IFIP WG 10.4 Meeting
June 29, 2014

Duke High Availability Assurance Laboratory (DHAAL)

Kishor Trivedi
Dept. of Electrical & Computer Engineering
Duke University
Email: ktrivedi@duke.edu
URL: www.ee.duke.edu/~ktrivedi

DHAAL

- Dhaal means “Shield” in Hindi/Gujarati
- DHAAL research is about shielding systems from various threats

Books Update

- Probability and Statistics with Reliability, Queuing, and Computer Science Applications, 1982; Second edition, John Wiley, 2001 (Blue book)

- Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package, Kluwer, 1996 (Red book)

- Queuing Networks and Markov Chains, 1998
  John Wiley, second edition, 2006 (White book)

Reliability and Availability Engineering, Cambridge University Press, 2015 (green book)

Recent Research

- Reliability Analysis of Boeing 787 Current Return Network for FAA Certification (Boeing)
- Reliability and Availability Analysis of SIP protocol on High Availability WebSphere (IBM)
- Security Quantification (DARPA,NSF, NATO)
- Survivability Quantification for Lucent POTS, for VPN and Smart Grid (Alcatel-Lucent, NTNU, Siemens)
- Software Aging and Rejuvenation: theory, experiments and implementation (IBM, NEC, Huawei)
- Cloud capacity planning including performance, availability, power (IBM,NEC)
- Failures data analytics (NASA-JPL, Wipro)
- DSRC VANET Reliability (NSF)
- Parametric (Epistemic) Uncertainty Propagation (JPL)

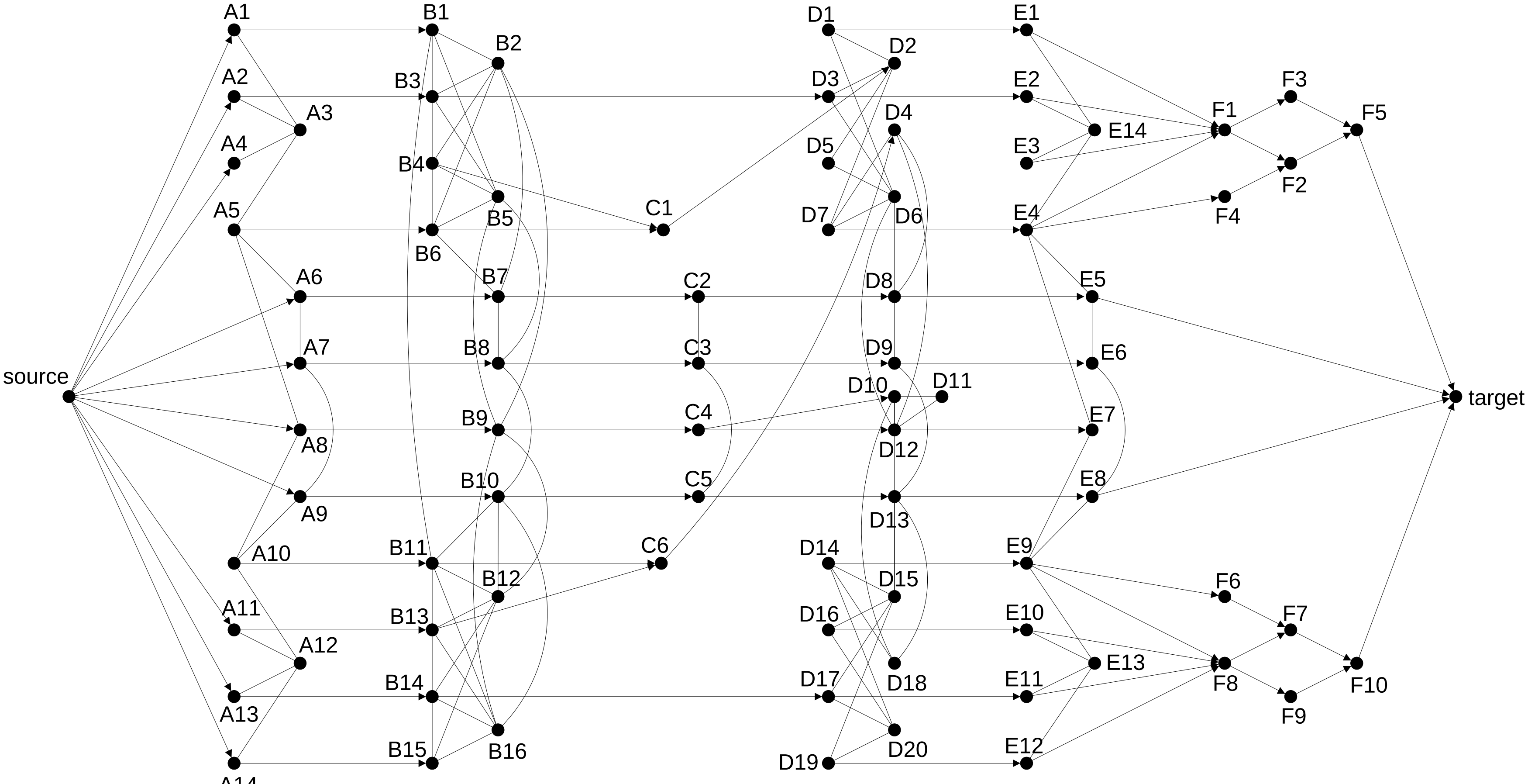
An Avionics Application

- Reliability analysis of each major subsystem of a commercial airplane needs to be carried out and presented to Federal Aviation Administration (FAA) for certification

