Dagstuhl Seminar

Quantitative Aspects of Designing and Validating Dependable Computing Systems – Calculations, Measurements and Simulations

Organizers:

Klaus Echtle, University of Essen, Germany
Winfried Görke, University of Karlsruhe, Germany
Jean-Claude Laprie, University of Toulouse, France
Winfrid Schneeweiss, University of Hagen, Germany

Increasingly complex computing systems are used for a variety of applications, from world-wide networks to massively parallel computing, from transaction processing to automation of safety-critical systems. The usefulness and success of these applications depends on hardware and software dependability. Achieving them can be rather difficult and may require highest efforts because system complexity is the origin of many design faults, especially in the software, and the required system performance often is in conflict with the redundancy needed for fault tolerance.

Consequently, a careful system design is required. Besides design tools like silicon compilers and CASE, special methods for the prediction and evaluation of system dependability play an important role. In contrast to some former reliabilility calculations dependability evaluation must be able to cope with complex system behaviour. One cannot expect that straight-forward methods are able to cover an extremely large state space or express the behavior of highly interacting subsystems with sufficient accuracy.

This seminar was concentrated on new methods to quantify the dependability of complex computing systems. The usefulness of design and validation techniques for various types of highly dependable systems was assessed and discussed. It clearly turned out that a sufficient dependability evaluation cannot be achieved by just a single method. Both the design process and the system operation must be accompanied by appropriate dependability quantification techniques from the very beginning of the conceptual phase throughout the system lifetime. Moreover, particular efforts are necessary for design fault quantification as well as field data collection and interpretation – appropriate methods must take into account that even extremely rare fault events have an impact on the quantification of highest dependability.

The seminar was subdivided into an *opening* session pointing out the general framework of dependable computing (see page 3) and the following six technical areas:

Opening

A Conceptual Framework for Dependable Computing

Algirdas Avizienis
University of California at Los Angeles, USA
and Vytautas Magnus University, Kaunas, Lithuania

A system is said to be dependable if it can be justifiably trusted to deliver the required service whenever needed. Dependability is a purely qualitative "umbrella" concept that has the quantifiable attributes of availability, reliability, maintainability, safety, integrity, and confidentiality. The fundamental concepts of dependable computing are reviewed and the origin and evolution of the concept of fault tolerance are discussed in this presentation. The presentation is illustrated by the examples of three recent failures to build functioning systems, including the $\$ 5 \cdot 10^9$ (5 billion) AAS (air traffic control) system in the USA.

Practical Needs

Numerical Safety Requirements in Railway Transportation

Bernhard Eschermann
ABB Corporate Research, Baden, Switzerland

For industrial control and protection systems, dependability issues are often important, but usually they are not the only issues to consider. Functionality, cost, delivery time and performance have to be taken into account as well when designing such systems. To find the best tradeoff among all these system characteristics is not an easy task.

In the area of railway transportation, safety is one of the most important system characteristics. One particular subsystem in railway transportation, namely automatic train protection, is used to illustrate the kinds of design constraints that are encountered when specifying numerical dependability goals.

The task of an automatic train protection system is to avoid that a train ever travels faster than it is allowed to. Thus it provides a safety backup for the driver. To achieve this, information about coming speed restrictions (e.g. red signals etc.) is transmitted to a train ahead of time. An on-board computer system in addition gets information about the current speed and the braking characteristics of the train to compute when the train has to start braking. In addition it sends information about the track ahead of the train to the driver and monitors the driver´s attention. It may even completely replace line-side optical signals, so that the driver´s only source of information about speed limits is the automatic train protection system.

The existing on-board systems in Europe were developed independently by different companies together with a particular national railway authority and according to particular national railway regulations. The missing interoperability between different systems requires either changing the locomotive at the border (which is no longer possible for closed train sets like TGV, ICE or X2000) or mounting different systems with separate driver interfaces, brake interfaces, tachometer interfaces etc. Therefore a new common European system shall replace these existing systems. To allow the new system to operate on existing tracks, it will be a modular system with different modules connected by a serial bus. The received track-side information is translated into a standard intermediate format by so-called specific transmission modules for all the existing national systems. Other common modules will use this information to compute the braking curves, interface to the driver etc.

