Dagstuhl Seminar 00131

Validation of Dependable Behaviour of Safety- and Mission-Critical Computing Systems

March 26-31, 2000
Schloss Dagstuhl, Wadern, Germany

Summary

The success of many applications strongly depends on the correct operation of computing systems – either because their malfunction would lead to a violation of safety or their non-availability would cause the failure of an important mission and thus end up in severe financial losses.

To prevent all of these cases special countermeasures against faults must be applied. Depending on safety, availability and performance requirements on one side and cost limitations on the other the designs of computing systems may extremely differ from each other. In the well-known field of fault-tolerant computing a variety of solutions have been developed and applied successfully. A lot of experience has been gained about the operation of fault-tolerant computing systems.

However, there are still challenges in the design and the evaluation of dependable computing systems. This Dagstuhl seminar was particularly focused on the following problems:

- How can one prove or at least demonstrate that a design exhibits in fact the desired dependable behaviour, even in the presence of any of the specified faults?

- How can the design process be controled or at least influenced by appropriate evaluation methods, in order to ensure dependable behaviour in early design phases – in later design phases redesign towards dependability is more expensive.

Usual testing methods cannot be applied because they are not able to cover realistic faults which are too rare, compared to the usual durations of testing (with respect to testing, the scarcity of faults causes problems, not benefits!). Moreover, testing requires the complete implementation which is only available at the end of the design process.

A general problem of the validation of dependable computing systems lies in their complexity. Efficiency requirements typically disallow straight-forward solutions to the dependability problem. Instead, tricky approaches must be chosen, which deal with different fault types, locations and times of occurrence in different ways. How can we make sure that no faults "slip through" the complex net of countermeasures?

Besides some exceptional contributions the solutions presented in the seminar fall into the following categories (and some combinations thereof):

- Structuring the design process
- Modeling and respective tools

- Strict formal verification (using first order predicate logic)
- Partial verification using reachability analysis
- Inspection of a design
- So-called pessimistic fault injection
- Combination of functional analysis and quantitative evaluation
- Efficient techniques for model-based dependability evaluation

As was expected there is no superior concept that solves all dependability problems at a time. Most promising is the combination of design principles and various analysis methods.

In general software and system development the usage of (partly) formal specification techniques like UML or SDL grows. As one of the main results of the seminar, it was concluded that the dependability-related specifications or models need not be different from their nature. Instead they can use the same modeling language. It only needs to be enriched by dependability parameters and conditions.

In this booklet, the abstracts of the contributions are presented. Special topics have been discussed deeply in working groups during the seminar. The reports of the working groups on

- Certification
- Development of safety-critical systems
- Formal methods

are included at the end of this booklet.

The organizers of the seminar, Andrea Bondavalli, Mario Dal Cin, Klaus Echtle and Erik Maehle express their thanks to all participants for their valuable contributions.

Abstracts of Presentations in Alphabetical Order of Participants

Andrea Bondavalli
Universita di Firenze, Departimento di Sistemi e Informatica, Firenze, Italy
a.bondavalli@cnuce.cnr.it

Dependability Modeling and Evaluation of Multiple Phased Systems and the DEEM Tool

Multiple-Phased Systems, whose operational life can be partitioned in a set of disjoint periods, called "phases", include several classes of systems such as Phased Mission Systems and Scheduled Maintenance Systems. Because of their deployment in critical applications, the dependability modeling and analysis of Multiple-Phased Systems is a task of primary relevance. However, the phased behavior adds a further degree of complexity to the analysis of these systems. We propose powerful and efficient methodology for the analytical dependability modeling and evaluation of Multiple Phased Systems, based on Deterministic and Stochastic Petri Nets and on Markov Regenerative Processes. Due to the special structure of Multiple Phased Systems, an analytical solution technique with a low computational complexity, basically dominated by the cost of the sepa-

rate analysis of the system inside each phase is defined. Last we describe DEEM, the dependability modeling and evaluation tool for Multiple Phased Systems, being currently developed to support our methodology.

Jehoshua (Shuki) Bruck

California Institute of Technology, Elec. Eng. Dept., Pasadena, CA, USA
bruck@paradise.caltech.edu

Computing in the RAIN: A Reliable Array of Independent Nodes

The RAIN project is a research collaboration between Caltech and NASA-JPL on distributed computing and data storage systems for future spaceborne missions. The goal of the project is to identify and develop key building blocks for reliable distributed systems built with inexpensive off-the-shelf components. The RAIN platform consists of a heterogeneous cluster of computing and/or storage nodes connected via multiple interfaces to networks configured in fault-tolerant topologies. The RAIN software components run in conjunction with operating system services and standard network protocols. Through software-implemented fault tolerance, the system tolerates multiple node, link, and switch failures, with no single point of failure. The RAIN technology has been transfered to RAINfinity, a start-up company focusing on creating clustered solutions for improving the performance, availability and scalability of Internet data centers. A paper describing the RAIN project is at: http://paradise.caltech.edu/papers/etr029.ps.

Pierre-Jacques Courtois

AIB Vincotte Nuclear, Nuclear Safety Analysis Dept., Bruxelles, Belgium
pjc@avn.be

Towards the Definition of an Architecture for Safety Cases of Computer Based Systems

My current work addresses the issue of structuring the validation process of dependable computer based systems. It is motivated by the desire to make the licensing and certification of these systems more reliable and efficient. The work is an attempt at analysing the structural, semantic and logic properties of the demonstration that a computer based system is adequately specified, designed and maintained in operations. Three classes of dependability claims are identified. Claims which address (i) the environment - system interface, (ii) the design and (iii) the operational behaviour. A structure is proposed to analyse the relations between these classes and the convergence of their supporting evidence. Relations and formal properties which should be satisfied by the underlying models, by the languages required for the interpretations of the real domains, and by the proof obligations are being studied. The implications of these properties on design criteria and design mechanisms are also investigated.

