Automated Testing of Reactive Systems: Report 0101 - 0111

Goals

The project aims to further develop techniques for automated testing of software systems. The focus of the project is on (semi-)automated generation of Test Programs and Test Sequences. The project does not intend to contribute significantly to techniques for automated test execution: such techniques depend on the details of the concrete system environment, and there are several commercially available tools for automated test execution (of which WinRunner is perhaps the most well-known).

Executive Summary

The project has been conducted along several tracks, which are:

- A Test oracles for embedded systems (with Volvo TD)
- B Work on MSC and testing (with Telelogic AB)
- C Work on functional testing of WWW-based services (with Validation AB)
- D Transformation of process structures in SDL. (with Telelogic AB)

The main things to report for the period have been

- A: Completion of M.Sc. thesis of Fredric Flink.
- B: Publication of paper in SDL Forum' 01
- C: Completion of M.Sc. thesis of Jonas Boustedt.
- Recruitment of Ph.D. student, Anders Hessel, who has been working since 01-09-15.

Plans for the future:

- Anders Hessel will produce a survey of the testing area, to be completed in jan/feb 02. More concrete project planning will be based on this survey.

Approach

The basic view of the project is that a program which tests the functionality of a system must perform (at least) two functions.

1. Generate input data to the system under test, which should represent a range of real world situations, such as potential failures and anomalous situations.
2. Check that the system's response to input data (i.e., the output) conforms to the system's specification.

The project has addressed the second of these problems, by developing techniques for automated generation of test oracles from formal specifications. A test oracle is a component which observes the system under test and reports when the observed behavior does not conform to the system specification. If such a test oracle can be generated from formal specifications, then much effort in writing test programs and test scripts in lower-level languages can be saved. Moreover, automated oracle generation rewards the creation of good high-level specifications by establishing a direct link between specifications and actual system behavior. The language for formal specifications can be seen as a high-level test
scripting language, and generating a test oracle can be seen as compiling the high-level description into an executable form. In addition, the formal specifications can be used for other purposes, such as documentation, formal verification, etc.

**Techniques for generation of test oracles from formal specifications** have been developed in two application domains:

- **A1** For embedded systems in automotive applications, a technique for generation of test oracles from formal requirements in a real-time temporal logic has been developed and implemented. Real-time logics are suitable for specifying requirements for embedded systems and safety-critical applications. The technique uses the connection between temporal logic and automata theory, which is well-established in model-checking, extended with some techniques for handling data. An M.Sc. project at Volvo Technical Development and Uppsala University has developed and implemented a technique for translating formal requirements written in TRIO, a real-time temporal logic, into FIL (Fault Injection Language), a language used for programming test equipment at Volvo Technical Development. As example requirements, the project uses safety requirements for an idealized cruise controller developed by Volvo Technical Development. In an earlier M.Sc. thesis by Johan Nielsen at Prover Technology AB, these requirements have been formalized in TRIO. This work is reported in John Haakansson’s M.Sc. thesis. A slight generalization of the work, for temporal logics in general is reported in a draft paper (Håkansson/Jonsson/Lundqvist).

- **A2** The work in item A1 is related to the M.Sc. project of Fredric Flink, completed during spring 01 at Volvo Technical Development, in which a simulation model for distributed systems using the CAN bus is developed in Simulink. The M.Sc. project has developed a model of components associated with the CAN bus in the MATLAB/Simulink environment. The oracle generation which was developed in item A1 will be adapted to generate oracles, in the form of C modules, that can “observe” simulations in this environment.

- **B** For telecom applications, a similar technique is being developed for generating test oracles from requirements formulated in the MSC (Message Sequence Charts) notation, according to the MSC 2000 standard by ITU-T. An M.Sc. project at Telelogic AB and Uppsala University is developing an execution model for the MSC 2000 standard. This execution model can directly be used to generate test oracles from requirements written in MSC, and can also be used to generate test sequences. A basic motivation for this M.Sc. project is to simplify the generation of test sequences from MSCs, which is performed by the tool Tau. In the current version of Tau, MSCs do not carry enough information to generate test sequences. For instance, MSCs do not specify the values of message data parameters, and do not specify initialization (preamble) that precedes a test sequence. The new standard MSC2000 contains facilities for specification of data parameters, and for specifying high-level control structures. This work is reported in the M.Sc. thesis of Gerardo Padilla, and in a paper presented at SDL Forum, June 01 (Jonsson/Padilla 01).

The project also addresses generation of test scripts from informal specifications and recorded behaviors of the system. The motivation for this part is that (semi-)formal specifications of the system are not available. Our approach to this problem is to first generate a semi-formal specification of the systems intended behavior. This specification can have the form of a control graph (or flow graph): the nodes represent different "modes" or "stages" in the behavior of the system, and edges are associated with operations on data in the system. Test suites and test scripts can be derived from such a representation.