Real world example from Boeing Commercial Airplane Company

Reliability analysis of Boeing 787

- ➢ Current Return Network Subsystem Modeled as a Reliability Graph (s-t connectedness; network reliability)
  - ✓ Consists of a set of nodes and edges
  - ✓ Edges represent components that can fail
  - ✓ Source and target nodes
  - ✓ System fails when no path from source to target
  - ✓ Compute probability of a path from source to target

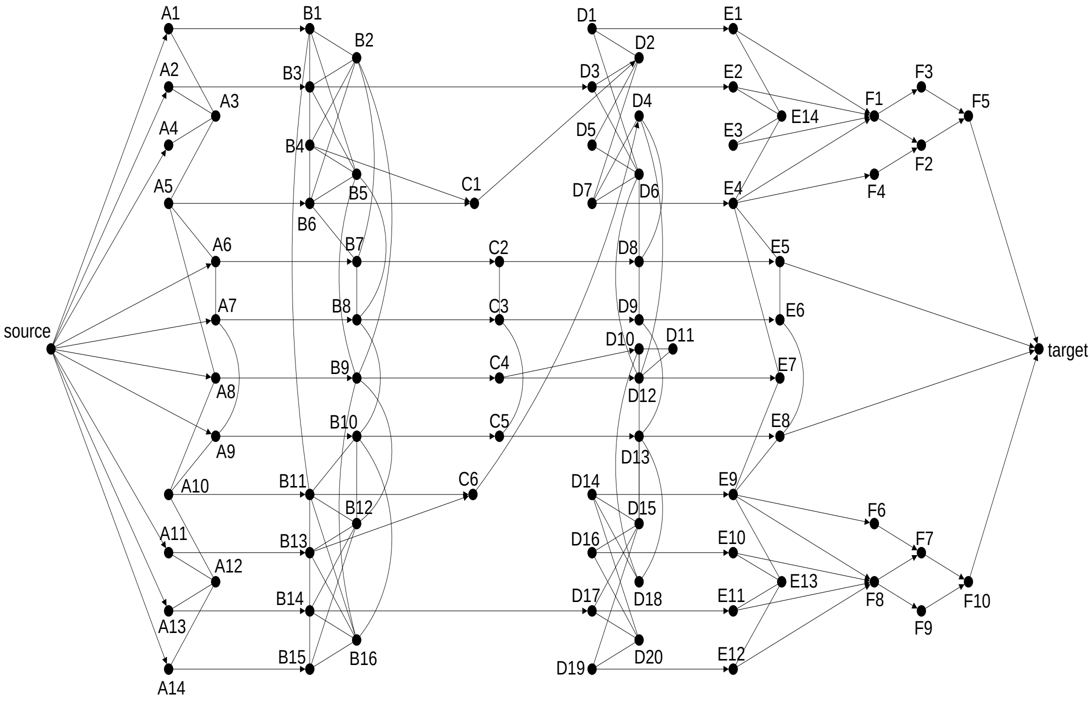


Reliability analysis of Boeing 787(contd.)

- ➢ Known solution methods for Relgraph
  - ✓ Find all minpaths followed by SDP (sum of disjoint products)
  - ✓ BDD (binary decision diagrams)-based method
- ➢ The above two methods have been implemented in our SHARPE software package
- ➢ Boeing tried to use SHARPE for this problem but it was too large to solve

Reliability analysis of Boeing 787(contd.)

- Too many minpaths



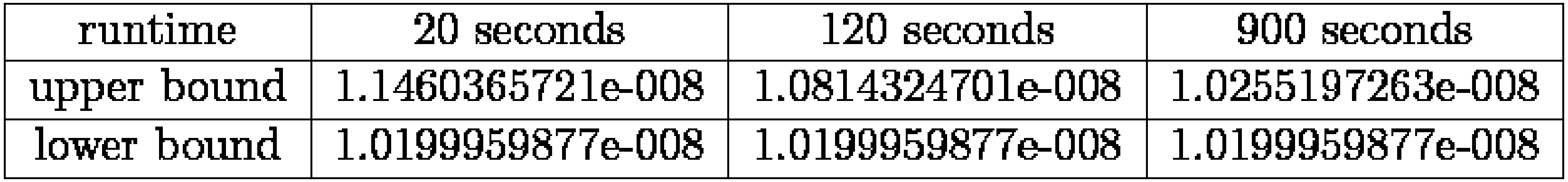
node	**#paths**
E_7 →target	40
D_{12} →target	143140
C_4 →target	308055
B_9 →target	21054950355
A_8 →target	461604232201
source → target	4248274506778≈ 4×10^{12}

Number of paths from source to target

- Idea: Compute bounds instead of exact reliability
- Lower bound by taking a subset of minpaths
- Upper bound by taking a subset of mincuts

Reliability analysis of Boeing 787(contd.)