György Csertán

Budapest University of Technology and Economics, Budapest, Hungary
csertan@mit.bme.hu

Optimal Maintenance Scheduling

Maintenance is an important aspect of dependable systems, especially after the system becomes operational. In this phase system life-cycle the main issue of companies is to reduce the cost of production and thus, of course the cost of maintenance. However this cost reduction must not result a decrease of dependability.

Assume a system, which is built of interconnected but not independent components that can be maintained separately. Usually there are many ways to maintain a component. These maintenance actions differ in fault coverage, cost, necessary resources, execution time, etc. Therefore the primary aim of maintenance planning is to select a subset of possible maintenance action that yield minimal cost maintenance, while preserving maximal efficiency/productivity and dependability of the system. Solution of this problem means in most of the cases the solution of an optimization problem.

Another phenomenon of maintenance is that it is usually done under constraints. Such constraints are for example the limitation on time and duration given for a maintenance action, or the limited number of mechanics who are able to execute maintenance actions above a given complexity level. Therefore a maintenance plan should be prepared that consists of the time ordered sequence of maintenance actions. This kind of problem belongs to the class of scheduling and allocation problems in mathematics.

For the solution of both of these problems currently probabilistic approaches are used. In most of the cases these approaches are based on some Markov-, Petri Net-, or Stochastic Activity Network model, where costs are modeled by reward functions [1]. These approaches can be very precise in computing the various dependability measures, due to the underlying stochastic model. Unfortunately their usefulness in optimization is very limited and scheduling is not solved yet.

In our work we suggest a combinatorial approach instead of the probabilistic one. Modeling is based on process networks that are mostly used for describing production systems [2]. Similarly to stochastic models, process networks too have an elaborated mathematical background and the solution algorithms are much more efficient. This approach allows for less precise evaluation of dependability measures, only the availability of the system can be computed, but the computation is based only on the mean values of the stochastic variables. On the other hand the approach is very promising for optimization and scheduling.

In future work we plan to elaborate a larger size case study and to combine the two approaches in order to exploit the advantages of both of them.

References

[1] A. Bondavalli, I. Mura, and K. S. Trivedi. Dependability Modeling and Sensitivity Analysis of Scheduled Maintenance Systems. *In Proceedings of the 3rd European Dependable Computing Conference EDCC-3*, ISBN 0302-9743, pp. 7-23, 1999.

[2] F. Friedler, L.T. Fan, and B. Imreh. Process Network Synthesis: Problem Definition. *Networks*, John Wiley & Sons, vol. 31, pp. 119-124, 1998.

Mario Dal Cin

Universität Erlangen-Nürnberg, Informatik 3, IMMD, Erlangen, Germany
dalcin@informatik.uni-erlangen.de

Can Visual Models of Dependable Systems be Evaluated?

A central requirement for dependability-critical systems is the ability to cope with faults. It is important that this non-functional property can be validated before the system is licensed for use. This requires a quantitative dependability analysis. For such an analysis, it is not only necessary to model the system's behavior. Also its interaction with its environment has to be modeled (closed-loop modeling), since the environment can be a source of faults which can give rise to errors in the system's behavior. Thus, dependability evaluation of embedded systems tends to be very complex causing the modeling problem to be notoriously elusive and error prone. Therefore, when modeling embedded systems a trade-off has to be made between the degree of details in modeling and the degree of possible automation of the analysis. This lead us to define a sub-class of UML-statecharts comprising so-called Guarded Statecharts. They are, however, not directly amenable to a quantitative analysis. Therefore, a method has to be introduced which transforms a set of concurrent statecharts into a mathematical model that can be evaluated quantitatively. We present a technique for transforming Guarded Statecharts, consistent with UML semantics, into a set of interacting Stochastic Reward Nets (SRN). Stochastic Reward Nets are extensions to Generalized Stochastic Petri Nets (GSPN). On the one hand, this gives us the possibility to employ the elaborate and well established Petri Net tools for the quantitative analysis of UML-models. On the other hand, this integrates the use of Petri Nets into the object-oriented modeling paradigm of UML.

Elmar Dilger
Bosch GmbH, Stuttgart, Germany
dilger@fli.sh.bosch.de

Complex safety related functions in future automotive systems will be more and more based on electronic components. They will no longer rely on mechanical or hydraulic back-up. The general tendency towards dry systems, the constructive advantages resulting from the simplified packaging of non-mechanical components, the big potential of increasing passive and active safety and the easy integration of driver assistance systems are some of the benefits of these so-called „X-By-Wire"-systems.

The "X" in "X-By-Wire" represents any safety related application such as steering, braking, power train or suspension control. These applications will greatly increase overall vehicle safety by liberating the driver from routine tasks and assisting the driver to cope with critical situations.

Due to the required level of safety an X-By-Wire system must be distributed and consisting of fault tolerant units connected by a reliable real time communication medium.

The work currently done comprises:

- Overall system design (e.g. time triggered design),
- Fault tolerance strategies,
- Layout of electronics and electric energy supply,
- Fault tolerant communication protocols,
- Software concepts,
- Selection and coupling of actuators and sensors,
- Fail-silent and fault-tolerant actuators and sensors,
- Development process.

Klaus Echtle

University of Essen, Department of Mathematics and Computer Sience, Dependability of Computing Systems, Essen, Germany – echtle@informatik.uni-essen.de

Fault Injection

Fault injection can be used to check the behaviour of a system in the presence of faults. Two software-implemented fault injectors have been developed for different purposes:

– EFI-Tool: Fault injection into the communication system serves for the **test** of fault tolerance protocols in distributed systems. The faults (more precisely: errors) to be injected are derived from reachability analysis of a formal model – a special type of an attributed Petri net model which is generated automatically from an SDL protocol specification.

