Validating Evolving Simulations in COERCE

Paul F. Reynolds Jr., Michael Spiegel, Xinyu Liu, and Ross Gore

Computer Science Department
University of Virginia
{reynolds, ms6ep, xl3t, rjg7v}@cs.virginia.edu

Abstract. We seek to increase user confidence in simulations as they are adapted to meet new requirements. Our approach includes formal representation of uncertainty, lightweight validation, and novel techniques for exploring emergent behavior. Uncertainty representation, using formalisms such as Dempster-Shafer theory, can capture designer insight about uncertainty, enabling formal analysis and improving communication with decision and policy makers. Lightweight validation employs targeted program analysis and automated regression testing to maintain user confidence as adaptations occur. Emergent behavior validation exploits the semi-automatic adaptation capability of COERCE to make exploration of such behavior efficient and productive. We describe our research on these three technologies and their impact on validating dynamically evolving simulations.

1 Introduction

Uncertainty pervades model assumptions, and so frequently model designers must base decisions on little more than informed guesses. This condition presents both an opportunity and a risk. Opportunity arises from the rich set of outcomes a model can produce while exercising reasonable alternatives created by uncertainty. Risk results from not knowing which assumptions, and combinations of assumptions, reflect truth. Exploiting the opportunity while managing the risk is our ultimate goal: we seek to limit the consequences of uncertainty while providing an opportunity to adapt simulations to meet new requirements. We place high priority on maintaining user confidence in the correctness of a simulation as adaptation proceeds.

No formal approach to representing uncertainty in model descriptions or simulation languages exists. Therefore our investigation into uncertainty representation has begun with a clean slate and a broad opportunity. Our approach is to explore uncertainty representation for all aspects of model uncertainty, and not just those that best serve the needs we find in simulation adaptation. However, our investigation of uncertainty representation methods treats support of adaptation as a high priority. Our goal is to enable formal representation of potential model inputs, model outcomes, and related likelihoods –probabilities and plausibilities– for the purpose of reducing risk. The representation formalism must support automated and semi-automated analysis, and improvement of communication among model designers and implementers and related policy and decision-making personnel and processes. We discuss our progress on uncertainty representation in section 2.

Simulation adaptation, both before execution (static) and during execution (dynamic), is a high payoff capability that can save mechanical, human and monetary

Y. Shi et al. (Eds.): ICCS 2007, Part I, LNCS 4487, pp. 1238 – 1245, 2007.

resources. Our adaptation technology, COERCE [1], exploits expert knowledge provided at design time for enhancing simulation adaptability both before and during execution. COERCE increases the potential for simulations to adapt to changes occurring in, for example, DDDAS environments that combine simulation and live data. COERCE employs software annotations, which we call "flexible points," to capture expert knowledge about model assumptions and alternatives. Flexible point alternatives often reflect a significant degree of uncertainty. Treatment of uncertainty, which can be daunting enough for models as defined in a traditional sense, can become considerably more complex when interactions among alternatives for model assumptions and design decisions – flexible points – are also considered. Thus arises the need for safeguards for ensuring user confidence in a simulation as is it progresses through the COERCE adaptation process

Factors that influence the efficacy of user confidence safeguards include management of uncertainty, cost of meeting user confidence goals, and technical feasibility of tools for supporting desired guarantees. Current software validation methods, which would seem to offer the best hope for maintaining user confidence, do not apply well to simulation adaptation. As a rule they cost too much and require more information than is generally available. Similarly, technologies for validating emergent behaviors in simulations are nascent [2]. We view efficient validation and emergent behavior explanation as essential safeguards during simulation adaptation. Thus, we have concluded that we must create methods for maintaining user confidence as a simulation is adapted, and we must provide support for exploring and validating emergent behaviors, all in the presence of model uncertainty, as adaptation proceeds. Our approaches for addressing user confidence issues include lightweight validation and semi-automatic emergent behavior exploration We discuss our approaches to lightweight validation and emergent behavior exploration in sections 3 and 4, resp.

2 Representations of Uncertainty

We are designing a language for the formal representation of uncertainty in modeling and simulation, for quantitative risk assessment. Modeling under uncertainty has been of paramount importance in the public and private sector for the past half century, as quantitative methods of analysis have been developed to take advantage of computational resources. Our work brings together the fields of computer science and risk analysis. Some prominent public policy examples of uncertainty analysis in simulation include the technical studies for the Yucca Mountain nuclear waste repository [3], the assessment reports of the Intergovernmental Panel on Climate Change [4], and the guidelines from the Office of Budget and Management that recommend formal quantitative uncertainty analysis for major rules involving annual economic effects of $1 billion or more [5].

