SAND99-0231C

MULTI-DOMAIN SURETY MODELING AND ANALYSIS FOR HIGH ASSURANCE SYSTEMS

James Davis, Jason Scott, Janos Sztipanovits
Institute for Software Integrated Systems /
Vanderbilt University
Box 1826 Station B
Nashville, Tennessee 37235 USA
bubba@isis.vanderbilt.edu

Marcus Martinez
Electromechanical Engineering Department
Sandia National Laboratories
1515 Eubank SE
Albuquerque, New Mexico 87123
martimj@sandia.gov

ABSTRACT

Engineering systems are becoming increasingly complex as state of the art technologies are incorporated into designs. Surety modeling and analysis is an emerging science that permits an engineer to qualitatively and quantitatively predict and assess the completeness and predictability of a design. Surety is a term often used in the Department of Defense (DoD) and Department of Energy (DOE) communities, which refers to the integration of safety, security, reliability and performance aspects of design. Current risk assessment technologies for analyzing complex systems fail to adequately describe the problem, thus making assessment fragmented and non-integrated. To address this problem, we have developed a methodology and extensible software toolset to address model integration and complexity for high consequence systems. The MultiGraph Architecture (MGA) facilitates multi-domain, model-integrated modeling and analyses of complex, high-assurance systems. The MGA modeling environment allows the engineer to customize the modeling environment to match a design paradigm representative of the actual design. Previous modeling tools have a predefined model space that forces the modeler to work in less than optimal environments. Current approaches force the problem to be bounded and constrained by requirements of the modeling tool and not the actual design problem. In some small cases, this is only marginally adequate. The MGA facilitates the implementation of a surety methodology, which is used to represent high assurance systems with respect to safety and reliability. Formal mathematical models are used to correctly describe design safety and reliability functionality and behavior. The functional and behavioral representations of the design are then analyzed using commercial-off-the-shelf (COTS) tools.

INTRODUCTION

The current high consequence system design environment is highly fragmented. The disciplines of safety, reliability, performance and security are typically considered in isolated scenarios by organizations separated physically and philosophically. This can result in highly suspect complex systems, which have a tendency to perform below design expectations and fail in unanticipated scenarios. Methods for inter-relating safety, reliability, performance, and security models are needed to ensure a complex design will meet system requirements.

The principle tools used in elements of surety analysis consist of fault tree analyses, failure mode and effects analysis, barrier analysis, adversarial analysis, and some form of global risk analysis. Under some toolset modeling environments, state space representations are used to capture the reactive nature of the system under consideration. There appears to be a consensus that state space descriptions are currently the best technology for dealing with complex reactive systems. See [1] for more information on state space descriptions.

Each of the above techniques has varying degrees of utility depending on the application and expertise of the systems engineer, component designer or analyst. Identifying a technique's strengths and employing these methodology fragments in a hybrid structure to complex design problems has the potential of solving inconsistencies in complex design problems. In the area of Surety at Sandia, the current technologies being applied have proven difficult in solving the complex predictive problems that will enable an engineer to certify a design solution.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

This paper discusses a new approach for integrated safety and reliability analysis using Model-Integrated Computing principles and tools. The essence of the approach is to perform system modeling using a modeling environment that allows an integrated, consistent representation of system models. This integrated model is translated into the input languages of COTS analysis tools, thereby maintaining the consistency among the tool-specific models.

BACKGROUND

Several key technologies necessary for representing and analyzing integrated system surety are examined in this paper. Sandia's paradigm for surety and methods for modeling high assurance systems is discussed. Model Integrated Computing, a technology for software integrated modeling environments, will be introduced. Ordered Binary Decision Diagrams (OBDD-s) are incorporated into the toolset to enable mathematical representation of surety models.

Sandia's Paradigm of Surety

Paradigms for surety involve the design and development of complex systems whose failure can result in significant loss of human life or corporate resources. Within the Department of Energy (DOE) complex, high consequence operations comprise the design and manufacture of nuclear weapon systems. The issues regarding the high consequence aspect are directed toward the inadvertent detonation, intentional or non-intentional, the dispersal of nuclear materials, or the loss of system control. These events will have significant environmental and political impacts as well as the potential loss of human life. As a result, significant effort is expended to ensure acceptable system behavior is achieved under all circumstances.