– ProFi-Tool: Fault injection into the processor serves for the **quantification** of fault coverage. ProFi can inject both temporary and permanent faults. The faults to be injected are specified by test personnel using a special fault injection language, called ProFiL. The injector ProFi is particularly designed to emulate faults in as many processor parts as possible, even those which are diffucult to access by software.

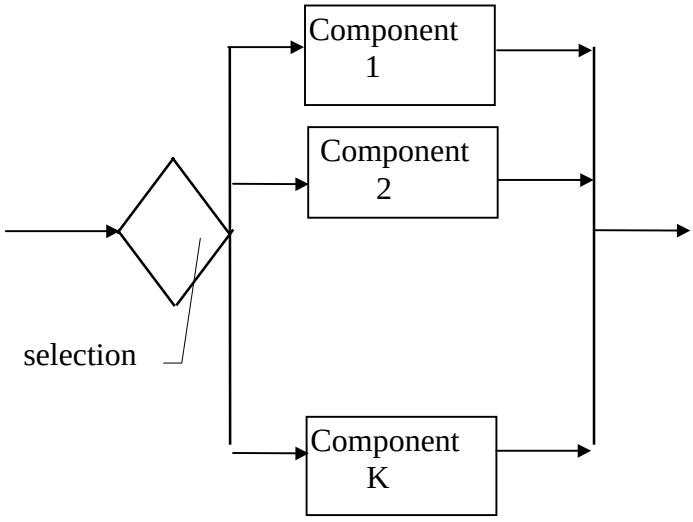
I hope for an interesting discussion on the representativeness of injected faults. Different injection techniques should be compared and the principal limitations of fault injection be expressed. For the further development of fault injection methods two questions get importance: Can we implement "pessimistic" fault injection? Can we define a "standard interface" for fault injectors?

Wolfgang Ehrenberger

Fachhochschule Fulda, FB AI, Fulda, Germany
Wolfgang.Ehrenberger@informatik.fh-fulda.de

Stratified Sampling –
Use of Operating Experience or Test Results for a New Demand Profile

When trying to exploit operating experience of software, it often turns out that the existing operating experience does not exactly reflect the usage of the software in the new application. The components as shown in the figure have been selected with probabilities different to what the future application will require.



These different demand profiles can be mapped on each other. Consideration of the different sizes of the demand strata leads us

to formulae that enable to calculate reliability figures at the levels of confidence required by the whole system. If necessary, any addionally needed number of test cases are determined for each component.

In principle two means of calculation exist: One starts from the addition theorem of the Poisson distribution, the other from fixed high levels of confidence for the failure probabilities of the individual components.

Rasaria Esposito

ANSALDO TRANSPORTI S.p.A.Napoli, Italy
esposito.rosaria@asf.ansaldo.it

Immacolata Marcarelli

ANSALDO TRANSPORTI S.p.A.Napoli, Italy
pellecchia.raffaele@atr.ansaldo.it

Ansaldo Segnalamento Ferroviario works in railway signalling critical system.

Railway signalling application can be divided into vital-systems and non-vital systems. Some example of vital-systems are:

- Automatic Train Protection (ATP)
- Train Routing (Railway stations)
- Train Spacing (Railway lines)

Non vital System are:

- Local/remote control of equipment of station and lines
- Automatic train circulation management
- And so on

ASF uses many tools and technologies in the V&V of critical and non critical applications. In particular the RAMS group uses:

- SDL and MSC, Model checking, Petri's net for verification of the specification
- Fault test with simulation technics for project hardware verification
- Software static analysis based on evaluation of quality metrics
- Internal tools and CASE tools for function test specification
- Internal toolset (LIVE) for Software dynamic analysis
- Fault injection for dependability tests

Felix Gärtner

Technische Universität Darmstadt, FB 20 – Informatik, Darmstadt, Germany
felix@informatik.tu-darmstadt.de

A Fault Classification Scheme Based on Safety and Liveness

In order to formally validate fault-tolerant systems, it is necessary to precisely describe the faults which are assumed to happen. As long as this faulty behavior is discrete, it can be modeled using the notion of a program transformation which adds states and transitions. If a system previously satisfied a property, the transformed system may violate it due to the additional behavior. We consider the two important property classes of safety and liveness and ask whether or not a given fault assumption has the potential of causing violations of these properties and whether or not such a violation can be tolerated by adding fault-tolerance mechanisms. This gives rise to ten precisely separable fault classes. We give examples of each of these classes and discuss their relation to the many fault classification schemes which exist in practice.

Winfried Görke

Universität Karlsruhe, Institut für Rechnerentwurf und Fehlertoleranz, Zirkel 2, Karlsruhe, Germany – goerke@ira.uka.de

Safety Requirements for Complex Systems – Lessons Learned from Tunnel Fire Hazards

Complexity increases not only continuously in information systems but also in other areas of concern to modern society, in particular in traffic and transportation. Although augmented frequently in small steps only which are scarcely noticed sometimes newly conceived complete systems are designed. Examples are supersonar aeroplanes, high velocity trains or rail or motor traffic connections of outstanding length across mountains or water. Looking at our big tunnels through the Alp mountain ridge and under the channel requirements and specifications for fire safety during the operational period are outlined and explained. It can be shown that there is evidence not only to plan and construct modern civil engineering systems with record breaking features of operation but also to achieve safety characteristics unsurpassed by existing structures, provided one can rely on human factors in operation, maintenance, and safety management.

Karl E. Grosspietsch

GMD, St. Augustin, Germany
karl-erwin.grosspietsch@gmd.de

1) Problems of Dependability in Mechatronic Systems

Mechatronic systems, combining complex digital, analog-electrical and non-electrical components, are currently being developed for a wide spectrum of applications. Many of them are to be used in fields, where either the performance is safety-critical (e.g. airbag control in automotive systems), or even a guaranteed continuous proper and undegraded service is necessary over a long application life-time, as e.g. in medical implants, where the system is practically inaccessible for external repair during all of its operational time. Thus, problems of reliability and safety will play a decisive role for successfully operating many of those systems.

