An introduction to Tangram[1,2]

Edited by Michiel van Osch

The Tangram project aims at a significant reduction of lead time and cost in the integration and test phase of complex high-tech products. At the same time the product quality should be maintained or improved. In this paper we give a brief overview of the Tangram project.

Introduction

The performance demands on high-tech products keep on growing; they should be faster, more accurate, their uptime should be increased, etc. The business demands on these products keep on growing as well; the time to get such products to the market is getting shorter and the same goes for the period in which the return on investment can be obtained. So while engineers have to do their utmost to deliver technology that sometimes has not been invented yet, the market dictates them to do it faster, cheaper and better. This challenge is never more present than when system parts from different projects and from different disciplines (optics, mechanics, electronics and software) have to be integrated and tested. The combination of this 'faster, cheaper and better' issue and integrating multi-disciplinary state of the art technology, gives ample reason to want a breakthrough. It gives ample reason to want TANGRAM.

Project Organization

Tangram has teamed up the expertise and competence required to establish a breakthrough. Three universities: Delft University of Technology; Eindhoven University of Technology and Radboud University Nijmegen are involved. The institutes ESI and TNO-TPD are involved. The industrial partners involved are Science & Technology and finally ASML as carrying industrial partner. The project is partly subsidized by the Dutch Ministry of Economic Affairs.

TANGRAM foresees research and development along four Lines of Attention (LoA) that will be constantly challenged by a real life industrial case: a wafer scanner at ASML (See Figure 1).

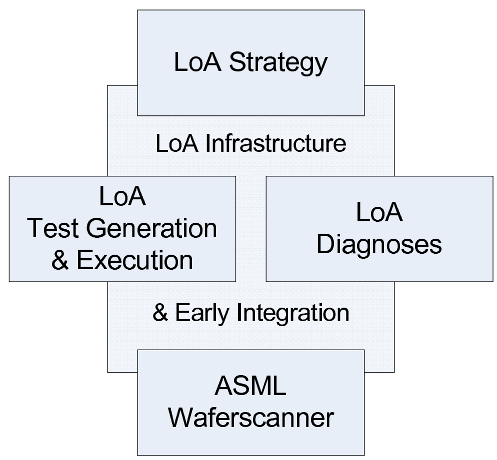

Figure 1: TANGRAM Organization

- The LoA with focus on strategy develops models of integration- and test processes featuring cost, parts to integrate, combinations to test, time to spend and product quality to achieve, as well as methods for test selection.

[1]This work has been carried out as part of the Tangram project under the responsibility of the Embedded Systems Institute. This project is partially sponsored by the Netherlands Ministry of Economic Affairs under grant TSIT 2026.

[2]Most of the content of this article is already published on the TANGRAM project site at `http://www.embeddedsystems.nl`.

- The LoA that concentrates on test generation & execution delivers model-based methods and tools for generating, executing and evaluating test cases as well as model-based simulations for parts that are not yet available.
- The LoA that aims at model-based diagnosis delivers methods and tools that are capable of diagnosing the internals of a system by monitoring its inputs and outputs.
- The infrastructure and early integration LoA delivers an environment that is capable of interconnecting the models and tools that result from the other LoA's, and develops methods and tools for early integration and test with models.

LoA Strategy

The goal of this LoA is to define the optimal test strategy for a certain product. An optimal test strategy in terms of total test duration and/or final product quality and cost is desired. The optimal test strategy for a product is influenced by the objectives of the test phase and the constraints that follow out of that. For instance a test strategy for a time-to-market driven environment (like ASML) is different from a test strategy for a quality driven environment (like aircraft industry). A good test strategy is therefore product dependent.

For complex systems an infinite number of test cases can be derived. Executing all possible test cases between the moment that a (sub-)system is ready and the system is released is therefore impossible. Selecting the optimal set of test cases is therefore a relevant question. This LoA investigates the possibility to solve this selection problem with test selection algorithms.

The third point of interest is the total duration of the test period and the resources required to do so. With total duration we mean the time it takes to execute all test cases successfully, so including fixing problems. Adding additional test resources is the common thing to do. This results in detecting more problems, which seems a good thing. But if your real bottleneck is in the fixing of problems, then adding test resources is not the best thing to do. So this means that the initial quality of the system, the available resources and the test cases to be executed determine the architecture of your 'Test Factory'. Different configurations of 'Test Factories' are modelled to investigate these effects and others on the end result, total test duration and end quality.

This magazine contains an article by Boumen et. al., in which they describe how to optimize test sequences such that it takes less cost or time and demonstrate this on an ASML TWINSCAN lithographic machine.

The current approach is to define the determination of the best test strategy into 3 phases: strategy selection, test selection and test scheduling. Optimal techniques for each phase are researched and developed. Additionally a simulator of the Test Factory has been developed to simulate the effect of the different developed test strategies on the total test duration and product quality.

LoA Test Generation and Execution

The objective of Line of Attention 3 is to improve the efficiency and effectiveness of the testing process by developing testing methodology, techniques, and tools using a model based approach.

In model based testing a model of the system under test (SUT) is developed. Models can be formal, such as those written in languages as Chi, Lotos, or Promela, or in semi-formal languages, such as state diagrams or UML models. A model is the basis for the automatic generation of test cases using a test derivation algorithm, and test results are automatically analyzed and evaluated with respect to the model. Moreover, a model can be used to simulate a part of the system under test during integration testing, if such a part is not yet available.

Starting points for Line of Attention 3 are models based on transition systems, the so-called ioco-test theory, and the prototype test tool TorX [3].

Via a case study driven approach we will work on extensions of these incorporating real-time testing, testing of complex data structures, testing of hybrid systems, compositional and integration testing, and testing of multi-disciplinary, non-software aspects.