- **Our Approach**: Developed a new efficient algorithm for (un)reliability bounds computation and incorporated in SHARPE



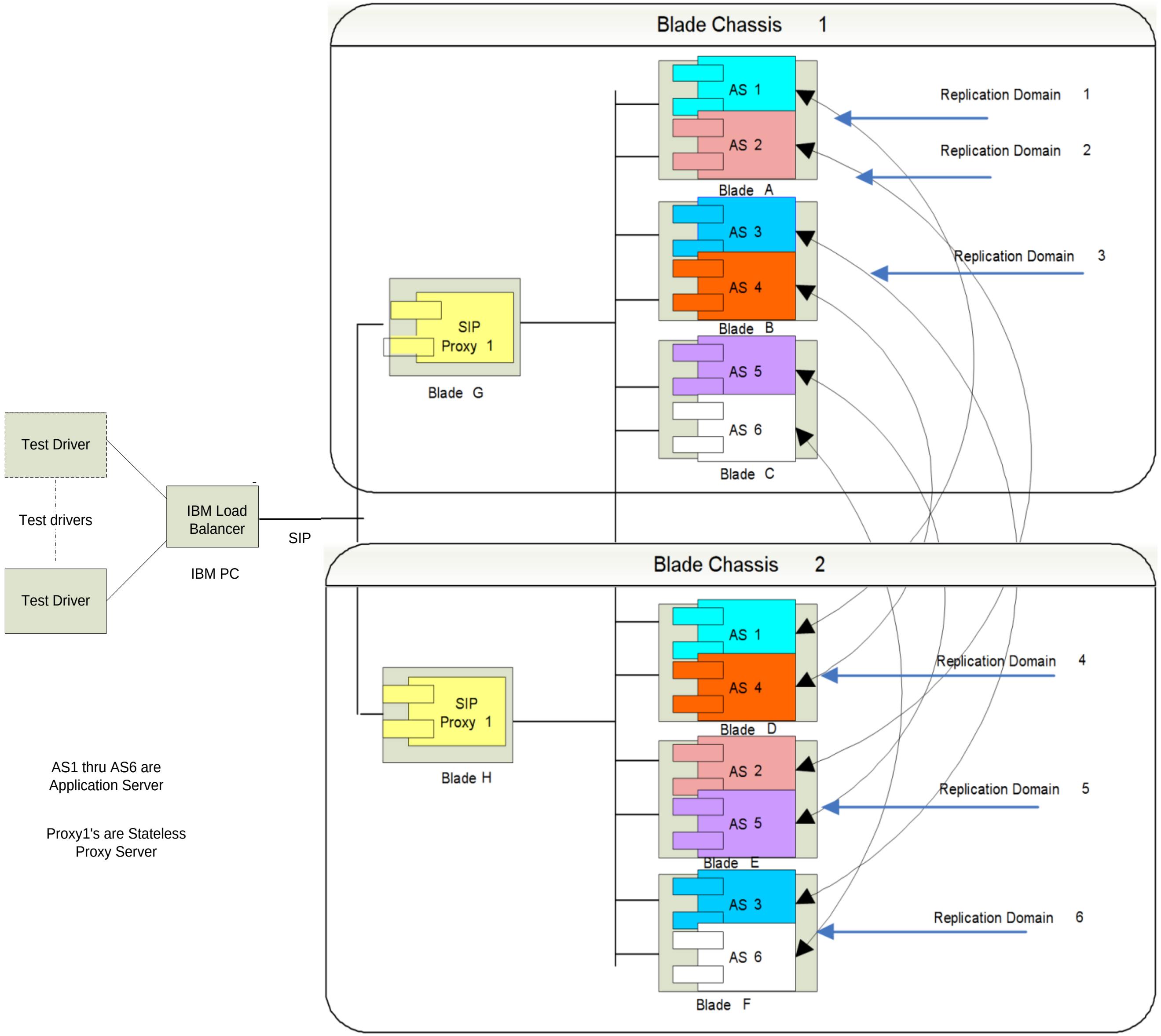
runtime	20 seconds	120 seconds	900 seconds
upper bound	1.1460365721e-008	1.0814324701e-008	1.0255197263e-008
lower bound	1.0199959877e-008	1.0199959877e-008	1.0199959877e-008

- 2011 patent for the algorithm jointly with Boeing/Duke
- A paper just appeared in EJOR, 2014

Model of SIP on IBM WebSphere

- Real problem from IBM
- SIP: Session Initiation Protocol
- Hardware platform: IBM BladeCenter
- Software platform: IBM WebSphere
- IBM's Potential Customer asked for a model
- I was asked to lead the modeling project
- Models published: PRDC 2008, ISSRE 2010 papers