Surety constitutes the integrated consideration of safety, security, reliability, and performance throughout the system life cycle. Security is comprised of two basic sub-elements: physical security and functional security, sometimes called use- control in the weapons communities. Reliability is achieving a high probability of successful operation under normal environments. Safety is preventing accidental nuclear detonation or dispersal of nuclear material under abnormal environments. The elements of surety can be applied to a broad spectrum of design activities including, but not limited to, weapon systems, national infrastructures, banking, chemical processing and biological technology. Surety concepts apply to any system designed to operate and perform high consequence actions.

Modeling of Surety Systems

Current surety designs represent the safety, reliability, performance and security of a system as disjoint, separate models and analyses. Separate organizations are responsible for evaluating and reporting the safety, reliability, and security of the system. These organizations are somewhat disjoint through out the product life-cycle process. Each organization has their preferred modeling and analysis techniques and applies these techniques for system verification and validation. Figure 1 represents the system surety engineering process [2].

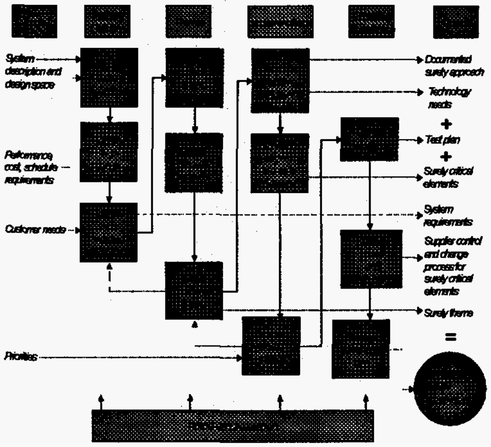
System description and design space
Performance, cost, schedule requirements
Customer needs
Priorities
Documented surety approach
Technology needs
+
Test plan
+
Surety critical elements
System requirements
Supplier control and change process for surety critical elements
Surety theme
=

Figure 1. The System Surety Process

Initially, small organizational entities receive problem input and define the problem. Subsequently, designers, manufacturing engineers, quality engineers etc. interact to build separate models and utilize different tools to design and manufacture the system. However, the models defined by each organization of the system are often disjoint and do not represent the same system. Utilizing a Model-Integrated approach for model construction allows for modification of the integrated system model in one aspect, which *should* affect the other aspects [3].

An *integrated model* for surety systems is based on *system* models. Safety, reliability and fault views of the models can be abstracted from a single common integrated model. Using an integrated approach, system level changes will show up in the other system model views *if* the modification affects the specific view.

A key element of integrated modeling involves the interaction of different modeling aspects. Previously these areas of overlap had to be dealt with manually. By using an integrated approach, changing the model in one aspect may affect many different aspects of the model. Instead of requiring separate safety and reliability models of a system, the system can be modeled in a manner that is more natural and intuitive to system designers and analysts. This system model is then augmented with safety and reliability characteristics. For example, the system may be modeled using a behavior model. The safety and reliability information about the system could then be incorporated to reflect specific behavioral traits of the system. When performing a safety or reliability analysis, the necessary fault information can be extracted from the augmented behavioral model and used to perform the necessary analysis.

Model-Integrated Computing

In *Model-integrated computing*, integrated, multiple-view, domain-specific models capture information relevant to the system under design. Models explicitly represent the designer's understanding of an entire system, including the information-processing architecture, physical architecture, and operating environment. Integrated modeling explicitly represents dependencies and constraints among various modeling aspects.

The Multigraph Architecture (MGA) is an infrastructure for model-integrated computing and is described in detail by Sztipanovits [4]. The integrated environment includes Modeling Tools, an Integrated Model Database, Analysis Tools, and Application Synthesis Tools. The Analysis Tools work with tool-specific analysis models; the applications are specified in terms of executable models. The modeling paradigm of the analysis tools and the executable models are domain independent. In a given domain, the relevant information about the design artifact is captured by a multiple-view, domain specific modeling paradigm. Key components of the model server are the "Model Interpreters". The role of the Model Interpreters is to translate the domain specific model into the analysis models for the tools and the executable models of applications to be synthesized. This architecture allows that the analysis and synthesis tools to share design information that is common without requiring that the tools use the same modeling paradigm.