In M.Sc. projects at Telia Validation and Uppsala University, techniques for generating specifications, and for generating test scripts have been developed for transaction-based
services with a WWW interface.

- **C 1** An M.Sc. project at *Telia Validation* (completed 99-12) has produced a proposal for constructing functional specifications of WWW applications in a graphical structure which can, but need not, be similar to the linked structure of HTML pages used for the application. An specification of a small WWW-based booking service has been carried out in this format, together with test scripts.

- **C 2** An M.Sc. project is developing and implementing techniques for constructing graph-structured functional specifications of services with WWW-based interfaces, automatically or semi-automatically, from observed interactions with the service. The main idea of the project is to record sequences of HTML pages, and from these sequences attempt to construct a graph which can generate such sequences. The project has finished, and is reported in the M.Sc. thesis of J. Boustedt (Dec. 01).

**E** A final subproject, named *Transformation of Static Process Structures* aims to develop and implement techniques for statically optimizing the process structure of an SDL specification. This project is closely related to the compilation projects, but is administratively accounted for in the testing project. The project is under completion, and an M.Sc. report is under preparation.

**Description:** An SDL specification usually consists of a collection of processes that interact by sending messages asynchronously. There are no global variables. In the naive approach to compilation of multi-process SDL specifications, each SDL process is translated into a C process, which is scheduled by a customized scheduler. This may sometimes create a substantial overhead. It may sometimes be advantageous to merge closely coupled SDL processes into one process, in order to avoid the overhead of asynchronous communication and process scheduling. The project should result in a technique for merging SDL processes, in the context of the Tau tool, and principles for deciding when such a merging is appropriate.

**Progress Report**

- **Goals of the Period**
  - to follow the ASTEC evaluation (Sept. 00), which remarked that the project needs focussing.
  - to recruit at least one Ph.D. student.
  - to establish testing as a research direction on the same level as the other directions in ASTEC, by submitting of papers to high-quality conferences, producing software that is used by others.

- **Accomplishments of the Period**
  - Completion of Flinks M.Sc. project D.
  - Recruitment of Anders Hessel a Ph.D. student.

- **Deviations from plan** were

- **Prospects for the future** are that the project will gain momentum with a Ph.D. student. There is potential to recruit one more Ph.D. student, and to achieve synergy effects with funding for basic research in the area.

- **Deliverables**
Important Events

Contributions of associated companies

<table>
<thead>
<tr>
<th>Company</th>
<th>Contribution</th>
<th>Affiliation</th>
<th>Time Period</th>
<th>Involvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telelogic AB</td>
<td>235 kSEK</td>
<td>Edvinsson, Nordin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validation AB</td>
<td>320 kSEK??</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volvo Technical Development</td>
<td>97 kSEK</td>
<td>Flink, Lonn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Personnel 2001

<table>
<thead>
<tr>
<th>Person</th>
<th>Affiliation</th>
<th>Time Period</th>
<th>Involvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bengt Jonsson</td>
<td>UU</td>
<td>0101-0112</td>
<td>20 %</td>
</tr>
<tr>
<td>Ola Lundkvist</td>
<td>Volvo TD</td>
<td>0101-0112</td>
<td></td>
</tr>
<tr>
<td>Henrik Lönn</td>
<td>Volvo TD</td>
<td>0101-0112</td>
<td>5 %</td>
</tr>
<tr>
<td>Fredric Flink</td>
<td>Volvo TD</td>
<td>0101-0112</td>
<td>1 exjobb</td>
</tr>
<tr>
<td>Jonas Boustedt</td>
<td>Gävle University</td>
<td>0101-0112</td>
<td>30 %</td>
</tr>
<tr>
<td>Stefan Magnenat</td>
<td>Validation AB</td>
<td>0101-0112</td>
<td></td>
</tr>
<tr>
<td>Anders Hessel</td>
<td>UU</td>
<td>0109-0112</td>
<td>80 %</td>
</tr>
<tr>
<td>Niklas Edvinsson</td>
<td>Telelogic AB, M.Sc.</td>
<td>0108-0112</td>
<td>100 %</td>
</tr>
<tr>
<td>Johan Nordin</td>
<td>Telelogic AB, supervisor</td>
<td>0108-0111</td>
<td>120h</td>
</tr>
</tbody>
</table>

Achievements during 2001

- **Results**
  - a simulation model for distributed systems using the CAN bus is developed in Simulink.
  - a technique (implemented) for generating control graphs for WWW applications.
- **Activities**
- **Interest** by other companies: Discussions with ENEA have been conducted.
- **Exchange** of personnel. Anders Hessel has been recruited from Ericsson as of 0109.
- **Cooperation** with other ASTEC projects: Collaboration with the UPPAAL team is discussed.
- **Cooperation** with undergraduate education: Two lectures on testing developed and regularly given on undergraduate SE courses.
- **Cooperation** with other research groups: Collaboration with Univ. of Dortmund is discussed regarding specification and testing of WWW services.