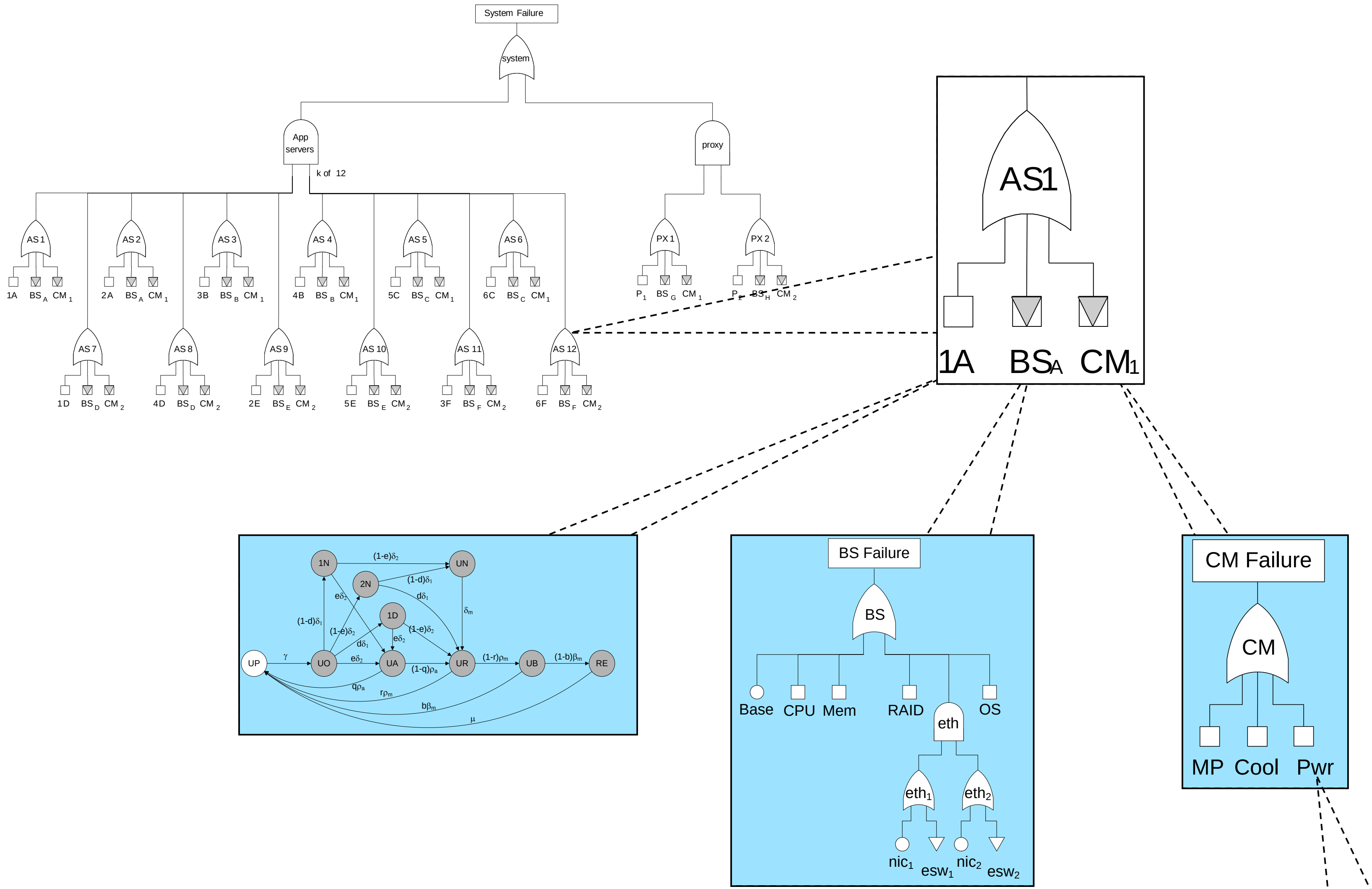
Architecture of SIP on IBM WebSphere

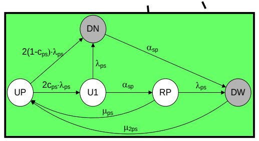


AS: WebSphere Appl. Server (WAS)

Replication domain	Nodes
1	A, D
2	A, E
3	B, F
4	B, D
5	C, E
6	C, F

Hierarchical composition approach





A single monolithic Markov model will have too many states

Key Contributions

- Developed a very comprehensive hierarchical availability model
  - ✓ Hardware and software failures;
  - ✓ Detection and Failover delays;
  - ✓ Escalated (multiple) levels of recovery;
  - ✓ Automated and manual restart, node reboot, repair;
  - ✓ Imperfect coverage (detection, failover, restart, reboot, …)
- Developed a new method for calculating DPM (defects per million)
  - ✓ Taking into account interaction between call flow and failure/recovery
  - ✓ Retry of messages (this model will be published in the future)
- Many but not all of the parameters collected from experiments
- Detailed sensitivity analysis to find bottlenecks
- This model was responsible for the sale of the system

Security Quantification

- Our papers on Security quantification
  - ✓ Madan, Goseva-Popstojanova, Vaidyanathan, Trivedi: "A method for modeling and quantifying the security attributes of intrusion tolerant systems," Perform. Eval. (2004)
  - ✓ A. Roy, D. Kim, K.S.Trivedi: "Attack countermeasure trees (ACT): towards unifying the constructs of attack and defense trees," Security and Communication Networks, 2012.
  - ✓ A. Roy, D. Kim, K.S. Trivedi: "Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees," in Proc. DSN 2012

