Topic descriptions for
Nokia Cooperative Education 2005.09.01

Automated Validation Execution Environment

Create a general Validation Execution Environment to run and analyze the output of single/multiple test suite with one command. The framework allows the software designer to easily configure the specific commands needed by each test suite.

Examples of execution commands configurable with the framework:

- Test case grouping
- Test case order
- Test case prerequisites
- Setting of environment variables
- Different compilation options

Automated Verification in IP Multimedia Subsystem environment

During the verification phase of the software life cycle test execution has to be repeated several times. Making this phase automated increases the quality of both the testing and the overall quality level of the developed software.

Applicant shall take part in development of an environment for automated functional verification of Nokia's 3G Call Processing Server (CPS). The task is to get familiar with the current methods of CPS verification and its environment. Collect improvement ideas from verification engineers and implement an easy-to-use macro system.

Automation of C++ test environment

Applicant shall create an automated test environment that is able to create/run/analyze simulator macros based on the input data that is predefined.

In details:

- Use Nokia internal tools to create the simulator macros
- Prepare the environment and execute the simulation
- Make scripts to automate the generation and run of the macros
- Automate the analysation of the created output logs
- Summarize the result of the executed simulators

Component simulation

Software implementation workflow includes thorough verification to ensure high quality product. This phase includes not only the simulation of correct and incorrect cases but also the continuous development of the simulation environment itself. As a cooperative student you shall examine the existing simulation environment and improve it based on your findings and ideas. In this task you shall work in close cooperation with software developer engineers to get familiar with the current situation and to highlight the areas that can be subject of improvement

GPT tool development

Module and subsystem testing is a key phase in the software development life cycle. Software engineers spend significant time to design and implement test cases that simulate correct and incorrect behavior that their module or subsystem have to handle. The quality and the usability of test tools have major impact on both software quality and implementation efficiency.

Nokia puts high priority on continuous development of its development and test environment.

The GPT tool is a windows based application that is used for test sequence design. The current version is able to edit message sequences and their content.

Applicant shall take active role in tool development. Your task will be to read tool requirement specification, to implement and pilot the tool and to provide assistance to software engineers for their testing.

Performance measurement in subsystem level

Throughput capacity is one of the most important performance factor of a telecom switch. Measurement of it traditionally happens when the complete software package is integrated. Since this happens only at a late phase of the software development the correction of the found bottlenecks can be very difficult. Early measurement of single components gives priceless help for designers to enhance the quality and capacity of their code.

With the guidance of Nokia experts your task is to design and implement a measurement environment and execute different measurements on selected components. Evaluate the results and propose improvements.

Functional verification

Software development starts with understanding a problem statement. This understanding is than applied in each phase of the software life cycle to achieve perfection.

Of course errors and bugs can always be found in a piece of code. Recognition and localization of those are the most challenging tasks an engineer can have. It requires general knowledge about the application area, good programming skills and sense to spot possible weak points.

Finding and correcting errors and bugs improves one’s programming skills the best. Your task on the verification field is to drive the system into all possible situations and check whether it reacts in the correct way. If it fails the operation you have to find the source of the problem by checking the execution logs and the code and give suggestions for correcting the faulty behavior.