Our uncertainty representation design effort possesses two primary goals. The first is representation of continuous and discrete random variables as first-class citizens in a programming language. We aim to employ multiple mathematical frameworks for the representation of random variables. Each mathematical framework displays a tradeoff of relative expressive power for ease of use. In following subsections, we will show how three mathematical frameworks can be appropriate when varying degrees of information are available. Probability theory suffers from three primary weaknesses when representing uncertainty [6]. First, a precise probability value must be assigned to each element in the set of possible outcomes. It may not be possible to assign exact values or even assign reasonable approximations when little information is available. Second, probability theory imposes Laplace's principle of insufficient reason when no information is available. When n mutually exclusive possible outcomes are indistinguishable except for their names, they must each be assigned a probability of 1/n. Third, conflicting evidence cannot be represented in traditional probability theory. By assigning probabilities to individual elements, we can express neither incompatibility nor a cooperative effect between multiple sources of information.

Our second design goal is the capacity to specify calibration techniques for uncertainty representations in all three mathematical frameworks. Modeling under uncertainty implies the absence of perfect information, but often partial information exists in the form of observations on the model's expected behavior. Simulation practitioners expect to make the best possible use of the information available to them. A Bayesian engine is able to support the calibration of probability theory, possibility theory, and probability mass functions, and we consider such an approach.

2.1 Imprecise Probabilities

Several different mathematical systems can be used to perform uncertainty analysis. We will focus on probability theory, probability boxes, and the Dempster-Shafer theory of evidence. Probability theory is the most traditional representation of uncertainty and the one most familiar to non-mathematicians. The use of probability theory attempts to provide a quantitative analysis to answer the following three questions: (1) what can go wrong, (2) how likely is it that will happen, and (3) if it does happen, what are the consequences? [7]. Probability as a representation of subjective belief is common in quantitative risk analysis. Safety assessments must deal with rare events and thus it is difficult to assess the relative frequencies of these events [8].

2.2 Probability Boxes

Probability boxes define upper and lower boundaries for the probabilities of a set of events [9]. These boundaries (represented by $\overline{P}(X)$ and $\underline{P}(X)$) can provide information not available using traditional probability theory. A gambler's interpretation of $\underline{P}(X)$ is that it represents the highest price s/he is willing to pay in order to receive one dollar if X occurs, or receive nothing if X does not occur. Similarly, $\overline{P}(X)$ represents the infimum selling price of an event, which is the lowest price that s/he is willing to receive in order to sell one dollar if X occurs. Probability boxes are the upper and lower distribution functions ($\overline{F}$ and $\underline{\mathrm{F}}$) of an event X where $\underline{F}(x) = \underline{P}(X \le x)$ and $\overline{F}(x) = \overline{P}(X \le x)$. Upper and lower distribution functions allow an analyst to make no assumptions about the shape of the true probability distribution function. A series of coherency axioms ensure that $\underline{F}(x) \le F(x) \le \overline{\mathrm{F}}(\mathrm{x})$ for all real numbers x. Probability boxes enable some separation of epistemic uncertainty and aleatory uncertainty [10], [11]. Under classical probability theory, the principle of indifference dictates one should select a uniform distribution when presented with a lack of information concerning the shape of that distribution. Traditional probabilistic analysis eliminates the epistemic uncertainty of the model and can result in misleading risk assessment calculations.

Regan et al. [12] investigated EPA calculations for Ecological Soil Screening Levels (Eco-SSLs) in Superfund ecological risk assessments. The study compared deterministic calculations of Eco-SSLs with a Monte Carlo approach and a probability bounds approach. Results show that Eco-SSL estimates using conservative deterministic methods were greater than estimates using probability bounds methods by two to three orders of magnitude. Median-based deterministic calculations resulted in estimates approximately one order of magnitude greater than conservative deterministic methods. Estimates based on Monte Carlo simulation generally fell between conservative and median-based deterministic estimates. The Monte Carlo simulation fails to produce a conservative estimate due to a combination of assumptions about dependencies between variables, and assumptions about the shape of the probability distribution curves. The authors "believe that probability bounds analysis is most useful as a tool for identifying the extent of uncertainty in model application and can assist in reducing this uncertainty."