In this talk first an approach is sketched to modify an existing simulator core to comfortably enable fault injection experiments especially for these domains. The utilized simulator has mainly been developed to analyse analog-electrical circuits. Often, however, mechanical components can be described by differential equations which are similar or even structurally identical to well-known equation systems used for the description of certain analog-electrical devices. In our ap-

proach, we try to systematically apply such structural relationships to make the simulator applicable also for the failure analysis of mechanical components. This means that the structural properties can be comfortably described by means of constructs which correspond to the analog-electrical circuit building blocks.

In addition, we shall discuss potential further dependability issues (testing, insertion of redundancy) which can be based on the introduced fault simulation approach.

2) Efficient Realization of Software Redundancy Schemes

Today, we can observe a growing number of applications where the computer is used to control safety-critical processes, as e.g. in avionics, industrial process control, and command or control systems. Here also the reliability of the implemented software is of decisive importance. To protect the application against software errors which inevitably might be created, redundancy schemes like N-version programming have been proposed. Usually, the tradeoff of this methodology is a considerable increase in the cost of software implementation. Therefore, methods seem necessary and useful which support the applicability of N-version programming by providing reduction of these costs, where possible.

In this talk, some methods for balancing cost vs. reliability by use of optimization methods are exhibited. This is described by means of a formalized model of the software system; this model considers reliability, costs and timing as well as resource requirement parameters of each component of the software system. Optimization is then performed by a novel adaptive, random-driven search within the search space of the component configurations.

In addition, some approaches for a more flexible implementation of control mechanisms, e.g. with regard to majority voting, are discussed.

Johan Hedberg

Swedish National Testing & Research Institute, Physics and Electrotechnics, Software & Saftety, Boras, Sweden – johan.hedberg@sp.se

I participate in a Swedish research project callled "PÅLBUS". "PÅLBUS" is a Swedish abbreviation for dependable bus systems. The main purpose is to gather the knowledge from industry, institutes and university regarding how to construct and validate distributed control systems with dependability demands. The project consists of following workpackages: Definitions: Define which terminology that should be used in the project and also give an overview of applicable standards. Comparison between different concepts: Overview of the differences between event triggered (CAN) and time triggered (TTP) protocols, both functional and also to what degree dependability aspects is introduced in the protocols. System construction and know how: Describe how increased complexity in the systems can be handled without making the system incalcuable and "incomprehensible". Specification: Which parameters should be included in a specification of a dependable bus system. Fault detection and fault handling: Which methods are suitable for diagnostics and handling of faults in distributed control systems. Design principles for dependable CAN and TTP systems: Which features must be considered to implement dependability. Tools and methods for development of distributed control systems: Which development tools are used today? Special emphasis will be put on support for validation. Study of test methods used in industry: State of the art in industry and how to improve the testing. Validation methods: How should a distributed control system be validated to reach a certain level of dependability. Present

a number of methods and examples. The last part of the project will consist of analysis and practical testing on a prototype system that is built up.

Günter Heiner

DaimlerChrysler, Research and Technology, Berlin, Germany
guenter.heiner@daimlerchrysler.com

Safety Engineering and Validation of Automotive Systems

My area of work is system safety of automotive systems. The work is focussed on safety and reliability processes, methods and tools for dependability analysis and prediction, safety and fault-tolerance architectures, and engineering of safety-related distributed real-time systems. The application areas cover mostly road vehicles, but also railway and aerospace systems. Examples are innovative systems for electronic vehicle guidance, e.g. future by-wire systems in cars, or train control systems. Our work can be characterised like that: We develop technologies and methods – we implement and evaluate them in prototypes or pilot projects – we apply them in projects with the business units and we consult our customers.

In my talk I tried to give an overview on the safety engineering for automotive systems, summarising the state of the art and highlighting some of our research activities. Main topics are the safety process, safety analysis methods, testing, and robustness analysis of design models.

Markus Jochim

University of Essen, Department of Mathematics and Computer Sience, Dependability of Computing Systems, Essen, Germany – jochim@informatik.uni-essen.de

A virtual duplex system (VDS) can be used to increase safety without the use of structural redundancy on a single machine. A VDS which calculates a given function f consists of two variants Pa and Pb of a program P, which are calculating the diverse functions fa and fb, respectively, such that f = fa = fb holds, if no error occures during the design and execution of Pa and Pb. For a given input i the VDS calculates and compares the values fa(i) and fb(i). The presence of an error can be detected if fa(i) <> fb(i). The main problem is, that the presence of an error might also lead to the scenario: "f(i) <> fa(i), fa(i) = fb(i)" which means that the error takes effect on the results, but remains undetected.

It has been shown that virtual duplex systems have a high ability to detect hardware errors. This ability stems from the fact that two diversified programs will use different parts of the hardware with different data with a certain probability. One would expect that this probability is the higher, the more diverse the two program variants are.

Numerous diversity techniques have been deeply investigated in the past with respect to the detection of hardware errors. Most of these techniques could be called "manual diversity techniques". Using manual techniques like design diversity is very time consuming and will therefore result in high cost. Therefore my work concentrates on the introduction and evaluation of "automatic diversity techniques". Roughly spoken you can automatize the generation of diversified program variants which are optimized to detect hardware faults by the use of rules which

describe how to modify a given assembler program combined with an appropriate optimization strategy which aims at diversity.

The quality of an automatically generated VDS of course has to be evaluated. This can be done by means of processor fault injection which forms another field of interest of mine.

Diego Latella

Consiglio Nazionale delle Ricerche – CNR, Instituto CNUCE, S. Cataldo, Italy
diego.latella@cnuce.cnr.it

With reference to the subject of the Seminar, my work regards the development of formal semantic models for UML Statechart Diagrams (UMLSD) for formal verification of UML models expressed using the above notation. The role of formal verification in the context of dependability is nowadays widely recognized. The obvious first step of our work was the definition of a formal operational semantics for (a behavioural subset of) UMLSD.