Since the different modules may be delivered by different companies and the complete system has to be put into operation all over Europe, national safety specifications are no longer applicable. Safety regulations should also avoid to specify particular design solutions, since that would impede competition between companies and give an unfair advantage to some companies who already used this particular design solution. Instead the tendency is to specify numerical requirements. Every company then is free to achieve the required safety characteristics by whatever means they think is most cost-effective. A set of common validation procedures should ascertain that the safety targets are actually met.

The goal of this presentation was to show that this numerical approach can lead to certain problems if the overall numerical requirements are split into more and more stringent requirements for smaller subsystems. In addition it does not make sense to require very ambitious safety figures from software subsystems if there is no way to validate them. Actually, specifying numerical requirements might lead to a concentration of design efforts on subsystems like the bus transmission system, whose characteristics are easily quantified, whereas other subsystems (e.g. software), which are not that easily quantified, but which have a much larger impact on system safety, are neglected.

Industrial Demands on Dependability and Realization Aspects

Ernst Schmitter
Siemens AG, Corporate Research and Development, München, Germany

This presentation is motivated by the discussion about different discrepancies of theoretical approaches solving dependability requirements and the transposition of results and experiences into real use and industrial implementation. Looking at the realization aspects industrial demands on dependable systems can be described as following:

- system dependability as results of a lot of single methods and their combinations,
- reduction of development costs and expenses by level-based granularity,
- improvement by interconnection of distributed system components, and
- application-oriented and domain specific implementation of dependability features.

Discussing the development we need more support especially by dependability engineering which includes model-based environment, requirement engineering and application analysis, reference models of dependable system architectures, and new inputs of system technology.

With respect to the topic of this Dagstuhl-Seminar "Quantitative Aspects of Designing and Validating Dependable Computing" the presentation would like to discuss aspects of practical relevance in order to improve the relation between industrial practice and academic objects.

A Design Paradigm for Fault-Tolerant Systems

Algirdas Avizienis
University of California at Los Angeles, USA
and Vytautas Magnus University, Kaunas, Lithuania

A set of guidelines for the design of fault-tolerant systems has been evolving since the original definition of fault tolerance in 1967. The current version of these guidelines, called a design paradigm is presented with special emphasis on the role of quantitative evaluations during the design process. The problem of using "off-the-shelf" subsystems in a fault-tolerant system is discussed and the resulting limitations on the attainable dependability are identified. In conclusion, an analogy of complex fault-tolerant systems and intelligent living organisms in proposed as a means to advance the unterstanding and use of fault tolerance in information processing systems.

Software Dependability

Models for the Evaluation of Software Reliability and Performability Considering Correlation Among Successive Inputs

Andrea Bondavalli
CNUCE/CNR, Pisa, Italy

This work deals with the dependability of programs of iterative nature. The dependability of software structures is usually analysed using models that are strongly limited in their realism by the assumption made to obtain mathematically tractable models and by the lack of experimental data. Among the assumptions made, the independence between outcomes of successive executions, which is often false, may lead to significant deviations of the results obtained from the real behaviour of the program under analysis. Experimental and theoretical justifications show the existence of contiguous failure regions in the program input space and that, for many applications, the inputs often follow a trajectory of contiguous points in the input space.

In this work we consider dependencies among input values of successive iterations and the possibility that repeated, non-fatal failures may together cause mission failure in studying the dependability of iterative software. Two models have been defined, the first uses as parameters the state transition probabilities and allows to appreciate the effects of different distributions for the correlation. The second uses steady-state probabilities obtained by testing the system in an environment slightly different from the one intended for real operations, i.e. without considering sequences of benign failures and that missions are of fixed length duration. We evaluate the effects of these different hypotheses on 1) the probability of completing a fixed-duration mission, an 2) a performability measure.

