Conformance Testing for OSEK/VDX Operating System based on Model Checking

CHEN Jiang (0910036)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 8, 2011

Keywords: OSEK OS; design model of OSEK OS; conformance requirement; test purpose; test model; model checking.

The OSEK/VDX operating system (further called OSEK OS) is a real-time operating system widely used in the automotive applications. The OSEK OS specification was specified by the OSEK/VDX group, which is a joint project of the automotive companies in Europe. Due to the standardized system services as specified in the specification, high portability and reusability of automotive application software can be achieved. However, the specification itself does not prescribe any particular implementation language of the system services and leaves a certain amount of flexibility. As a result, it may raise the risk that an implementation of OSEK OS does not comply with its specification. Therefore, conformance testing is needed to check whether an implementation is correct with respect to the OSEK OS specification. This research aims at proposing an original approach to conformance testing for OSEK OS.

To support checking the conformance of OSEK OS implementations to the OSEK OS specification, the OSEK/VDX group has founded a project named MODISTARC. In this project, a standard conformance testing methodology has been developed. It serves as the basis for the development of the test specifications and test tools. However, we found that based on this methodology, it is difficult to assure the correctness of the

test purposes with respect to the specification without verifying them. In addition, since test cases are designed manually, it takes much effort to make test cases achieve the corresponding test purposes. These shortages may impact test quality. Hence, it is desirable to develop a new method to promise the correctness of test purposes as well as automatic generation of test cases.

In recent years, we have developed a formal model of OSEK OS in PROMELA called as the design model of OSEK OS. It models functionality of system services as specified in the OSEK specification and can manipulate such as task state, resource state or other OS primitive objects by calling system services. In other related research, huge amounts of experiments using model checker SPIN to verify the design model of OSEK OS have been conducted. We have got sufficient confidence of its correctness with respect to the OSEK OS specification. Therefore, we consider taking advantage of the existence of this model and use it as a test oracle to automatic generate test cases by using model checking technique.

In the proposed approach, we concentrate mainly on developing a method to automatic generation of test cases by model checking the test model with the design model of OSEK OS. The whole process of test generation includes five steps as follows:

1. Extract the conformance requirements from the OSEK OS specification. In general, before testing a system implementation, the requirements of testing must be clarified. From conformance testing perspective, test requirements consists of two kinds of requirements, namely static conformance requirements and dynamic conformance requirements. In this research, we define the conformance requirements in the context of the OSEK OS specification. Moreover, we introduce the concept of test purpose and make it relate to the dynamic conformance requirements.

2. Formalize the static conformance requirements and test purposes. To remove ambiguous and give a precise description of the extracted conformance requirements, we use formal specification language to for-

malized them. The formalism is called as the formal test specification. Another intended purpose of the formal test specification is to serve as a basis for constructing the test model.

3. Construct test model based on the formal test specification. To achieve automatic translation from the formal test specification to the test model, a translation algorithm is proposed.

4. Make correction of the test purposes. Since the OSEK OS specification does not state so-called conformance requirements explicitly, we extract the conformance requirements based on our knowledge of the specification. As a result, it is possible that test purposes are not correct with respect to the design model of OSEK OS. By model checking the test model with the design model of OSEK OS, we can get feedbacks from the checking result. If a violation is detected, SPIN will stop searching and report the violation place in the test model. Then we can make correction of the corresponding test purposes according to the checking result.

5. Derive test cases from the witnesses. If the test purposes are correct with respect to the design model of OSEK OS, by model checking the test model with the design model of OSEK OS, no violation will be detected, instead, the exhaustive state space searching will be conducted and the witnesses will be generated. Then we can derive the test cases from the witnesses by using our test case generation tool.

To evaluate the proposed approach, test purposes are developed from the system services in the scope of task management and resource management in the OSEK OS specification. Furthermore, we compare the test cases generated from the test model with the test cases defined in the MODISTARC.

Based on the proposed approach, test model can be constructed in a systematic way. We can assure the correctness of the test purposes with respect to the design model of OSEK OS. Moreover, the generated test cases from test model can achieve the corresponding test purposes. Therefore, test quality can be significantly enhanced.