There are several benefits of model based testing. First, a model can serve as a precise and unambiguous basis for testing, thus allowing formal validation of tests. Second, models make it possible to automatically derive test cases and evaluate test results, thus considerably reducing the manual effort of testing. In particular in case of modifications in the system, a small adaptation in the model is sufficient to generate a complete new set of test cases. Although making models requires some effort, it is expected that this effort is more than compensated by the advantages of faster, more efficient, and higher quality testing.

The challenge of this Line of Attention is to extend the existing state of the art in model based testing in such a way that, on the one hand, there is a solid and well-founded theoretical basis, and on the other hand it leads to high applicability for testing the ASML systems.

This magazine contains an article by Braspenning et. al., about a case-study on automatic model-based testing with TorX using χ as specification language [2, 3]. As a result, they found interface discrepancies between the laser unit (3rd party) and controller of a lithography machine.

LoA Diagnosis

Throughout the design, integration, and operational phase, systems are plagued by faults. Finding the root cause of system malfunction typically consumes many resources that could be spent much more efficiently. This fault diagnosis process becomes even more problematic as system become more complex. Model-Based Diagnosis (MBD) is a computerized technique that considerably increases the efficiency and accuracy of fault diagnosis.

Current MBD techniques, however, are still not adequate to handle, complex, multidisciplinary systems as found in ASML. Given an adequate MBD technique, in turn, a subsequent problem is model specification, which is a labor-intensive and error-prone process. In this line-of-attention, we aim:

- to extend current MBD technology by including features such as state, time, and probabilities in order to provide the modelling capabilities required;
- to develop a technique to (semi-) automatically derive/integrate (partial) fault diagnostic system models from/within existing design specifications.

The MBD approach is based on decomposing the diagnostic system (software) in two major components:

- the system-specific reference model, describing normative and fault behaviors, and
- a generic, diagnostic inference engine that executes the search process (the 'diagnosis algorithm'), guided by comparing actual system measurements with predictions from the reference model.

This diagnostic algorithm includes both exclusion and deduction, reasoning probabilistically over time. Consequently, development of diagnosis software reduces to reference model specification, which acts as a source code from which the diagnostic (embedded) software is automatically generated.

Aimed to provide proof-of-concept, in this line-of-attention we conduct case studies where we develop diagnostic models of relevant subsystems, apply them to realistic test data, and evaluate their diagnostic performance by comparing their diagnostic output with the diagnosis found in practice. Based on this feedback, we iteratively refine the diagnostic models and algorithms in order to determine a good trade-off between diagnosis effort (manual and computational) and diagnostic performance. The research is conducted by Delft University of Technology in cooperation with Science & Technology BV.

This magazine contains an article by Pietersma et. al., which describes the model-based diagnosis methodology as a solution for the fault diagnosis of an integrated system by inferring the health of a system from a compositional system model and real-world measurements

LoA Infrastructure and Early Integration

Infrastructure

The modelling, simulation, testing, and diagnoses techniques developed by the other LoA's need to be integrated in the ASML test and verification methodology and tools. It will be investigated how these modeling and simulation techniques can be integrated in the ASML test and verification methodology. For instance, in case the simulation models do not reflect the reality correctly (anymore), the models should be easily maintainable. Furthermore, we also want to integrate other existing test and integration techniques, and tools, into the ASML methodology.

Therefore, the main objective is the conception of an integrated simulation and test environment that has the following features:

- support for real implementations (software, hardware) as well as simulation models;
- support for component integration;
- support for batch mode testing (e.g. for regression testing);
- support for automated test execution and pass/fail verdict;
- support for (remote) model based diagnostics, using same interfacing for models and real implementations;
- support for hybrid (discrete event, continuous time) models;
- support for real-time and simulation time execution.

Early Integration

In current practice testing is mainly performed after completion of the product development and prior to shipment. This implies that testing directly influences the shipment date. To test multi-disciplinary components (e.g. combining software with electronics, mechanics or optics), all components need to be available. For some (mechanical or optical) components this results in a significant investment to have the actual components available. When time-to-market concerns limit the amount of testing time, the rigor of testing is reduced. Consequently, the risk increases that certain malfunctions are not found prior to shipment. The resulting reduction in availability directly impacts market share. Given the above (three fold) problem statement, this results into the following observations:

Component dependencies (software, hardware) and availability of those components directly limits the test scheduling. This directly determines the time to market and predictability of shipment date. Components abstracting hardware cannot be tested without the actual underlying hardware. This directly results in the need to use expensive resources up to complete production quality systems. Currently, testing is mainly manual and the implementation, documentation, and evaluation of test procedures influences the product quality. The time to market pressure dictates the amount and rigor (coverage) of tests.

To address the above problems, the usage of modelling and simulation techniques will be investigated in this Line of Attention. With following this approach:

- Testing can be started before all components are completed,
- Testing of combined multi-disciplinary components can be done with simulated hardware components, and
- Testing can be made concurrent with system development, allowing an increase in the total testing rigor.

References

[1] TorX, `http://www.purl.org/net/TorX/`,2005

[2] R.R.H. Schiffelers, D.A. van Beek, K.L. Man, M.A. Reniers, J.E. Rooda, *A Hybrid Language for Modelling, Simulation and Verification*, `http://yp.wtb.tue.nl/pdfs/5281.pdf`, 2005

[3] D.A. van Beek, K.L. Man, M.A. Reniers, J.E. Rooda, R.R.H. Schiffelers, *Syntax and Consistent Semantics of Hybrid Chi*, CS-Report 04-37, `http://w3.wtb.tue.nl/nl/people_pages/?\&script=showabstract.php\&outid=4880`, November 2004