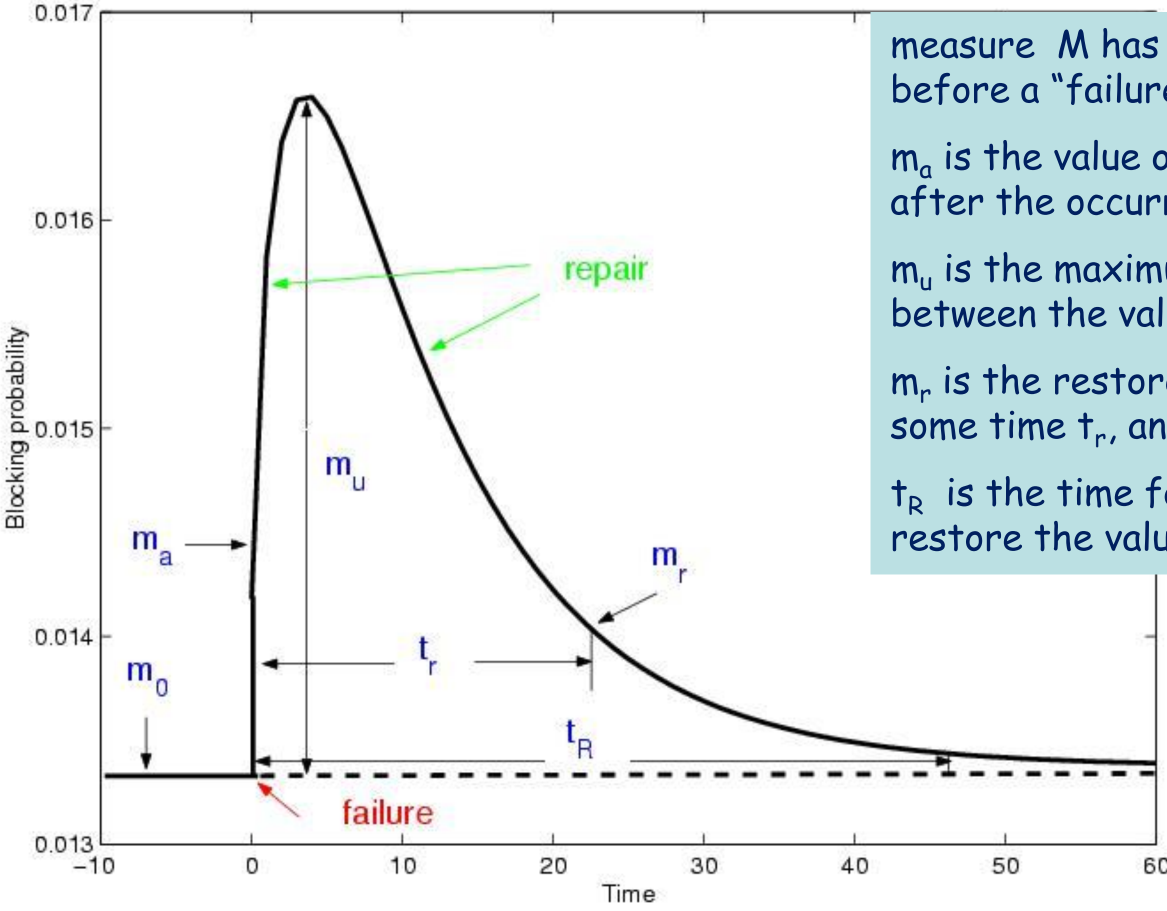
T1A1.2 survivability definition

- Suppose a measure of interest M has the value m_0 just before a failure happens. The survivability behavior can be depicted by the following attributes:
  - ✓ m_a is the value of M immediately after the occurrence of failure,
  - ✓ m_u is the maximum difference between the value of M and m_a after the failure,
  - ✓ m_r is the restored value of M after some time t_r, and
  - ✓ t_R is the time for the system to restore the value m_0.
- This definition is proposed by the T1A1.2 network survivability performance working group. By this definition, survivability depicts the time-varying behavior of the system **after** a failure/attack/disaster occurs (until system stabilizes).

Survivability

- Transient behavior
- After the occurrence of an undesired event
- Performance measure being considered (M)
- Undesired Event: failure, attack, disaster
- So survivability is transient performability (of critical services), conditional upon the occurrence of a failure, attack or disaster
- Can be seen as a generalization of "recovery time"

Quantitative Definition of Survivability



measure M has initial value m_0 just before a “failure”

m_a is the value of M immediately after the occurrence of failure,

m_u is the maximum difference between the value of M and m_a

m_r is the restored value of M after some time t_r, and

t_R is the time for the system to restore the value m_0.

Survivability Quantification

- Our papers on Survivability quantification
  - Y. Liu, Mendiratta, and K. Trivedi, "Survivability analysis of telephone access network," in IEEE Intl. Symposium on Software Reliability Engineering, 2004.
  - ✓ P. Heegaard and K. Trivedi, "Survivability Quantification of Communication Services," DSN 2008
  - ✓ P. Heegard, K. Trivedi, Network survivability modeling, Computer Networks, 2009.
  - ✓ A.Avritzer, S.Suresh, D.Menasche, R.Leao, E. de Souza e Silva, M.Diniz, K.Trivedi,A.Happe and, A.Koziolek."Survivability Models for the Assessment of Smart Grid Distribution Automation Network Designs," in Proc. ACM/SPEC, 2013
  - ✓ D.Menasche, R.M.M.Leao, E. de Souza e Silva, A.Avritzer, S.Suresh,K.Trivedi,R.A.Marie,L.Happe and, A.Koziolek. "Survivability Analysis of Power Distribution in Smart Grids with Active and Reactive Power Modeling," In Greenmetrics Workshop(special issue of Performance Evaluation Review), 2012