On the basis of such a semantic model

i. A translation from UMLSD to PROMELA has been defined, proven correct and implemented. It provides a tool for the automatic verification of properties of UMLSD behaviours expressed in linear time temporal logics.
ii. A translation to labelled transition systems has been defined. It provides a tool for the automatic verification of properties of UMLSD behaviours expressed in branching time temporal logics. A related tool is under development.
iii. Some preliminary study took/is taking place for:
    a) the use of networks of automata as one way for dealing with memory problems, due to state explosion. Such an approach proved already convenient in other contexts of model checking.
    b) the extension of our semantics to more than one statechart.
    c) the use of true concurrency in the semantics.
    d) the enrichment of the semantic model with time.

Erik Maehle

Institute of Computer Engineering, Medical University of Lübeck, Lübeck, Germany
maehle@iti.mu-luebeck.de

PC-Clusters as Scalable Parallel Servers for Dependable High-Performance Network Computing

With the tremendous success of the internet and its applications like information services, e-business or entertainment there is a large demand for dependable servers which make use of fault tolerance. Today rather simple solutions like failover for two or three nodes dominate which are rather expensive and do not scale well with inceasing workload. Another important area is high-performance computing. Here, currently rather expensive parallel computers dominate. Therefore at our institute we are developing techniques to use standard scalable PC-Clusters (with up to hundreds of processors) with highspeed interconnection networks like Myrinet or SCI (Scalable Coherent Interface) as a cost-effective alternative. Sample medical applications as well as fault-

tolerant media servers based on object replication implemented on the Störtebeker Cluster of our institute (96 Pentium II processors, Myrinet interconnect) show that high performance can be achieved provided a fast network is used. A particular project in this context deals with the development of a rule-based routing chip for highspeed interconnection networks which is able to execute state-of-the-art fault-tolerant and adaptive routing algorithms. Currently a prototype is implemented based on Myrinet and simulations are carried out to predict the expected performance.

Istvan Majzik

Technical University of Budapest, Dept. of Information Management, Budapest, Hungary
majzik@mit.bme.hu

Completeness and Consistency Analysis of UML Statechart Specifications

UML can be used to construct software specification of embedded systems, often implementing safety-critical functions. Unfortunately, the specification is often incomplete, inconsistent and ambiguous. The errors in the specification are not only difficult and expensive to correct in the further phases of the life cycle, but may also lead to safety related failures.

Our work aims at the elaboration of methods and tools for the checking of some aspects of completeness and consistency in UML models. We concentrate especially on the behavioral part of UML, namely the statechart diagrams. Three types of analysis are presented. The first one checks completeness and consistency based on the static structure of the specification. This method does not require the generation of the reachability graph, thus it scales up well to large systems. The second one performs dynamic analysis by checking safety related reachability properties with the help of a model checker. It is restricted to core critical parts of the system. The third one helps the designer by automatically generating the fault tree of a system built upon redundant software components.

Finally, the automatic tool is presented which implements the static checking by combining Prolog questions (formulating the completeness and consistency rules) and SQL commands (navigation and search in the UML model database).

Andras Pataricza

Technical University of Budapest, Dept. of Information Management, Budapest, Hungary
pataric@mit.bme.hu

Quantitative Evaluation of Dependable Systems by Means of Operation Research Methods

The simultaneous growth in the complexity of information technology systems and the requirements towards the dependability of the services delivered by them requires more and more a proven soundness of the system concept and its implementation.

No current method can cover currently all the aspects relevant for such proofs. Pure logic (qualitative) methods can explore huge state spaces, however, they lack of an expressive power of some main quantitative features. Quantitative methods include these properties, but the current systems

enforce the use of oversimplified models due to mathematical tractability (eq. distributions) and state space size (10^8).

In the contribution it was shown that an automatic abstraction leading to a system description using a few enumerated types can still faithfully model the system. An algorithm was sketched on the algebraic reformulation of such systems and on the use of integer algebra algorithms in the solution/optimization problems in the dependability scene.

Rüdiger Reischuk
Universität Lübeck, Institut für Theoretische Informatik, Lübeck, Germany
reischuk@tcs.mu-luebeck.de

Analysing Data Access Strategies in Cache Coherent Architectures

The conception of multiprocessor systems based on shared memory is to provide logically uniform data access, even in systems with many processors. Since data requests that cannot be satisfied physically within the processor environment suffer from high latencies, these systems employ memory hierarchies (including caches) as known from uniprocessor systems. The main techniques to reduce and hide latency in memory hierarchies are buffering and pipelining of data accesses. In order to utilize these techniques in global shared-memory architectures, new memory consistency models have been defined and partially realized in hardware. The additional effort cannot be neglected and the validation of such an implementation turns out to be a difficult problem.

We discuss the cache coherence problem and three consistency models that are commonly used to enhance performance. A uniform parameterized model for different parallel architectures is developed and the latency for data access in each consistency model is estimated abstracting from the details of hardware mechanisms. The goal of our approach is to obtain realistic predictions of running-times for the various models. Depending on the structure of data access we can quantify the speedups of relaxed consistency for different implementations.
(joint work with Karin Genther)

John Rushby
SRI International, Computer Science Laboratory, Menlo Park, CA, USA
rushby@csl.sri.com

My work focuses on the application of automated formal methods to critical systems. I am interested in delineating the assurance that can be derived from use of formal methods, in methods for combining that assurance with assurances derived by other means, and in methods for increasing the automation and utility of formal methods. My group developed the PVS verification system and we continue to develop that system and its adjuncts, with particular interest in efficient decision procedures, in automated abstraction, and in combining theorem proving with model chekking. My colleagues and I have applied these tools to several issues and algorithms in critical systems design, including clock synchronization, consensus, and group membership.