Modeling Computer Systems Evolutions: Non-Stationary Processes and Stochastic Petri Nets – Application to Dependability Growth[1)]

Jean-Claude Laprie, Mohamed Kaâniche, Karama Kanoun
LAAS - CNRS, Toulouse, France

[1)] Invited paper to PNPM´95 – the 6th International Workshop on Petri Nets and Performance Models, Durham, North Carolina, 1995; proceedings: pp. 221 – 230

Stochastic Petri nets (SPNs) have emerged over the years as a favored approach for performance and dependability modeling and evaluation. Their usual utilization assumes that systems specification and design do not evolve, in opposition to real-life. This paper is aimed at a preliminary exploration of how to take advantage of the existing body of results on SPNs for modeling the evolution of computer systems, i.e. to model non-stationary stochastic processes. It focuses on dependability evolutions which result from successive releases.

The paper is composed of three sections. In the first section, the behavior of atomic systems is characterized and several elementary SPNs are proposed for modeling evolutions via multi-stage homogeneous Markov processes, multi-stage semi-Markov processes, and non homogeneous Markov processes. The second section addresses the utilization of the SPNs proposed in the first section in the modeling of multi-component systems, i.e. when accounting for the system structure. Finally, the third section summarizes the approach reported in [1], which is based on the simulation of reliability growth via an hyperexponential phase-type expansion.

[1] J.-C. Laprie, K. Kanoun, C. Béounes and M. Kaâniche, "The KAT (Knowledge-Action-Transformation) Approach to the Modeling and Evaluation of Reliability and Availability Growth", IEEE Trans. Software Engineering, SE-17 (4), pp. 370 – 82, 1991.

An Approach to Developping and Evaluation of Fault-Tolerant Systems

Fevzi Belli
University of Paderborn, Germany

An approach will be introduced to systematically integrate fault tolerance properties and validation aspects into the design and implementation of complex systems, especially software systems. This is achieved by exploiting a formal specification of the system where the amount of necessary redundancy can be determined. The system description is based thereby on a combination of a predicate/transitions nets with regular expressions.

The net model provides a formal – but nevertheless lucid – overview of the system behavior in general, supporting the correct unterstanding of potential concurrency in the system processes. Regular expressions are used to model the sequential behavior of single-system components in detail. Both model layers provide well-defined levels of error detection, the regular expressions enable the system designer to also determine and introduce redundancy to achieve error correction.

The implementation will be materialized through automatically mapping the system specification, i.e. the predicate/transition model into a corresponding logic program, conventionally into parallel PROLOG. The validation and evaluation [1,□2] of the imple mented system can be carried out by means of test cases which will be automatically generated using the regular expressions that model the system behavior in detail [3].

The approach will be illuminated by a case study which contains a stepwise refined specification and analysis of a multistorey shelving system model which has been implemented following the method described here [4].

References:

[1] F. Belli, O. Jach, "An Implementation-Based Analysis and Testing of Prolog Programs", Proc. International Symposium on Software Testing and Analysis, ACM Press (1993), pp. 70 – 80.

[2] A. Azem, F. Belli, P. Jedrzejowicz, "Relability Prediction and Estimation of Prolog-Programs", IEEE Trans. Reliability (1994), pp. 542 – 549.

[3] F. Belli, J. Meyer, "System Specification, Analysis and Validation by Means of Tined Predicate/Transition Nets on Logic Programming", Proc. International Symposium on Software Reliability Engineering, IEEE Computer Press (1995), pp. 68 –□77.

[4] F. Belli, K.-E. Großpietsch, "Specification of Fault-Tolerant-System Issues by Predicate/Transitions Nets and Regular Expressions – Approach and Case Study“, IEEE Trans. Software Engineering (1991), pp. 513 –□526.

Dependability Modeling

Dependability Evaluation in the Industrial Practice

Ernst Schmitter
Siemens AG, Corporate Research and Development, München, Germany

This contribution discusses industrial aspects of system evaluation based on a methodology which places at system developer's disposal methods and tools to establish the relation between measures as a quantitative representation of the evaluation criteria and different parameters of system architecture and of workload, especially the status-quo of dependability evaluation. Recently, the use of stochastic Petri Nets of various types has been recognized as a useful modeling approach solving the requirements of performance and dependability evaluation of computer systems, communication networks, and production systems.