An integrated tool environment is built in the following steps using the MGA infrastructure:

1. Systems and domain experts conduct domain analysis and specify an integrated modeling paradigm, which is designed to capture key aspects of the system. The modeling paradigm is comprised of the concepts, relationships, model composition principles and constraints that are specific to the domain.
2. Using the formal representation of modeling paradigms, systems and domain experts specify and create a domain specific model building, model analysis, and software/system synthesis (model integrated program synthesis) environment. The environment includes reusable domain specific components, general building blocks, domain specific model analysis tools, and software synthesis tools. Completion of this step is supported by MGA meta-tools.
3. Within the modeling environment framework, domain and application engineers build integrated multiple view models of systems to be designed and implemented. The multiple view models typically include requirement and design models.
4. Domain and application engineers analyze the models according to the nature and needs of the domain. The domain specific models are translated into the input languages or input data structures of the selected analysis tools. MGA model interpreters complete the model translation.

Multi-Domain Modeling

High consequence, high assurance engineering design and development is a complex process that often incorporates diverse, often conflicting requirements, new technologies, and involves many diverse disciplines. System engineers must identify objectives and requirements and formulate metrics that can be used by the design teams to assess the viability of the concepts in satisfying the design and development objectives.

The *model-integrated computing* approach has the ability to incorporate strengths from various modeling and analytical techniques and employs methodology fragments in a hybrid structure to solve complex design problems. In the specific problem domain of surety, the current technologies being applied in a non-integrated fashion cannot solve the complex predictive problems that will enable a designer to certify a design solution. System certification is crucial in the design of High Consequence systems. The approach taken with *model-integrated computing* is to take the strengths of a number of analytical techniques and define/develop an integrated approach that surpasses current approaches and also provides a venue for inclusion of new technologies that can be incorporated into the MGA framework and tools.

Integrated Safety and Reliability

Integrating safety and reliability addresses both complex design and coupled modeling simulation. To accomplish these objectives, formal languages [1] representative of the problem and solution domains are incorporated to specify all functions and relationships for the specific domains (e.g. reliability, safety).

The objective of reliability modeling and analysis is to represent the major functions of the design in terms of expected and desired sub-system and component behaviors. This process is referred to as modeling and the usual result is a diagrammatic representation of the inter-relating component behaviors and a corresponding set of "reliability mathematical equations". Assumptions affect the accuracy of the mathematical equation and its evaluation. Successful design functions require successful operation of all events modeled. Single objects represent some operations, while others have two or more objects - any of which can provide the needed operation. These functional relationships lead to a mathematical expression relating design failure probability to component behavior failure probabilities.

Safety modeling and analysis must address external and internal events which, when subjected to a design, can lead to unsafe operation or conditions. Safe design is directed toward minimizing non-engineered or poorly engineered hazard controls. Safety modeling is an extension of reliability modeling and includes an assessment of how frequently an excursion from the design results in a hazard. The analysis is extended to a more formal manner to include consideration of event sequences, which transform the hazard into an accident.

Integrating safety and reliability approaches under the framework of the MGA requires safety and reliability domain experts to possess and maintain in-depth knowledge of individual sub-systems and components used in the problem domain (the system being designed) that affect the solution domain. It is the responsibility of the domain experts to formalize the design under a common formal language. The use of a common formal language suitable for integrated modeling and analysis allows the synthesis of the multi-domain problem structure to be synthesized into singular aspect domain model structures. It is the singular aspect domain structures that allow domain experts to perform specific analyses in the area of concern. This methodology allows both complexity and coupled model simulation issues to be addressed.

Ordered Binary Decision Diagrams (OBDD)

Safety and reliability analyses use discrete models and operations over finite domains. The most general difficulty in all of the analysis techniques is the size of the state space in large-scale systems. Combinatorial explosion is the result of the exponential increase in the number of discrete elements (states, events, hypotheses, etc.) during operations, which eventually makes access to the individual elements unfeasible. By introducing a binary encoding for the elements, the individual elements, and sets of the elements, the relations among them can be expressed as Boolean functions. Using Boolean function representations, we can express operations and algorithms in diagnosis and safety analysis in symbolic form, by means of symbolic Boolean function manipulations.

OBDDs provide a symbolic representation for Boolean functions in the form of directed acyclic graphs. [5] They are a restricted, canonical form version of Binary Decision Diagrams (BDD). [6] Bryant [7] described a set of algorithms that implement operations on Boolean functions as graph algorithms on OBDDs. Taking advantage of the efficient symbolic manipulations, researchers have solved a wide range of problems in hardware verification, testing, real-time systems, and mathematical logic using OBDDs that would have been otherwise impossible due to combinatorial explosion. Symbolic model checking is extensively used in hardware design (see, e.g., [8]), and has shown to be efficient in state space sizes 10^{120} and beyond.