Software Aging and Rejuvenation

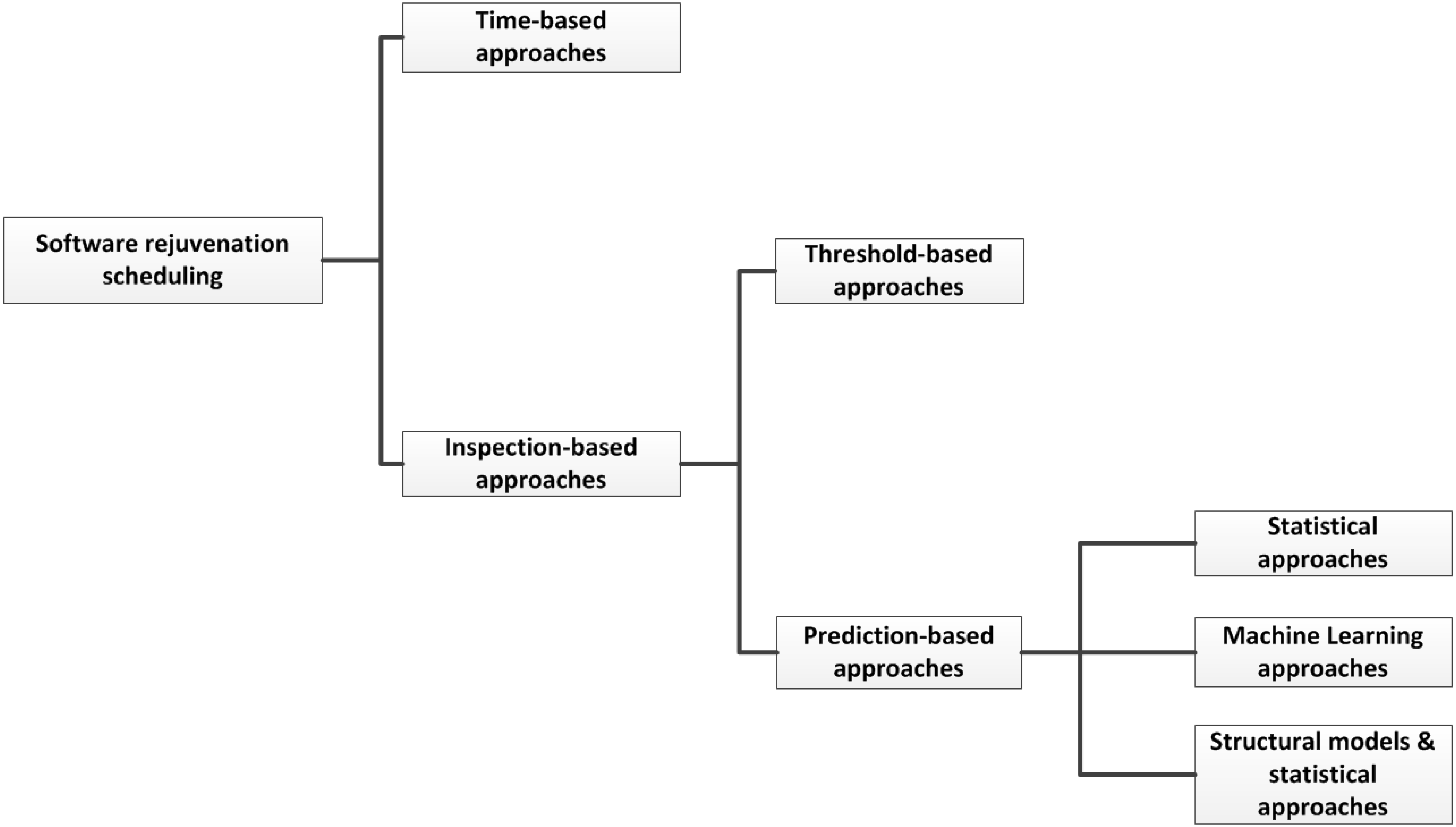
- Experimentally showed software aging and developed a prediction of time to resource exhaustion (Garg et al, ISSRE 1998)
- Helped implement software rejuvenation in IBM X-series (Castelli et al, IBM JRD, 2001)
- Annual workshop on software aging and rejuvenation as part of ISSRE since 2008
- Special issues of Performance Evaluation 2013, ACM JETC 2014
- Collaboration with NEC
- Collaboration with Huawei

Software Rejuvenation

- SOFTWARE REJUVENATION
  - ✓ Proactive fault management technique aimed at postponing/preventing crash failures and/or performance degradation
    - Involves occasionally stopping the running software, “cleaning” its internal state and/or its environment and restarting it
    - Rejuvenation of the environment, not of software

- Technique applied in any field and software:
  - ✓ AT&T billing applications,
  - ✓ JPL REE System,
  - ✓ Patriot missile system software - switch off and on every 8 hours,
  - ✓ On-board preventive maintenance for long-life deep space missions (NASA’s X2000 Advanced Flight Systems Program),
  - ✓ IBM Director Software Rejuvenation (x-series),
  - ✓ Recycling pool in Microsoft IIS 5.0, Apache, ORACLE DBMS, JBOSS, etc.
  - ✓ ISS FS SSC (ISS File system) - switch off and on every 2 months
  - ✓ Tens of US Patents related with this technology

