement of requirements. Also, the thesis provides a comprehensive understanding of the various facets of ambiguity in RE. This understanding clarifies the currently ambiguous use of the term in the literature.