MODELING OF HIGH ASSURANCE, HIGH CONSEQUENCE SYSTEMS

An integrated model for high assurance, high consequence systems is based on system behavior models and system hardware models. Our work has focused on integrating the safety and reliability aspects of surety. Future work will entail adding the security and performance surety aspects to the integrated toolset. Safety and reliability views of the models can be abstracted from the integrated model. Using an integrated approach, system level changes will show up in the other system model views only if the modification affects the specific view.

A key element of integrated modeling involves the interaction of different modeling aspects. Previously these areas of overlap had to be dealt with manually. By using an integrated approach, changing the model in one aspect may affect many different views of the model. The system can be modeled in a more natural format. Instead of requiring safety and reliability models of a system, the system can be modeled in a manner that is more natural to system designers. This integrated system model is then augmented with safety and reliability features. For example, the system may be modeled using a model describing the behavior of the system in terms of safety and reliability. The safety and reliability information about the system could then be attached to specific behavioral traits of the system. When performing a safety or reliability analysis, the specific information of interest can be extracted from the augmented behavioral model.

Safety and reliability are not separate, independent traits of a system. Instead, both safety and reliability are functions of a system's components, how the components are assembled, how the components can fail, and the system's environment.

SELECTION OF DOMAIN-SPECIFIC MODELING PARADIGM

Safety and reliability analysis algorithms work with a "model" (a suitable representation) of the system. The required depth of the analyses determines the level of detail in the models.

Models for Safety Analysis. Safety analysis requires the development of models that represent the relationship between failure modes (or fault events) of physical components and discrepancies (or discrepancy events) in the high-level behavior of the system. Taking into consideration of the characteristics of the high impact system category (complexity, dynamic behavior), we selected the following model organization:

- The Behavioral Model represents the system behavior in the discrete state space in terms of hierarchical, parallel state machines. The Behavioral Model includes both functional and fault behaviors by representing functional and fault states, and transitions among these states triggered by input, local, and fault events. We have selected the StateChart notation [1] for behavior modeling because the StateChart models are expressive, scalable, and support incremental modeling.
- The Physical Model captures the component hierarchy of the system. The physical components are modeled as component assemblies and sub-assemblies. Each physical component has a fault model view. The fault model view lists the physically possible and functionally meaningful fault modes of the components.
- The interdependencies between the Behavioral Model and Physical Model are represented in the form of references between these models.

Explicit representation of the interdependencies between behavioral models and physical models is a critical element of the integrated modeling paradigm. It guides the model builder to understand their relationship, and enforces the systematic analysis of the effects of the fault modes of components.

Models for Reliability Analysis. The analysis environment includes a reliability analysis tool, WinR® [9], utilizes fault trees for system model representation. The fault tree represents the logical relationship between a top event and fault modes in the form of an AND-OR tree. Utilizing the fault tree models, and the failure rate information of the components, the tool calculates the expected rate for the occurrence of the selected critical system state defined by the top event.

The models for reliability analysis have strong overlap with the models for safety analysis and fault analysis. The most important relationships regarding reliability analysis are the following:

- The top event in reliability analysis corresponds to a transition into a selected critical system state, which is modeled as part of the behavioral models.
- The fault events correspond to fault modes of components that are contained in the physical models.
- The fault tree can be derived from the set of all possible state trajectories that lead to the selected critical system state. These trajectories are fully defined by the behavioral models.

The conclusion is that the behavioral and physical models contain all the information required for reliability analysis except failure rate data for the component fault modes. Therefore, by extending the component fault models with probabilistic information, the modeling paradigm will allow safety, diagnosability and reliability analysis from the same model set. It is important to note that the relationship between the fault tree models required by the reliability analysis tool and the behavioral models is quite complicated.

FORMAL MODEL FOR INTEGRATED ANALYSIS

The role of a formal model is to give a domain independent, mathematical specification for the models. The selected domain-specific form of the Behavioral Model is the StateChart notation. While StateCharts are convenient for building large-scale, parallel state machine specifications, the analysis algorithms require a formal mathematical model, which captures the precise semantics of the hierarchical, parallel state machines. We use Discrete Event System (DES) models for this purpose.