Software Rejuvenation Scheduling



Software rejuvenation scheduling

Time-based approaches

Inspection-based approaches

Threshold-based approaches

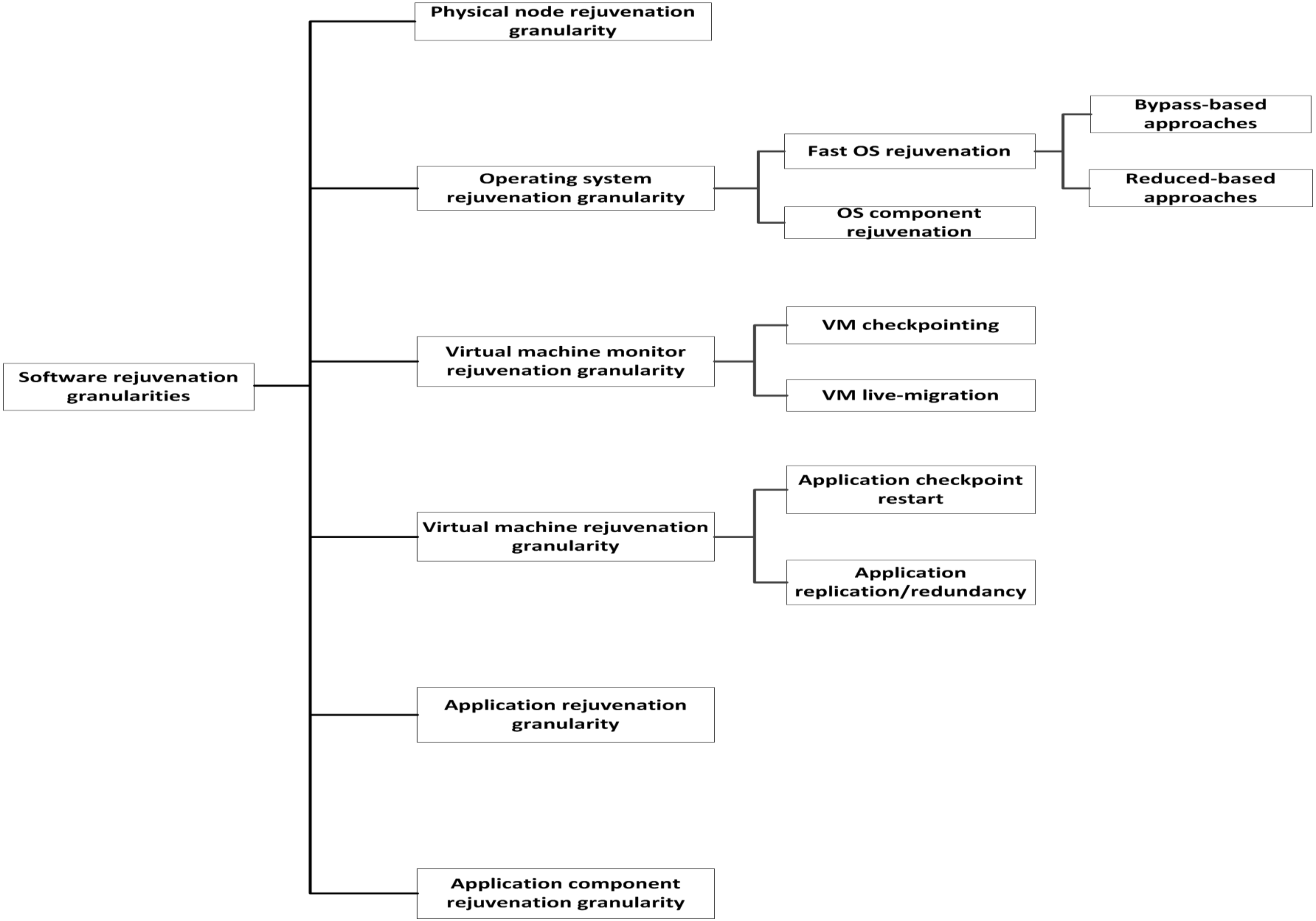
Prediction-based approaches

Statistical approaches

Machine Learning approaches

Structural models & statistical approaches

Software Rejuvenation Granularities



Software rejuvenation granularities

Physical node rejuvenation granularity

Operating system rejuvenation granularity

Fast OS rejuvenation

Bypass-based approaches

Reduced-based approaches

OS component rejuvenation

Virtual machine monitor rejuvenation granularity

VM checkpointing

VM live-migration

Virtual machine rejuvenation granularity

Application checkpoint restart

Application replication/redundancy

Application rejuvenation granularity

Application component rejuvenation granularity

Software Aging and Rejuvenation

- References

  - ✓ A Comparative experimental study of software rejuvenation overhead, J. Alonso, R. Matias, E. Vicente, A. Maria, K. S. Trivedi. Performance Evaluation, Vol. 70, Issue 3, 2013, 231-250.
  - ✓ Proactive management of software aging, V. Castelli, R. E. Harper, P. Heidelberger, S. W. Hunter, K. S. Trivedi, K. Vaidyanathan, W. P. Zeggert, IBM Journal of Research and Development 45 (2), 311-332, 2001
  - ✓ A comprehensive model for software rejuvenation, K. Vaidyanathan, K. S Trivedi, Dependable and Secure Computing, IEEE Transactions on, Vol. 2, N. 2, 124-137, 2005

Software Aging and Rejuvenation

- References (contd.)

  - Ensuring the performance of Apache HTTP server affected by aging,.Jing Zhao, Kishor S. Trivedi, Michael Grottke, Javier Alonso, YanBin Wang. Accepted for publication in IEEE Transactions on dependability and secure computing (TDSC), Sept. 2013.
  - A comprehensive approach to optimal software rejuvenation. Jing Zhao, Yanbin Wang, GaoRong Ning, Kishor S. Trivedi, Rivalino Matias Jr., Kai-Yuan Cai, Perform. Eval. 70(11): 917-933, 2013.
  - Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration. Fumio Machida, Dong Seong Kim, Kishor S. Trivedi, Perform. Eval. 70(3): 212-230, 2013.
  - Statistical Non-Parametric Algorithms to Estimate the Optimal Software Rejuvenation Schedule, T. Dohi, PRDC 2000
  - Accelerated Degradation tests Applied to Software aging experiments, R. Matias, P.A. Barbetta, K.S. Trivedi, P.J.F. Filho. IEEE Transactions on Reliability, vol 59, issue 1, 2010

JPL/NASA Failure data analytics

- Analyze the underlying faults causing software failures in flight (on-board) software
  - ✓ Bohrbugs (63.5%)
  - ✓ Mandelbugs (30.7%)
  - ✓ Aging-related bugs (3.8%)

- Hypothesis: Mandelbugs more frequently found during operation than Bohrbugs

- Results: Bohrbugs are more frequent during operation even in critical software than Mandelbugs

JPL/NASA Failure data analytics (contd.)