Francesca Saglietti

Institute for Safety Technology (ISTec) GmbH, Garching, Germany
saf@istec.grs.de

Quantitative Techniques for Software Reliability Assessment

This talk focuses on the assessment of ultrahigh software reliability for safety-critical applications. Different forms of evidence are considered.

While non-operational evidence is felt to be qualitatively essential for certification, it does not allow at the time a significant statistical estimation of survival probability, due to variability of programming environment complexity and reliability demands in subjective engineering judgement.

Reliability quantification has rather to be based on operational evidence gained by independent simulation runs.

In this context, on-going work is dealing with the evaluation of operational experience gained with pre-developed software with respect to different past usage profiles, for the purpose of assessing its applicability to a new system.

Oscar Slotosch

Technische Universität München, Institut für Informatik, München, Germany
oscar@slotosch.de

Development of Safety Critical Systems Using AutoFocus and Quest

AutoFocus is a CASE-Tool Prototype for the development of correct embedded systems. Similar to other CASE-Tools it allows to describe the developed systems graphically using several different views. AutoFocus builds upon formal methods concepts. The available views are:

- Interface and structure view: By using System Structure Diagrams (SSDs) users define the components of the developed system and the interfaces between them and the environment.
- Behaviour view: State Transition Diagrams (STDs) describe the behaviour of a component in the system.
- Interaction view: Extended Event Traces (EETs) capture the dynamic interactions between components (and the environment). EETs are used to specify test cases or example runs of the systems.
- Data view: In (textual) the Data Type Definition (DTD) view define the data types for the description of structure, behaviour and interaction diagrams. We use functional datatypes.

All views are hierachic to support descriptions at different levels of detail. AutoFocus can check the consistency between different views using an integrated consistency mechanism. AutoFocus offers a simulation facility to validate the specifications based on rapid prototyping.

Quest consists of validation tools that are connected to AutoFocus to validate the models.

- SMV for model checking;
- SATO for bounded model checking;
- CTE for the classfikation of test values for the models;
- VSE II for interactive theorem proving.

Furthermore there is an abstraction chooser (and an abstraction theory) integrated, that allow to reduce complex models to simpler ones. Proof obligations (for VSE) are generated that allow to ensure the correctness of the properties in the complex system, if they have been verified in the simpler, abstract system. Specification-based generation of testcases from the model is also supported.

Lorenzeno Strigini

City University, Center for Software Reliability, London, Great Britain
strigini@csr.city.ac.uk

Some Central Issues in Software/Design Dependability Modelling

Quantitative reliability assessment of systems for what concerns design faults suffers from heavy uncertainties, and has gained a bad name because the mis-application of overly detailed models has let to falsely precise and obviously irrelevant predictions. Many people thus rely on qualitative, vague arguments of experience and common sense, which give a false sense of confidence because they are too vague and thus impervious to rigorous analysis. My talk gives examples of rigorous use of very simple probabilistic models to clarify difficult problems. These examples are about recent work we have done on design diversity, shedding some light on the three related problems of how much gain we should expect from diversity, how we should evaluate a specific diverse system and how is it best to pursue diversity in development.

Mark-Alexander Sujan

Universität Karlsruhe, Institut für Rechnerentwurf und Fehlertoleranz, Zirkel 2, Karlsruhe, Germany – sujan@ira.uka.de

Towards a Holistic Evaluation of Dependability

During the last few years there have been increasingly research efforts driven by the recognition, that the human element is not to be treated as a source of possible errors, but as a unique component, being part of any complex system, which excels with its creativity and flexibility, especially when abnormal situations are encountered. This is reflected by research like Naturalistic Decision Making, Human-Centred Design, Situated Action, or Distributed Cognition, where the major emphasis is respectively on the cognitive support, the quality in use, the situational dependency, and the distribution of knowledge. In spite of this evidence, both current design practice and dependability evaluation are based mostly on a separation of the individual resources. Often the focus during design is on the efficient realisation of new echnological artefacts, and system dependability is evaluated accordingly using a collection of methodologies and techniques (such as Fault-Tree Analysis, Failure-Modes-Effects Analysis, and Reliability Growth Modelling), which have proved to be successful in predicting the behaviour of technical components. These do not explicitly include human behaviour in their analysis, and they are therefore frequently complemented by techniques for Human Reliability Analysis (HRA), which are generally derived from methodologies for hardware and software reliability assessment (e.g. THERP), in order to allow an easy integration into existing approaches (like Probabilistic Safety Assessement). Most of the HRA approaches employ empirical estimations of human error probabilities (HEP), and are not

supported by a sound psychological description of human behaviour. In this contribution we would like to propose a research effort directed towards a holistic approach to the dependability evaluation of complex systems. This approach is based on the activity-theoretic conception of human behaviour, which is rooted in the cultural-historical school of Soviet psychology. It emphasises the central role played by artefacts (in the most general sense, including any representation used in an activity, whether internal or external to the subject) in human cognition. The knowledge required to perform a specific process is conceived to be distributed across people, artefacts and the rules and procedures, which guide the use of these artefacts. Therefore, a complex system consisting of hardware artefacts, software, rules, procedures, and humans has to be analysed as a single cognitive entity. This assumption has important implications for the development of a methodology for system dependability evaluation. First of all, it implies that human reliability cannot be assessed without taking into account the material and social environment within which human activity takes place. Secondly, and more far ranging, it implies that a complex system cannot be analysed as the simple sum of its hardware and software components. Their behaviour is strongly affected by human behaviour, which in turn cannot be viewed as belonging to the external environment. Therefore, system modelling needs to be based on a detailed analysis and description of all the resources interacting in order to achieve the system mission.

Neeraj Suri

Chalmers University of Technology, Electrical and Computer Engineering, Göteborg, Sweden
suri@ce.chalmers.se

FT-RT Protocol Validation: Can Formal Methods Help Identify Test Cases?

