

High-Level Reliability Languages Using A General Intermediate Domain

Robert Painter, M.S.
David Coppit, Ph.D.

Department of Computer Science,
College of William and Mary

Outline

- Introduction
- Problem of High-level Modeling Languages
- Reliability Modeling Example
- Our Approach: Intermediate Modeling Language
 - Failure Automaton (FA)
- Implementation
- Evaluation
- Conclusion
- Future Work

2

Reliability Engineering

- Goal: Determine the probability that a system/product will operate for a specified amount of time
- Method:
 - Know the reliability of components or rates that failures occur
 - Know the relationships between components and/or events
 - Use a solver that takes a high level language as input

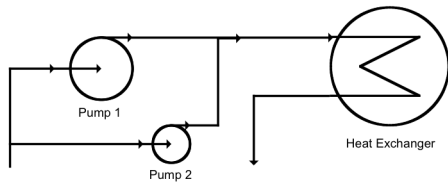
3

Reliability Modeling Problems

- Languages are defined *independently* in a *semi-formal* way, leading to
 - Expensive development effort
 - Ambiguous languages
 - Untrustworthy implementations
 - Difficulties in language comparison/sharing

4

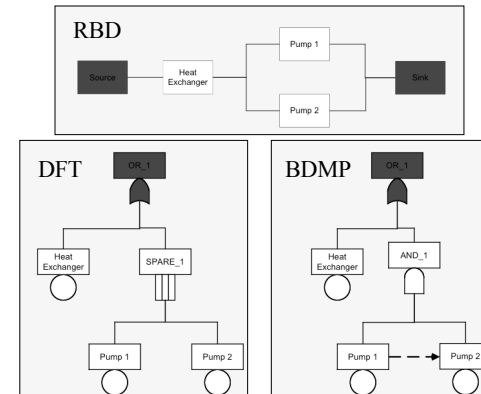
Reliability Modeling Example



- The system is operational while the heat exchanger and at least one pump is operational
- Pump 2 is dormant until Pump 1 fails

5

3 Example Models



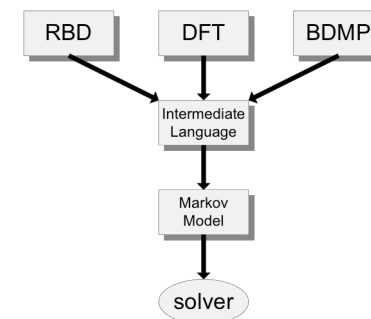
6

Our Approach

- We *formally* define high level languages with the use of a *common intermediate language*, Failure Automaton (FA)
- Our experiment tests our approach on a representative group of high-level reliability modeling languages
 - Dynamic Fault Trees (DFTs)
 - Reliability Block Diagrams (RBDs)
 - Boolean-Driven Markov Processes (BDMPs)

7

Language Mapping



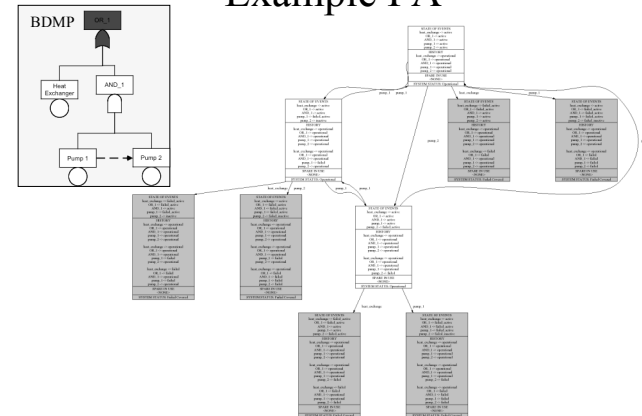
8

The Failure Automaton (FA)

- Our intermediate reliability modeling language
 - Domain specific formalism
 - State-based language structured like Markov models
- Revised earlier FA made by Coppit for DFTs
 - Refactored model (e.g. removed replication)
 - Added repair and failure on demand

9

Example FA



Separation of Concerns

- The FA represents the dependencies and interactions of basic events
- The independent behavior of basic events is represented with Basic Event Models (BEM)
- An FA and corresponding BEM can be translated into a Markov model

11

Implementation

- Three solvers based on the specification
 - Nova_solver solves DFT models
 - Firefly_solver solves RBD models
 - BDMP_solver solves BDMP models
- Use the FA solver as a library

12

Implementation Experiences

- Our method reduced initial implementation costs
 - The FA solver was a reusable library
 - The coding was conceptually easier with the intermediate language
- Debugging was simplified because FA is an easy to read notation that is full of rich semantic information

13

Benefits of Our Method

- Cost of language development and implementation is reduced
- Precise meaning of languages is specified
- Reduced the semantic gap by separating concerns
- Reduced the distance to the target by using a higher level language
 - Advances can be shared between languages
 - Understanding differences between languages is easier

14

Drawbacks of Our Method

- Not sure if the FA is general enough to be a true intermediate language for reliability modeling
- Formal approach may be more costly
- FA not familiar to language designers
- Not sure that other costs won't increase, such as maintenance

15

Conclusion

- Feasible: using a formally specified intermediate language in the mapping of high-level languages to low-level languages
- General: We have formally specified three high-level languages in terms of the FA
- Practical: We have built solver tools for the three languages based on the specification using the `fa_solver` as a library

16

Current and Future Work

- Our approach eases the cost of porting features between languages
 - Added *repair* and *coverage modeling* to RBDs
 - Added *coverage modeling* to BDMPs
- Need to evaluate other costs
- Need to make FA available to researchers

17

References

- Marc Bouissou. Boolean logic driven markov processes: A powerful new formalism for specifying and solving very large markov models. In *Proceedings of the 6th International Conference on Probabilistic Safety Assessment and Management*, San Juan, Puerto Rico, USA, 23–28 June 2002.
- David Coppit. *Engineering Modeling and Analysis: Sound Methods and Effective Tools*. PhD thesis, The University of Virginia, Charlottesville, Virginia, January 2003. URL: <http://www.cs.wm.edu/~coppit/papers/dissertation.pdf>.
- David Coppit, Robert R. Painter, and Kevin J. Sullivan. Shared semantic domains for computational reliability engineering. In *Proceedings of the International Symposium on Software Reliability Engineering*, pages 169–80, Denver, Colorado, 17–20 November 2003. IEEE.
- David Coppit, Kevin J. Sullivan, and Joanne Bechta Dugan. Formal semantics of models for computational engineering: A case study on dynamic fault trees. In *Proceedings of the International Symposium on Software Reliability Engineering*, pages 270–282, San Jose, California, 8–11 October 2000. IEEE.
- J. M. Spivey. *The Z Notation: A Reference Manual*. Prentice Hall International Series in Computer Science, 2nd edition, 1992.

18

Thank You