Formal Verification and Validation of Smart Cards

Stefan Kriebel, Giesecke & Devrient
Stefan.kriebel@de.gi-de.com

SOCRATES is a test tool for chip card tests. It is suitable to produce automatic tests for all kind of chip cards. SOCRATES is an open tool which supports a variety of different test strategies. This variety goes from simple action word testing to formal testing.
The components of SOCRATES are decision tables, test scripts, simulators if necessary and the kernel of the tool itself.

Decision Tables: For each command there is a so called decision table in form of an Excel sheet. The decision table describes the different cases of a command. A case is named by a letter form A to Z and consists of a set of variable settings and a expected return value. For the variable settings a PROLOG like expression is used. With these expressions the command parameters are defined and pre- and post-conditions can be set. If a command case is activated during runtime, the pre-conditions are checked and the post-conditions are written after successful command execution. The whole system of conditions is stored in a database which determines the actual state of the chip card. For further details see below.
Test Scripts: The test script contains a sequence of commands also called action words. A command is defined by a command name and a case letter as listed in the decision table. Parameters set in the decision table can be overwritten in the script. The script allows also control structures like for-loops, if-else statements and includes of other scripts.
Simulators: Beside the simulation provided by the pre- and post-conditions in the decision table, additional Simulators can be attached to SOCRATES. For each command a C++ code is generated where the simulator routines can be placed. This may be used for complex data modelling e.g. cryptographic computations.
SOCRATES Core: The core reads the test script, finds the command in the decision table, evaluates the conditions in the table and computes the actual command parameters. If necessary a simulator is called to calculate command data. The command is sent to the chip card and the response is received. The response is checked against the expected return value and the expected data. If the check is successful the post-conditions are set.

Socrates can be used in a variety of ways. Two strategies are briefly explained, formal testing and script based testing.
Formal testing in the sense used here means to map requirements given in the specification of the product to a formal description in the decision table. In the decision table a PROLOG like expression, a predicate, can be used to define the requirements. These predicates, i.e. the transformed requirements, create the previously designed formal model when used as pre- and post-conditions. Requirement 1, for example, is described with a pre-condition that the variable „df_exist“ shall contain the current file identifier FID provided by the command SelectFile. The variable „df_exist“ is set as post condition when the file is created. Another requirement 3 denotes, that P1 has a good case value of 0. Requirement 2 is implemented by setting the variable „cur_df“ to the actual file-Id. In the matrix of the decision table the value "1" means the condition has to be fulfilled, the value "0" means the conditions must not be fulfilled and the value "*" means the condition is ignored.
The test coverage can be proved by checking that for each pre-condition at least a "1" and a "0" occurs in the table. If all cases of all commands are run, all captured requirements are covered. In an enhanced version of SOCRATES a command tracking feature can be used where the command sequence is automatically generated out of the cross linking of pre and post conditions. If e.g. in the upper example a good case is issued and no file is created, SOCRATES searches for a command which sets the post-condition (df_exist FID) in this case a Create File command.
Experience with this method showed that for complex requirements the decision tables may grow extremly large. In this case they are difficult to understand and to maintain because the cross link of pre and post conditions may not be easy to trace.