A key feature in fault injection (FI) based validation is identifying the relevant test cases to inject. This problem is exacerbated at the protocol level where the lack of detailed fault distributions limits the use of statistical approaches in deriving and estimating the number of test cases to inject. In this talk we develop and demonstrate the capabilities of a formal approach to protocol validation, where the deductive and computational analysis capabilities of formal methods are shown to be able to identify very specific test cases, and analytically identify equivalence classes of test cases.

Working Group Reports

Report of the Discussion on Certification

Moderator: Winfried Görke

Participants: P.-J. Courtois, K. Echtle, J. Hedberg, W. Ehrenberger, F. Saglietti, F. Gärtner, L. Strigini

The terms indicated were formalism versus practitioners with the relation to certification, the implication of formal methods and their influence upon standards. The times allowed to the group

meetings where spread across two days during the seminar week. There were two morning sessions on March 29 and March 30 and also two afternoon sessions on March 30. No special agenda with the exception of the above mentioned terms was prepared. So in the beginning there were just informal discussions of the range of problems related to the terms assigned to the group. During the first hour plenty of suggestions resulted from all participants which also included frequent references to the presentations given before in the seminar plenary sessions.

Due to the personal background of the persons participating at the group discussion a broad field of areas was addressed. It will be described in the following without any specific order. The certification process in different application fields, namely nuclear power generation, aircraft industry or chemical plant engineering, can be characterised by a large number of influences. The differences are frequently due to the history of the particular field, the culture or education of the people involved, and the particular areas of application concerned where the level in the design and development process are of particular interest, namely:

- the plant and its environment,
- the design and composition of the system components, and
- the operation and maintenance period, which is especially important, once the operational period has started.

A different field of discussion considered the approaches related to probabilistic or logic methods and description of the safety engineering part of the system. Formalism should help to minimise or prevent misunderstanding. In many cases a consideration of functionality is frequently not sufficient. In particular the failure behaviour of COTS (components off the shelf) describes a subject to be investigated in much more detail. Another factor concerns modifications unavoidable during the operational life of a system. Also the languages used must be understandable by people of a different background. A design example broadly discussed by contradictory opinions is the system TELEPERM. It was pointed out that it shows the behaviour of an insufficient approach such that no influences or deductions are possible once the description has been put down in the particular form required.

Safety integrity levels (SIL) offer a classification according to the failure probabilities being reached by a particular design. But it was pointed out that it is a great danger to just rely on figures without looking more into the detail of their meaning. Generally always a third party (assessor or regulator) has to be included into the overall development process between provider of the system and the later customer or operator in charge of its operation. The requirements in future should more and to a larger extend than today make use of existing software and should not exclude the operating system or compilers or other tools available to the application field. In particular there was an agreement that in future no product related to safety applications should obtain a license by the lowest standards available, e.g. in some of the set of countries where the final product is expected to be offered on the market.

A certain list of unsolved problems were compiled including mainly:

- Today in many cases there is no software hazard analysis available although recently a first publication in the field has been reported. Operating systems are generally not documented with respect to their behaviour. This must be considered to be a missing link of a safety description for certification.

- Stratified test data seem to offer an interesting approach. Is it possible to reduce the effort (e.g. counted by the number of tests required to reach a particular level of confidence) by making use of this test data transformation in order to maintain the level of confidence reached before?
- Composition of large systems is another difficult area of interest. How to be sure that the original assumptions which were valid for the single component hold also for the overall system?
- Open questions concern also psychological research. It is necessary to vestigate in a quantitative manner human behaviour as part of the control loop of the technical system.
- Finally the role of formalisms in the licensing process remains largely open. Right now there are different processes in different countries. So far only Great Britain requires an analysis leading to a quantitative value for the expected failure probabilities.

Consensus could be achieved on most of the following subareas:

- The certification of safety integrity levels is confusing because of a dispute of the numbers reached namely concerning the probability of failure.
- There is a difference between the certification of a particular system and its licensing. The latter concerns mostly the operational permit for the final operational life.
- In particular in chemical engineering the introduction of digital control systems leads to a special situation. The control engineers in this field tend toward the possibility to make changes of their own according to the production process. Can they intervene into the system without inspections from outside? The reason that they do not want anybody to look at the details of their system consists in a certain fear of competition and dissipation of knowledge achieved by their company.
- A special target consists in achieving an approved formal system to transform the specification into a language to be used in the licensing process. Can this be reached in the near future?
- Another area concerns type tested components. How far can they be used in a particular plant without reexercising the whole licensing process?
- The last point addressed languages and standards. It was pointed out that the results of research in computer science often are not accepted by industry. An important area demonstrating this effect consists in high level languages which avoid disadvantages undisputed in the field and at the same time are not easily included by a common agreement of all the companies involved.

Finally a personal remark should conclude this short compilation of highlights of the discussion in the certification group. I had the impression that the participants were very much engaged, demonstrated a deep technical interest into their field, reported frankly about their experience they had gained within their own projects, and were willing to share their ideas in order to come ahead in the whole field of system certification and licensing. I personally learned quite a lot during these discussions.

Report of the Discussion on Development of Safety-Critical Systems

Moderator: John Rushby

Participants: Shuki Bruck, Karl Grosspietsch, Markus Jochim, Diego Latella, Erik Maehle, András Pataricza, Rüdiger Reischuk, Oscar Slotosch, Mark Sujan, Neeraj Suri

Special efforts in the validation of dependable behavior are necessary across the traditional lifecycle phases of safety-critical systems: the requirements engineering phase, various design and implementation phases, the operation phase including maintainance, and finally the decommissioning phase. Depending on the application area, fail-safe behavior (in case of, say, train control) or continuous availability (as is required by flight control, for example) must be validated thoroughly.

