Institut für Informatik und Praktische Mathematik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Formal Methods and the Development of Dependable Systems

Jan Peleska

e-mail: jp@informatik.uni-bremen.de

Bericht Nr. 9612
Dezember 1996

Dieser Bericht enthält die Habilitationsschrift des Verfassers.

Referent: Prof. Dr. Willem-Paul de Roever
Koreferenten: Prof. Dr. Manfred Broy, Prof. Dr. Mathai Joseph, Prof. Dr. Hans Langmaack, Dr. Tech. Anders P. Ravn

Preface

This *Habilitationsschrift*[1] focuses on methods for the development of dependable software-based systems. It summarises, discusses and extends my publications cited in the bibliography, which reflect the efforts and experiences gained in this field during the last decade, working as a software engineer, project leader and manager at *Philips GmbH*, *Deutsche System-Technik GmbH* and as a consultant for various other companies.

The applications to be discussed as examples will mostly be chosen from the field of non-military information and control systems. Such a specialisation appears to be necessary, because the type of application influences the objectives to be met by dependability mechanisms. For example, in dependable and secure military applications (see [50]), the aspect of confidentiality (mandatory access control, covert channels etc.) plays a much more important rôle than for a railway control system. As a consequence, different application areas lead to different approaches with respect to system design, development techniques and underlying specification and design methods.

The methodological framework used is based on *Formal Methods*. I would like to point out that though my university education was rather theoretical (see [72]), the motivation to use theory for building software was purely motivated by the fact that the informal heuristics applied for constructive or analytic software quality assurance are completely insufficient when applied to systems where correctness of software really matters. The key ideas described in the chapters to follow might be summarised by three statements reflecting my conviction how the software crisis should be tackled in the future:

- The complexity of today's applications can only be managed by applying *a combination of methods*, each of them specialised to support specific development steps in an optimised way during the system development process.
- The application of formal methods should be supported by *development standards*, i. e., explanations or "recipes" showing how to apply the methods in the most efficient way to a specific type of development task. Indeed, tool support and development standards may be regarded as essential for the success of formal methods in an industrial context.
- The application of formal methods for the development of dependable systems will only become cost-effective if the degree of *re-usability* is increased by means of re-usable (*generic*) specifications, re-usable proofs, code and even re-usable development processes.

As a consequence, the objective of this *Habilitationsschrift* is not to introduce new specification languages, proof techniques *et cetera*, but to show how new combinations of existing methods can be applied more efficiently to solve problems in the field of dependable systems.

[1] *Postdoctoral thesis*, required in Germany for the qualification of a university lecturer; since the English language does not provide a proper counterpart for this term, we keep the German expression instead of using an English circumscription.

The "architecture" of this work is depicted in Figure 0.1. In the introductory Chapter 1 some basic definitions used in the context of dependability are presented and discussed, since the "invasion" of software into this field has forced specialists to re-interpret important terms previously used only in the context of mechanical and electrical engineering. Moreover, an example is given which reflects the complexity to be encountered when trying to combine different dependability mechanisms (e. g. a fault-tolerant protocol together with a security mechanism) in order to "add up" the benefits of these mechanisms in one system. In Chapter 2 I will describe an approach for the systematic formal development of dependable systems. This approach offers a design technique systematically dealing with situations where not just one threat to system dependability but a collection of threats has to be taken into account. At the same time it is embedded in a general development framework that has proved to be practical for large projects and is accepted by (at least the more enlightened species of) today's software engineers in industry. This embedding associates a theoretic foundation with an informal development standard, using selected formal methods to produce the contents of the documents pre-defined in the standard. The example of Chapter 1 is used to demonstrate how the approach can be put into practice. This work has been mainly motivated by my cooperation with ELPRO LET GmbH and the University of Oldenburg in the field of distributed railway interlocking systems.

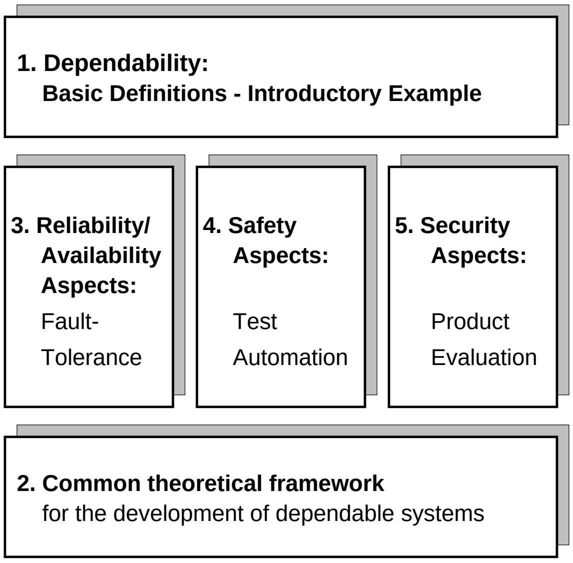

Figure 0.1: "Architecture" of the *Habilitationsschrift*.

Chapters 3 to 5 focus on specific aspects and applications in the field of dependable systems: reliability/availability, safety and security. In Chapter 3 the formal specification and verification of a dual computer system is described, where the reliability of the system is ensured by means of a fault-tolerance mechanism. The underlying design and verification concepts have been worked out during my time with Philips, where I was confronted with the development of a fault-tolerant database server. In Chapter 4 a concept for the automated test of safety-critical embedded systems will be introduced. The concept and the corresponding tool support have been developed for and applied in the field of railway interlocking systems for small private railways or tramways. For these systems, the degree of complexity is considerably lower than in the case of large-scale railway networks. Furthermore, safety aspects play the dominant rôle, while reliability and availability are subordinate aspects. Therefore we can achieve a much higher degree of automation for the test and verification process than for the case of dependable systems in general. In Chapter 5 security aspects of dependable systems are investigated. Here we concentrate on the evaluation of commercially available IT security products. The objective of this evaluation is to certify in a trustworthy way that the system is capable to protect its users against the specified types of security attacks. This work has been initiated while I was manager of a department at DST that was specialised on developing dependable systems. Associated with the department was an evaluation laboratory which was accredited at the *Bundesamt für Sicherheit in der Informationstechnik BSI*, the German authority for the certification of IT security products.

You may have noticed that I did not use the Preface to give a "sales talk" about how much we are threatened by software of insufficient quality and how only formal methods can deliver us from this evil. Firstly, I assume that this *Habilitationsschrift* will only be read by experts anyway, and either you will already be a member of the formal methods community or you will be too much of an expert to be convinced by my arguments. Secondly, I have the impression that the common understanding about the *status quo* of the software crisis, what formal methods can do about it and where the limitations of formal methods lie has improved very quickly during the last five years. This even holds for hard-boiled software practitioners to be encountered in software industry. If you still feel like reading some motivating thoughts about all this, I am sure the articles of Gibbs [31], Heisel and Weber-Wulff [38] and the Hamer-Hörcher-Peleska formal methods primer [79] will satisfy your needs.