7 Application of MTCC Models

This chapter examines the use of the Application Test Model for test construction. In order to facilitate the construction of tests by domain experts, the MTCC editor is used to allow the representation and configuration of the Application Test Model in a graphical user interface. MTCC supports the reuse of existing Test Configurations for multiple testees in a system family based on the transfer of Test Configuration instances in order to execute an abstract Test Configuration. To test, MTCC uses code generation to generate a concrete test script.

Section 7.1 examines the representation of the dynamic and structural parts of the Application Test Model in the MTCC editor. The editor represents tests as sequences of configurable Test Steps. Domain experts create tests by selecting and configuring the Test Step instances that represent the actions that are intended to exercise the testee. Test Step instance selection and configuration are done in the editor. The editor is a graphical user interface that represents specific features and subtrees of features by standard elements of graphical user interfaces like lists of values, test input fields or check boxes.

One purpose of the editor is the configuration of the Test Step instances defined in the Application Test Model and thus the removal of all variable elements of the feature model. While MTCC does not require or implement all specialization steps defined for feature models with cardinalities [CHE05c], it is important that specializations applied to Test Step instances are reversible to allow for the correction of mistakes. In order to support the reversal of specialization steps, MTCC introduces configuration nodes that support the explicit representation and manipulation of selection steps in MTCC feature models. Section 7.2 discusses the realization of the configuration activities for feature models in MTCC.

MTCC considers testees in the context of software system families and aims to support the reuse of Test Configuration models created based on the Application Test Model of one testee for other testees. Section 7.3 discusses the requirements for such reuse in MTCC and the techniques used to support the transfer of tests systems.

In order for tests to be transferred from one system to another, it is necessary that the target system supports the execution of all Test Steps used in the test in the same order in which they are used in the test.

Additionally, the Test Steps for the initial system and the target system must be compatible in the sense that the feature model of each Test Step instance in the
target systems supports the configuration that was applied to the feature model of
the corresponding Test Step instance for the initial system.

If more than one sequence of Test Step instances for the target system is com-
patible with the Test Step instances in the initial system, an automatic decision on
the sequence that best represents the semantics of the original test is not possible.
It is also not possible for MTCC to decide whether a test that can be technically
transfered by MTCC really represents the intention of the test on the target system.
For these reasons, test reuse in MTCC is a semiautomatic process.

The Test Configurations created in the test modeling process do not contain
information about the technical details required for test execution. In order to ex-
ercise a system based on the tests, these models must either be interpreted by a
runtime system or must serve as the basis for test code generation. Section 7.4 dis-
cusses the approach to test code generation used in MTCC. MTCC uses a template-
based code generator that generates test cases for xUnit test runners. MTCC test
cases depend on a library that contains implementations for all Test Step instances
of a given system under test.

7.1 Representation of Test Steps in the Editor

The purpose of the MTCC editor is to support domain experts’ formal modeling
or programing skills in the specification of tests.

The test models created by the domain experts have to be formal in the sense
that they support the generation of scripts without the need for further refinement
or rework by domain engineers. In order to meet this requirement, the editor needs
to fulfill the following conditions:

- The editor has to be able to represent the information contained in the Application Test Model in a form that allows a domain expert to express a testing intent in a MTCC test. The possible configurations of the MTCC editor have to be represented by GUI elements in an understandable way.

- The editor must be able to save tests in a format suitable of code generation.

- The editor has to be able to translate the interactions of a user with the GUI into specialization steps that configure the feature model of a Test Step instance. The specialization steps have to be expressed in a way that allows for changes when the configuration of a Test Step instance is changed in the editor.