During the discussion most participants agreed that the later lifecycle stages are (more or less) adequately controlled. During these phases one concentrates on cost reduction, improvement of assurance, etc. However, dependability improvement is not the main focus. There was some dispute whether the incidence of implementation bugs is small or zero.

It was concluded that the main issues are in the early lifecycle stages where the principal decisions have to be taken with respect to dependability – and other central system properties as well. Therefor, measures are needed to control the usage of models in order to keep the balance among the efforts towards the various design goals.

The problem of requirements validation has been discussed in more detail, because this first phase was identified as critical. Its problems are caused by the transition from the informal to the formal world, and from the continuous to the discrete. More vague pre-requirements must be formulated as concrete requirements. The latter can be checked for internal consistency and relative completeness (you mention what to do if x, then what should happen if not x). However, there is still the problem that global omissions might remain undetected. In particular, dependability properties can fall into this critical category. Some of the extremely numerous potential fault effects and error scenarios might be overlooked. Furthermore, some faults and/or their consequences fall into categories models usually abstract from.

The group discussed several methods for getting pre-requirements and for transforming them into concrete requirements, where different perspectives and viewpoints must be considered. There are some ethnographic techniques like videos, observations and questionnaires. More structured techniques like FMECA, FTA and Hazop are designed to pose questions in a well-defined framework. But these are finally "reviews" and hence imperfect (cf. Mars Polar Lander report). On the other side, formal models are complete in the sense that they model a system and its environment, and generate all possible interactions. However, severe abstractions are usually necessary to make modeling and model evaluation feasible.

A comparison of "aspect-oriented analysis" and "framework-oriented analysis" turned out the following features: Aspect-oriented analysis has appeal: Most dependability issues (fault tolerance, safety, responsiveness) cut accross a traditional functional decomposition. These issues can be "factored out" for the analysis. "Frameworks" like TTA have also appeal: Here, the realization of some aspects can be "factored out". Different levels of abstraction are supported by different frameworks (control diagrams, SCADE, statecharts, TTA). So their integration is an important concern.

Report of the Discussion on Formal Methods

Moderator: Andrea Bondavalli

Participants: G. Csertán, M. Dal Cin, R. Esposito, I. Majzik and I. Marcarelli.

Formal Methods

From a first round of discussion the common feeling of the participants was that we were going to provide questions more than answers, on the role and limitations of formal methods (FM) in the provision and operation of dependability critical systems and what is needed for their success.

Most of the participants have experience on stochastic models and formalisms more than on 'Formal Methods' in the classical sense. Therefore during the discussion very often we tried to trace a parallel between Formal Methods and their stochastic extensions. In the following the items discussed and the questions we formulated with the partial answers proposed are grouped into main subjects: Formal Methods and Languages, Design Process, Issues in model solving.

Formal Methods and Languages

As previously mentioned, many participants are more involved with stochastic models and formalisms than on 'Formal Methods', so the first philosophic question that arose has been what should be intended as "Formal Methods"?

We agreed that any language/method that has a rigorous mathematical foundation is a 'formal method'. Obviously, in such sense, quantitative methods are part of the family.

Next the discussion moved on requirements and, while there was common understanding on 'qualitative requirements' and the way they may be expressed within the various formalisms, we asked ourselves the question on 'how should quantitative requirements be expressed in proper terms?'.

They should be expressed without making unnecessary assumptions that belong to system design (and should be in case made and justified by system designer). E.g. "the system will not fail due to a single fault event" or "there must not be any single point of failure".

Design Process

In the last years many standards for dependability critical fields started asking for the application of formal methods, and many companies adopted them under this pressure. All the participants agreed that this has been a start, but the immediate question arising is: from now on and in the future which conditions have new advanced method to satisfy in order to replace older ones? Under which condition switching from one 'formal method' to another can be afforded in industry? The two main criteria that emerged are cost and clear superiority.

The next topic was the question of when, during development, formal methods should be used. Followed immediately by the question whether we need to distinguish between formal methods and their stochastic extensions. All participants agreed in saying all phases. Moreover some important issues were identified such as the traceability of requirements, the automatic derivation of

test cases, automatic code generation, automatic verification. Regarding stochastic extensions we underlined their generally cheap cost and their especial usefulness in early phases.

Another topic of interesting discussion was about the extent to which quantitative models and formal methods complement each other. The opinion emerged during discussion is that stochastic analysis and formal verification provide partial (non-conflicting) answers to the question: will my system fail? Formal verification can provide answers like: the system will not fail due to logical flown, given some (Boolean) assumptions. Stochastic analysis usually states: eventually you fail with a given probability (making also assumption on the quantitative side).

The discussion went on up to the formulation of the following question. “Will stochastic model checking (and Process Algebras) substitute Markov and Petri net based formalisms?” This received no real answer.

Issues related to solving the models

The capability to reduce the space of states we have to explore appears to be a key issue for FM to develop further. So we start discussing whether this can be achieved and how? The key directions in which work and new results are felt fundamental for formal methods to prove are composition and abstraction...

Composition: Easy/solved (and very much used) for formal languages (that which leads to modularity in specification). Still it appears to be an open problem to obtain separate ‘solution’ of the modules and from these to get the solutions regarding the entire system (compositionality).

Abstraction: Do we have to look for different ways for formal methods and their stochastic extensions? Which mechanisms are to be used? Abstraction appears to have been long practice in mathematics – in stochastic methods is usual to have coarse models, later refined and made more precise. There is also the need of rules and to prove that results converge. Abstraction in FM may prevent you to answer some questions for which the exploration of the (entire?), detailed state space is required. A golden rule appears to adopt a conservative attitude. There is a need of ‘pessimism’. Again here we would need rules! An abstract model is pessimistic if it accounts for more types of and more likely failures than a more detailed model. The absence of failures in the abstract model guarantees the absence of failures also in more detailed models, whereas failures in the abstract model do not imply failures in the more detailed model: we have to analyse the detailed model.