Framework for Developing and Testing Dependable and Safety Critical Systems

1 Objective

The purpose of the project is to develop an open methodology and framework for model analysis, which increases the quality of software design for embedded and reactive systems.
Rather than a complete, stand-alone design technology, add-on elements are aimed to be created for existing tools (based on the standards of software development) that would radically increase the quality of design. Such a standardized tool is the Unified Modeling Language (UML), regarded as the leading and most efficient visual software-development tool.

The scope of the project is the field of distributed, embedded and reactive systems, as the most dynamic improvements can be expected in that field. The design process of such systems necessitates more sophisticated methodology than the one of traditional IT systems as:

- Their life-time exceeds the life time of the latter ones, thus the probability of both hardware faults and the manifestation of hidden software faults during a longer time is very high.
- Their interaction with the environment cannot be limited.

2 Problems to Be Solved

Assuring the quality of service and increased productivity are the most important questions of developing embedded and reactive systems. For this purpose, exhaustive fault detection and prevention is necessary. However, faults cannot be corrected in a pure implementational, technological way.
The main causes of failures occurred (based on international statistics):

- Specification faults: the assurance of specification correctness and completeness is impossible by using conventional methods
- On one hand the number of permanent faults of control electronics has decreased due to technical development, on the other hand, the number of transient faults has increased due to the fact that miniaturized technology is more sensitive to noises and electromagnetic interference.
- A high rate of operation failures is a consequence of misconcepts in software design (especially on a system level).

The design methodology of embedded systems has to guarantee the faultlessness of specification, design and implementation, moreover, the handling of occurring faults up to a certain risk limit.

3 The Scope of the Project

UML, the standard object-oriented modelling language standardized by OMG turned to be a breakthrough in the field of visual programming. It summarizes and integrates modern software development paradigms (like object-orientation, modularization, design patterns, etc.). The visual programming followed by automatic code generation eliminates human mistakes from the final step of implementation, however, it cannot assure avoiding system technological faults during the development phase. Even visual models can be syntactically correct but semantically incorrect at the same time. These semantic faults can be so complex that they can only be detected by mathematical analysis.

The scope of the project is to develop a UML-based add-on mathematical model-analysis framework and methodology guaranteeing the quality of service, which:

- examines and proves specification correctness and completeness
- proves the behavioral correctness of the system, examining the results of faults in the environment and the control system

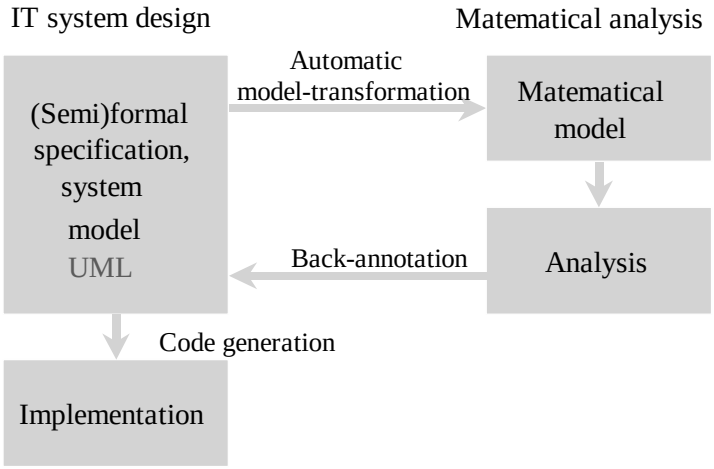
- proves the correctness of the logical control of internal processes

The system fulfilling the previous requirements will be realized in an open way:

- Using standard data-interchange formats allowing the use of a broad scale of development and verification tools
- Allowing the application of novel verification methods on user demand

Further goals are the following:

- Design decisions of developers should be recorded according to the ISO 9000 standard in a documented and verifiable way
- The methodology and tool set should support the reuse of modelling solutions



As a result of an integrated design process (shown in the figure), mathematical models are generated from a (semi-)formal UML based specification. These mathematical models are verified by different formal techniques in order to reveal quality bottlenecks, and the results are back-annotated to the system model. As a result, design methodology is raised to such a level by the integrated expert tool that a normal developer could never reach due to the lack of skill or available time.

3.1 Verifying Specifications

The first goal is to check the correctness and completeness of the system defined by semi-formal UML specification.

Completeness of an embedded system means that the system's response has to be specified to an arbitrary input sequence including changes in timing requirements.

Completeness of the specification means that neither conflicting requirements nor explicitly planned non-determinism in behaviour are present.

The main tasks of the project in concerned with verifying specification are:

- to collect the main verbal standards, and convert them into UML criteria
- to verify correctness and completeness of UML models from an XMI based description by static analysis
- to elaborate a transformation from UML models and standard criteria to temporal logic (LTL) formulae
- to provide a mechanism for the back-annotation of trace files generated by SPIN

These conceptional tasks will be verified on two benchmarks:

- In collaboration with our industrial partner No. 2 (Prolan Inc.) a train control system will be analyzed

- In cooperation with our partner No. 3 (B. Braun Medical Ltd.), the completeness and availability of an internal bus architecture designed for a safety critical medical system will be analyzed

3.2 Fault Propagation and Testability Analysis

It is a requirement for every highly-available system that its reactions to each fault considered have to be specified. Design for fault tolerance appears on two levels:

- The structure of the control system should not allow internal faults to affect the process controlled to such an extent that ends up in a catastrophic state.
- The system should react to faults occurred in an external process according to the functional specification.

It is possible in UML to automatically analyze both approaches in a single modelling paradigm.
Fault propagation and testability analyses have a data flow-network based model, while a hierarchical approach is applied traditionally for fault modelling. On the basis of such a model the following tasks are to be carried out in the project:

- A mechanism for fault propagation based on lists
- Testability analysis
- Test generation

3.3 Formal Verification of Control Processes

According to previous experiences, the verification of control processes is extremely critical in the design process of embedded systems. The typical tasks of control flow verification will be supported by mathematical analysis tools.
According to international practice, temporal logic is used traditionally for the underlying mathematics.
However, foregoing experiments have proved, that although the expressional strength of temporal logic used for describing specification is sufficient

- Run time performance cannot be predicted,
- Each semantic search method has to traverse the entire state-space ending up in a combinatorial state-explosion

For these reasons, our alternate approach to be elaborated is based on a Petri-net model, which should be created by automatic transformations from a compact model description.

- During the first phase, a comparative analysis on assessing efficiency will be carried out between the CPLEX system (regarded as the leading optimization tool) and the traditional approach based on temporal logic.
- During the second phase, a dedicated solution method will be elaborated which exploits symmetry embedded in mathematical structures resulting from logical verification and validation in order to increase efficiency. By applying that method, state spaces of models would traditionally be decreased and run-time performance would be improved to a great extent.