As an example the development of PENPET (PEtri Net based Performability Evaluation Tool) based on the well-known TOMSPIN (TOol for Modeling based on Stochastic PetrI Nets) supports system designer during the whole performability evaluation process. Different use of the tool is presented describing applications of the industrial practice, and also some limitations of modeling by Petri Nets.

Nevertheless, stochastic Petri Nets improve efficient methods for description of questionings as to systems architectures and evaluation methodology which is necessary for suffient dependability engineering.

Computable Dependability Bounds for Large Markov Models

Pierre-Jacques Courtois
University of Louvain-la-Neuve, Belgium

This presentation introduced a new technique which can be used to efficiently design and analyze large Markov models. Lower and upper bounds can be computed for reliability and dependability measures of fault-tolerant systems. By large models, were meant models that may have tens of millions of states. The method dispenses from generating the whole state space, and is specially suited, and even indispensable, for those state spaces that are too large to use classical procedures.

The technique is based on decomposition an aggregation, and integrates new concepts and properties of eigenvector polyhedra. For each subset of states considered in isolation, these

properties make possible to determine lower and upper bounds on the relative steady state values of the variables and probabilities associated with each state of that subset. Besides, these bounds are proven to be the tightest ones that can be obtained, given the part of the state space that is considered. Locality or near decomposability are important factors contributing to tightness.

A model of a repairable fault-tolerant system with more than 16 millions of states was used as an example to illustrate the method. Tight bounds on its availability were obtained by considering small fractions (less than one percent) of the total state space. The method is potentially useful to bound other types of dependability requirements. The complexity of the method was analyzed on this example model, and shown to be at most square in the size of the largest subset being considered. Further possible reductions in complexity were discussed.

References:

[1] P.-J. Courtois and P. Semal, "Bounds for the Positive Eigenvectors of Non-Negative Matrices and for their Approximations by Decomposition", J. ACM 31 (4), pp. 804 – 825, 1984.

[2] P.-J. Courtois and P. Semal, "Computable Dependability Bounds for Large Markov Chains", in Predictably Dependable Computing Systems, B. Randell and others, Ed. Springer, 1995.

A Tighter Upper Bound for System Unavailability

Winfrid Schneeweiss
University of Hagen, Germany

It was shown that – trading computational simplicity for finding a true optimum – one can find a tighter upper bound for the probability of the union of m random events $a_1, \ldots, a_m$ than the one given by the 1st Bonferroni inequality (= Hamming´s inequality). First, the 1967 theorem by Hunter was invoked, saying that

$$P\left(\bigcup_{i=1}^{m} a_i\right) \le \sum_{i=1}^{m} P(a_i) - \sum_{(i,j)\in I} P(a_i \cap a_j) \qquad (1)$$

where I is the set of sets of m – 1 pairs of (i, j) each, which define, via the associated edges (v_i, v_j), a spanning tree in K_m, the complete graph with $|V| = m$. The practical problem is that in order to find the best (tightest) bound of the type (1) one has to solve an optimization problem, identifying (one of) the best of m^{m-2} (Cayley´s result) trees.

The (little) contribution of this report is that with almost no computational overhead above the complexity of determining all the $P(a_i \cap a_j)$ for the 2nd Bonferroni inequality (needed

anyway to get a lower bound, too) one can find a good second sum in (1) by taking the best sum out of a set of (m − 1)! of them. To be more precise, the following "receipt" is shown to make sense:

1) Determine the $P_{i,j} := P(a_i \cap a_j)$ linewise for fixed i and j running from 1 to i − 1 (lower half-matrix of the $P_{i,j}$).
2) Determine the maximum $P_{i,j}$ in each line.
3) Add all the m − 1 $P_{i,j}$ and use this sum as the second sum in (1).