A Boolean representation of the DES model can be created. [10] The Boolean representation of the DES model can be directly translated into an OBDD form, allowing the symbolic evaluation of the analysis algorithms. See Figure 2.

INTEGRATED ANALYSIS WITH OBDD-s

The primary difficulty with safety and reliability analysis with state space modeling representations is combinatorial state space explosion. For example, the generation of a fault tree from the behavioral model requires the exhaustive enumeration of all possible state trajectories that may lead from an initial state (or a set of possible initial states) to a critical state under all fault conditions. By representing the Behavioral Model symbolically as an OBDD, the required calculations can be completed symbolically without explicitly enumerating the exponential number of alternatives.

The application of OBDD-s for the analysis requires the following steps:

1. Mapping the Behavioral Models into OBDD-s: This step is completed automatically. In accordance to the general framework of the Multigraph Architecture (MGA), the StateChart models in the Model Database are traversed by a Model Interpreter, which selects a binary encoding for the sates and incrementally builds up the OBDD representation for the relational model.

2. Safety analysis: The safety analysis tool receives the OBDD representation of the Behavioral Model and performs forward reachability analysis on the state machine. Given a set of initial states S_0, reachability analysis calculates the set of reachable states $S^*(S_0)$ under all possible combination of $x \in X$ input events, $f_S \in F_S$ and $f_I \in F_I$ fault events. The goal of the safety analysis is to show that selected critical events are not elements of the reachability set. The reachability set is calculated symbolically, therefore the analysis is feasible even for very large state spaces.

3. Reliability analysis: As it was mentioned above, the reliability analysis tool, WinR®, expects a fault tree that represents all possible combinations of fault events leading to a selected top event. The analysis algorithm generates all of the state trajectories leading to the top event using backward propagation, and simultaneously builds up the logic relationship between the fault events and the top event.

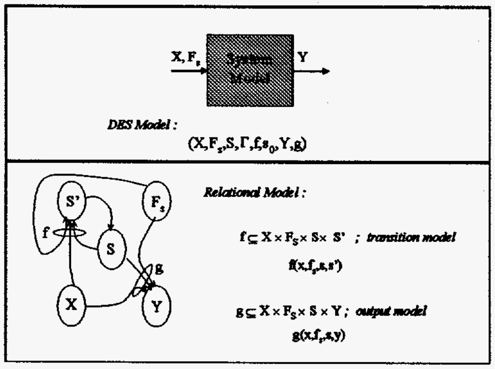
X, F$_s$
System Model
Y
DES Model :
(X,F$_s$,S,Γ,f,s$_0$,Y,g)
S'
F$_s$
f
S
g
X
Y
Relational Model :
f ⊆ X × F$_S$ × S × S' ; transition model
f(x,f$_s$,s,s')
g ⊆ X × F$_S$ × S × Y ; output model
g(x,f$_s$,s,y)

Figure 2: DES and relational models for dynamic systems

SCALING ISSUES WITH OBDD-BASED MODELS

The first approach to mapping the behavioral models into OBDD-s involved creating one OBDD to represent the transition relation for the entire behavioral model (both hardware and system behaviors). Upon building models with a total design space of 2^{40}, the monolithic transition relation grew too large to compute. Performance of the analysis tools was unacceptable once the transition relation had to be swapped to virtual memory.

A distributed method for computing the transition relation was developed. The transition relation is stored in a tree structure that mimics the structure of the behavioral models. The transition relation nodes corresponding to leaf states in the behavioral model have OBDD-s to represent the transitions leaving the corresponding state in the behavioral model. Nodes representing parallel states use a conjunction of their children's transition OBDD-s for their transition relation. Nodes that represent sequential states use a disjunction of their children's transition OBDD-s for their transition relation.

With this new approach, models consisting of a total design space size of 2^{85} have been analyzed. Further work on the scaling of OBDD based models is still needed to better understand when scaling problems will arise.

MODELING AND ANALYSIS TOOL ARCHITECTURE

The Model-Integrated Safety and Reliability Analysis tool architecture is an instance of the generic architecture of Model-Integrated Computing Environments discussed before.

The domain specific models are built by the Metaprogrammable Visual Model Builder, and are stored in the Model Server. The constraints defined in the meta-language representation of the modeling paradigm. The capture constraints are enforced by the Visual Model Builder and allow the user to create only valid models.

