München, den 4.9.2007

Diploma / Master Thesis

Test Generation from Natural Language like Requirement Specifications

MQM
MODEL-BASED SYSTEMS & QUALITATIVE REASONING

within a joint research project with AUDI.

Background and Initial Situation

The increasing importance and complexity of software and electronics in the automotive sector enforce the manufacturer to take novel and intensified actions regarding quality assurance. Here requirement-based testing, where the individual requirements of a product specification are tested against the real product (or its prototype), plays a key role.

In order to drive on the development of requirement-based methods the Audi AG provided us with a requirement document as a case study within a joint research project. The document captures the requirements of a real product, the Adaptive Cruise Control (ACC), as it is used in modern vehicles (see figure 1).

The system regulates the vehicles velocity depending on the distance to the car in front and the set speed given by the driver. The distance is measured via Long-Range-Radar (LRR).

Figure 1: The Adaptive Cruise Control (ACC)

In a first step a natural like requirement specification formalism FRL has been developed and the ACC requirement rules have been transformed from the initial natural language to the FRL syntax via sentence templates (see figure 2). The objective for developing FRL was on the one side to find a formal, computer processible representation. On the other side, the formalism has to be near to the natural language in order to, for instance, offer the engineer a intuitive way for writing and reading requirements.

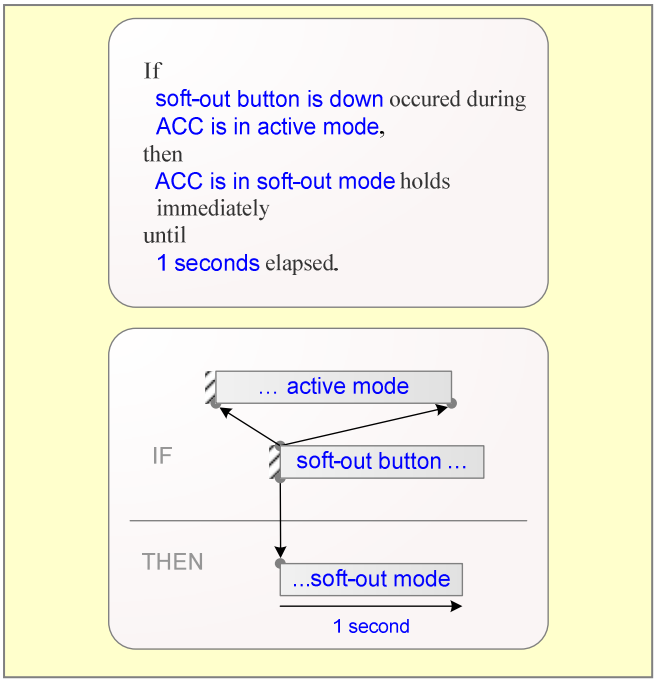

Figure 2: An ACC requirement formulated with a template (top) and in FRL (bottom).

The objective of the research project is to develop and evaluate fault-model-based test generation methods. Note that test generation (TG) aims at finding proper test cases but does not address the issue of (automatic) test execution (like HIL testing). In fault-model-based TG the model of the correct system is compared with a fault-model representing the system with a specific fault. Each input to the

system leading in the ok-model to an observable reaction that is different (disjunct) from the one in the fault-model, represents a reasonable test input (see figure 3). For this a set of proper fault types and the corresponding fault-model generators have been already defined.

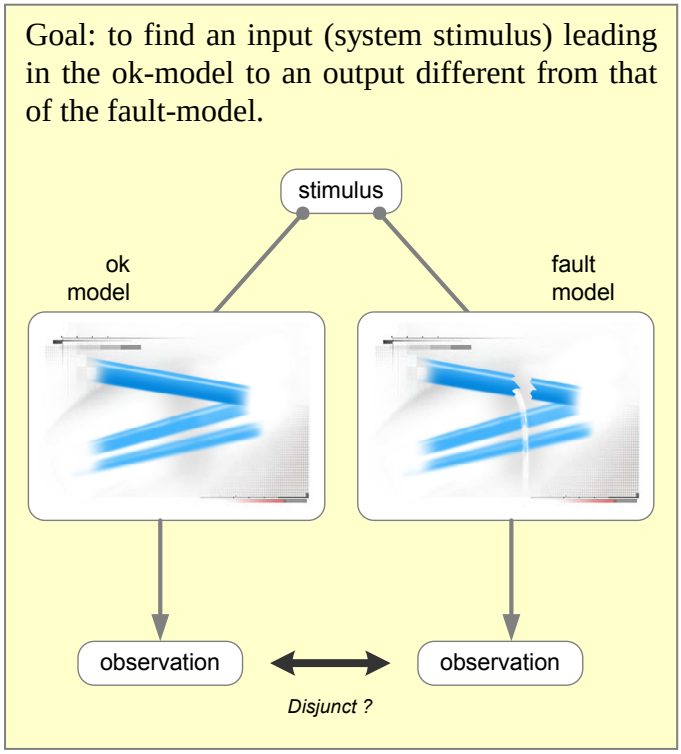

Figure 3: The Task of fault-model-based test generation

Objective and Task Definition

- Designing an algorithm for fault-model-based test generation (on the base of dependency graphs)
- Implementing the algorithm in C#
- Evaluating the algorithm in comparison with an existing reference implementation of another algorithm.

High value is set on a clear documentation of the mathematical descriptions, the design (UML) and the implementation.

Demanded Skills

- Programming skills in C# or Java, as well as
- Not being shy to use formal methods and concepts (finite automata, logic)

Contact Person

Michael Esser
Michael.Esser@in.tum.de
Raum 00.13.53

Head of Project

Prof. Peter Struss
Model-Based Systems & Qualitative Reasoning Group
Chair for Image Understanding and Knowledge-Based Systems
Technische Universität München