- REFERENCES:

- ✓ Alonso, J.; Grottke, M.; Nikora, A.P.; Trivedi, K.S., "An empirical investigation of fault repairs and mitigations in space mission system software," Dependable Systems and Networks (DSN), 2013 43rd Annual IEEE/IFIP International Conference on, pp.1,8, 24-27 June 2013
- ✓ Alonso, J.; Grottke, M.; Nikora, A.P.; Trivedi, K.S., "The Nature of the Times to Flight Software Failure during Space Missions," Software Reliability Engineering (ISSRE), 2012 IEEE 23rd International Symposium on, pp.331,340, 27-30 Nov. 2012
- ✓ Grottke, M.; Nikora, A.P.; Trivedi, K.S., "An empirical investigation of fault types in space mission system software," Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on, pp.447,456, June 28 2010-July 1 2010

Uncertainty Propagation through Dependability Models

Epistemic and Aleatory Uncertainty

- Stochastic models widely used to study Dependability & Performance of systems
- Capture inherent randomness in the system
  - ✓ Random time to failure/repair/service time/ time to arrival etc.. (captured by their distribution – exponential, Weibull …)
  - ✓ Known as Aleatory Uncertainty (also known as variability, stochastic uncertainty or irreducible uncertainty)
- Assume fixed values for model parameters
- Result dependent on the validity of assumed parameter values and their distributions

Epistemic and Aleatory Uncertainty

- In real life, parameter values of aleatory models determined from
  - ✓ Observed data
    - Lifetime experiments, field failure data, maintenance logs, workload data
    - Finite sample size
  - ✓ Expressed as confidence intervals, bounds or distributions of parameter values themselves
- Aleatory Models per se, do not consider epistemic uncertainty
  - ✓ Aleatory model evaluated at a fixed set of parameter values
    - Model output conditioned on the set of input parameter values
- Need to determine uncertainty in aleatory model output due to epistemic uncertainty in aleatory model input parameters

Accounting for Epistemic Uncertainty (contd.)

- The distribution of model output obtained as:

$$F_M(m) = \int \dots \int I(M(.) \leq m) f(.) d\theta_1 \dots d\theta_k$$

where, $I(\xi)$ *is the indicator variable for the event* ξ

- Expectation can be obtained as:

$$E[M] = \int \dots \int M(.) f(.) d\theta_1 \dots \theta_k$$

- Second Moment as $E[M^2] = \int \dots \int (M(.))^2 f(.) d\theta_1 \dots d\theta_k$
- Variance as:

$$Var[M] = E[M^2] - (E[M])^2$$

Uncertainty Propagation through Dependability Models

- Epistemic uncertainty propagation through aleatory models of varying complexity
- Analytic closed-form approach for uncertainty propagation
  - ✓ Exact expressions for distribution, expectation and variance of aleatory model o/p due to epistemic uncertainties, derived
  - ✓ Analyzed to study their limiting behavior
- Numerical Integration Approach for Uncertainty propagation
- A method to derive epistemic distribution from supplied confidence interval and aleatory distribution
- Sampling based approach for complex aleatory models

Uncertainty Propagation through Dependability Models (contd.)

- Wide range of aleatory model types
  - ✓ simple non-state-space models with a few input parameters
  - ✓ large state space models or
  - ✓ Hierarchical models with a large number of input parameters.
  - ✓ Also applied to aleatory models with simulative solutions.
- Applied to a wide range of dependability and performance models for computer systems.
  - ✓ Some of the model output metrics considered are
    - Reliability and availability of computer systems
    - Response time of a web service
    - Capacity oriented availability of a communication system
    - Security (probability of successful attack) of a network routing session
    - Expected number of jobs in a queueing system with breakdown-repair of servers
    - Call handoff probability of a cellular wireless communication cell.

Epistemic Uncertainty Propagation: Different Aleatory Model Types

Aleatory Model Types	Epistemic Uncertainty Propagation Method		
	Analytic Closed-Form	Numerical Integration	Sampling Based
Analytic Model: Closed-Form Solution	Applicable (Simple Expressions)	Applicable	Applicable
Analytic Model: Numerical Solution	Not Applicable	Applicable	Applicable
Aleatory Model with Simulative Solution	Not Applicable	Applicable	Applicable

Epistemic Unceraintу Propagation : Summary

- A sampling based non-intrusive approach for epistemic uncertainty propagation through large and complex aleatory models is presented
  - ✓ The form and parameters of the epistemic distributions are not arbitrarily assumed (as done in other sampling based methods), but derived based on the aleatory distribution and the provided confidence intervals.
  - ✓ More robust (demonstrated in the illustrative examples)
  - ✓ Successfully applied for uncertainty propagation through several dependability and performance models of computer systems (including rather large hierarchical models with a large number of model input parameters and even an aleatory model with simulative solution)

Epistemic Uncertainty Propagation : Summary

- Assumption of epistemic independence relaxed – rank correlation in the aleatory model input parameter value considered
  - ✓ Uncertainty propagation considering epistemic dependence between input parameters is also applied to some of the dependability models.
- The entire uncertainty propagation approach is applied to non-state-space, state-space, hierarchical and fixed point iterative aleatory models and aleatory model with simulative solution, solved using SHARPE, SPNP and Mathematica software packages

References

- K. Mishra and K. S. Trivedi. “Uncertainty Propagation through Software Dependability Models”, Int. Symp. on Software Reliability Engineering (ISSRE 2011).
- K. Mishra and K.S. Trivedi, “Closed-form approach for Epsitemic Uncertainty propagation in Analytic Models”. Stochastic Reliability and Maintenance Modeling, Springer, 2011.
- K. Mishra and K. S. Trivedi. An unobtrusive method for uncertainty propagation in stochastic dependability models. International Journal of Reliability and Quality Performance (IJRQP). Dec. 2011
- K. Mishra, K. S. Trivedi and R. R. Some. "Uncertainty Analysis of the Remote Exploration and Experimentation System", Journal of Spacecraft and Rockets, Vol. 49, No. 6 (2012)