It can be shown (not done so in the talk) that the $P_{i,j}$s as found above really have index pairs as in Hunter´s 1967 theorem. An example was presented and the audience was told (it´s really easily verified) that in the 2-out-of-3 system case incidentially the above tighter upper bound is the exact value.

Using Standard Fault Tree Analysis in Cases of s-Dependent Components

Winfrid Schneeweiss
University of Hagen, Germany

It was first pointed out that the fault tree is only the peak of an iceberg of a complex dependability problem, especially in that dependencies between the binary indicator processes of the system components are determined elsewhere. However, the fault tree´s Boolean function is correct and – as is shown below – very helpful also in cases of s-dependent components indicators X_i; i = 1, ... , n for components.

In order to warm the audience up, it was quickly reviewed how standard fault tree analysis can proceed efficiently these days, especially when using the Shannon decomposition for getting rid of the Boolean operators of the fault tree´s function φ, since unavailability U_S (for a system S) is by definition $P(\varphi = 1) = E(\varphi)$ and the latter is best determined from a "usual algebra" form of φ and not a Boolean algebra one.

In the main part of the presentation it was shown via a few small-scale examples that the different terms T_k of a pseudo-polynomic form of φ can be very helpful when finding the $E(T_k)$ also in cases, where factoring of $E(T_k)$ is not allowed. The two main example cases were that of

1) cold standby, where e. g. $X_i \cdot X_j$ leads to $\int_0^t f_{L_i}(\varepsilon) \cdot F_{L_j}(t-\varepsilon)\, d\varepsilon$ instead of $U_i \cdot U_j$;
2) a system with a 1-out-of-2 (or duplex) Markov-type subsystem where in the stationary case $X_1 \cdot X_2$ leads to

$$E(X_1 \cdot X_2) = \frac{a\lambda^2}{a\lambda^2 + a\lambda\mu + \mu^2}$$

when $a\lambda^2$ is the (failure) rate from the very best system state to the intermediate state with one component "good" and the other one "bad", and $b\mu$ is the repair rate from the system failure state to the above intermediate state.

Testing

Hierarchical Test Synthesis for Digital Systems Using Alternative Graph Model

Raimund Ubar
Technical University of Tallinn, Estonia

A uniform approach to create tools for hierarchical mixed-level test pattern generation for digital systems has been developed based on alternative graphs (AG). Logical level AGs represent in a compressed form the topology of gate-level circuits, and therefore, unlike to the analogical binary decision diagrams (BDD), they directly support test generation for gate-level structural faults without representing them explicitly. BDDs do not represent the topology and therefore they can be used only for generating functional tests with uncertain quality of detecting structural faults. On the other hand, in general, AGs support uniform approach to digital test design at different system levels whereas BDDs support only the Boolean level.

Testing and diagnosis of digital electronics systems has faced with a lot of problems produced mainly by the complexity of systems. One solution is to treat systems under test hierarchically reducing in such a way the complexity of the task. A general theory for diagnosing hierarchically represented systems by uniform methods on different levels is missing. Alternative graphs could be a reasonable way of representing diagnostic information of digital systems uniformly on different levels of a system. They can be regarded simultaneously as a procedural notation (a program), or as a data structure (decision tree) to be processed, or as a representation form of diagnostic knowledge of the system, or as a compact way to represent all possible test modes for the system. The homogenity of the model makes the backtracking procedure very easy, in fact, AG itself can be regarded as a search tree.

Different ways of synthesis of AGs bring along different features which can be simultaneously exploited supported by the same form of the model. For example, SAGs can serve as gate- or macro-level structural representations, FAGs (or BDDs) afford an effective way for functional however still logical level description, whereas higher-level AGs support procedural, register transfer or behaviour aspects in system representation. Experiments have shown the advantage of using topological ATPG based on SAGs compared to using the gate-level topology. A dynamic combination of FAGs and SAGs can contribute to more efficient test generation or fault simulation for large digital circuits. For the new promising trend in ATPG area of combining symbolic techniques based on BDDs and traditional topological algorithms, the AG approach is providing a uniform theoretical basis. In such