There is a separate model interpreter for each analysis tool. The model interpreters traverse the domain specific models and collect/translate the information into the required input data structures of the tools. This solution enables the reuse of the tools even if the domain specific modeling paradigm is changing.

The WinR reliability analysis tool is an 'external' component in the tool architecture. It is important to note that the WinR® tool has a separate model building interface, therefore the tool can be used independently from the integrated environment. The advantage of using WinR® in the configuration above is that the overlapping modeling views are kept consistent by the integrated modeling environment.

EXAMPLE

The example system is a simplified version of an automotive braking system. Different sets discrete failures trigger the transitions between states. A hardware failure can lead to other failures in the system. For example, if the front brake line ruptures, the front brake cylinders will become non-operational. Then the brake shoes cannot contact the brake rotors, so the front brakes have failed. When the system is analyzed, these separate hardware state machines are analyzed as if they are parallel components of the same FSM.

This event tree only contains Boolean AND, Boolean OR, and the Boolean encodings for the failure events. This event tree can be exported to WinR® for fault tree analysis (the nodes of the event tree correspond to component failures). Probabilistic and cut set information about the system is then assessed with WinR®

For our example, the number of failure trajectories is quite large compared to the size of the system's Behavioral models. The fault tree generation algorithm examines over 3 million trajectories for this small example. The simplified fault tree contains approximately 40 nodes. Even for this limited example, the number of failure trajectories would be difficult to discover manually.

SUMMARY

Integrated surety analysis is a difficult problem for two primary reasons. First, the models to be used in these analyses are not independent from each other. Therefore guaranteeing the consistency of the analysis results is a major concern. Second, the generally used discrete, finite state modeling techniques require analysis methods that are plagued by combinatorial explosion of the state and event sets. The described model-integrated modeling and analysis environment and the described analysis methods address both problems. The introduction of an integrated modeling paradigm allows the construction of models that are domain specific, and consistent for each analysis task. The problem of combinatorial explosion is mitigated by the use of relational models and OBDD representations. Although symbolic manipulations offer tremendous advantages in the analysis of large-scale systems, scalability remains an important issue in analyzing these systems. Our experience with the analysis of a variety of systems has shown the feasibility of the approach.

Future work must address analyzing systems with regard to unintended consequences. Additionally, the scalability of the described techniques must be examined. When adding a new type of analysis to the desired analysis packages, the modeling environment can change. How this affects the desired analyses is unknown.

ACKNOWLEDGMENTS

Support for this project comes from Sandia National Laboratories and the Defense Advanced Research Projects Agency, Information Technology Office, under contract #F30602-96-0227. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract De-AC04-94AL000.

REFERENCES

[1] Harel, D., "StateCharts: A Visual Formalism For Complex Systems", *Science of Computer Programming 8*, pp. 231-278, 1987.

[2] D'Antonio, P., Werner, P., Covan, J., "High Consequence Systems Engineering," Sandia National Laboratories Technical Report, Sandia National Laboratories, October 17, 1997.

[3] Misra, A, et al, "Diagnosability of Dynamical Systems," *Proc. of the Third International Workshop on Principles of Diagnosis*, pp. 239-244, Rosario, WA 1992.

[4] Sztipanovits, et al., "MULTIGRAPH: An Architecture for Model-Integrated Computing" *Proc. of the IEEE ICECCS'95* Ft. Lauderdale, Florida, Nov. 6-10. 1995.

[5] Bryant, R. E., "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, C-35(8), 1986.

[6] Lee, C. Y., "Representation of Switching Circuits by Binary Decision Programs," *Bell System technical Journal* pp. 985-999, 1959.

[7] Bryant, R. E., "Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams", *Technical Report CMU-CS-92-160*, School of Computer Science, Carnegie Mellon University, June 1992.

[8] Burch, J. R., Clarke, E.M., Long, D. E., "Symbolic Model Checking for Sequential Circuit Verification," Technical Report, CMU-CS-93-211, Carnegie Mellon University, July 15, 1993.

[9] Sandia National Laboratories, *WinR Reliability Analysis Software*, Systems Reliability Department, 1996.

[10] Davis, J., et al, "Integrated Analysis Environment for High Impact Systems", *Proc. of Engineering of Computer Based Systems*, pp. 218-225, Jerusalem, Israel, April 1998.

[11] Leveson, N.: Safeware: System Safety and Computers, Addison Wesley, New York, 1995.