A major cost factor in the software developing process is the *functional testing* of the software. During functional testing, the system under test (SUT) is executed with input values that are chosen by a test designer. The aim of functional testing is to check if the SUT behaves according to the requirement specifications of the software.

While the execution of tests is already highly automated the developing of test cases is still done manually in most cases. One approach to automate the process of test case developing is the so called *model-based testing*. In model-based testing the test cases are automatically derived from an abstract formal model of the system under test.

The problem with that approach is that the models are described using a formal languages such as *Timed Automata* or the *B-Method* and that the use of such formal methods is not well accepted in the industry. In fact, most of the requirement specification are written in natural language. Therefore the informal requirements have to be manually transformed into the formal representation by the test designer.

The aim of this thesis is to narrow the gap between the requirements written in natural language and model-based testing. To do so a controlled natural language is presented. The language is a restricted version of English and orients itself on existing requirement specifications. The restrictions are carried out in a way such that the language is unambiguous and can be automatically translated into a formal model.

In the next step test cases are automatically generated from the formal model by using methods from the area of AI planning. The developed technique is able to handle requirement specifications that describe non-deterministic timing behavior. It is shown, that the generated test cases fulfill the MC/DC coverage criterion. Finally, case studies show, that the chosen method is able to handle requirement specifications from industrial practice.