2.3 Dempster-Shafer Theory of Evidence

In Dempster-Shafer theory, the concept of imprecise probabilities is extended to account for both non-specificity and discord of available evidence [13]. Probability boxes account for non-specificity by propagating lower and upper bounds without specifying the shape of the distribution. But probability boxes require that all the available evidence concludes in one non-overlapping interval. Dempster-Shafer theory allows a decision maker to reason about several candidate probability intervals for a random process, even when they conflict with one another. Dempster-Shafer theory is formulated in terms of a function known as the basic probability assignment. If Ω is the set of all possible outcomes and 2^{Ω} its power set, then a basic probability assignment is defined as $m(A) : 2^{\Omega} \rightarrow [0, 1]$ such that: $m(\emptyset) = 0$ and $\sum_{A \in 2^{\Omega}} m(A) = 1$.

Helton et al. [14] present a Dempster-Shafer risk analysis of a hypothetical safety system that is exposed to fire. The safety system consists of one weak link (WL) component and one strong link (SL) component that are both exposed to thermal heating. Both components will ultimately fail at sufficiently high temperatures. The weak link component is designed to fail safe during accidents and render the system inoperational. The strong link component is designed to be robust and resistant to extreme environments. Risk analysis is performed to assess the likelihood that the WL component will fail first. A time-dependent thermal response curve is used to model the high temperature scenario. The model contains 11 uncertain parameters such as initial temperatures, maximum temperatures, thermal constants, frequency responses, and expected values and standard deviations of normal distributions.

Dempster-Shafer theory enables the expression of several forms of partial information concerning uncertain parameters. For example, peak amplitude of the WL temperature transient (T) was measured in laboratory environments. Three measurement techniques resulted in different recorded intervals for the parameter: $T_1 = -500 \pm 40°C$, $T_2 = -1000 \pm 60°C$, $T_3 = -1800 \pm 80°C$. All three sources are considered equally credible, yet the intervals give conflicting information. Equal credibility can be expressed by assigning $m(T_1) = m(T_2) = m(T_3) = 1/3$. Evidence theory can also express nested probability structures. The thermal heating time constant (H) of the WL temperature transient is expressed with the following confidence intervals: $H_1 = 0.27 \leq H \leq 0.30$ min^{-1} with 30% confidence, $H_2 = 0.25 \leq H \leq 0.35$ min^{-1} with 50% confidence, and $H_3 = 0.20 \leq H \leq 0.40$ min^{-1} with 100% confidence. H_1, H_2, and H_3 are nested intervals that can be interpreted as plausibility measurements on H. Calculating backwards from the plausibility measurements yields the basic probability assignments $m(H_1) = 0.3$, $m(H_2) = 0.2$, and $m(H_3) = 0.5$.

Formal representations of uncertainty can be treated as sources of information regarding both model flexibility and bounds on model correctness. In the following sections we report on our approach to ensuring user confidence in model correctness as alternatives are explored in the COERCE adaptation process, and we report on exploiting the flexibility that representations of uncertainties present for validating emergent behaviors in simulation execution.

3 Lightweight Validation

Automated lightweight validation is meant to maintain user confidence that a simulation adaptation is proceeding in accordance with expectations. The approach is triggered when a user explores alternatives that uncertainty and model assumptions present, and then requires assurance that certain *correctness properties* have been maintained. A subset of requirements –correctness properties– most important to the user is identified, thus reducing analysis cost. This approach reflects a lightweight cost-efficient analysis that builds confidence, as a replacement for traditionally expensive, full validation or even regression testing methods. The concept is designed and parts have been tested as we discuss more below.

While COERCE improves the cost-effectiveness of simulation adaptation, there is a risk of introducing errors into the simulation during adaptation. When changes are made through the adaptation process, there may be inconsistencies and even conflicts among the changes. Therefore, the results of the adaptations must be validated or verified. Complete validation using statistical methods [15] is generally too expensive to be applied at each step of the adaptation process and should be applied only after a user believes that completed adaptations will not be reversed.

Our work has focused on using abstraction methods to improve the cost-effectiveness of validation [16]. We have identified two uses of abstraction: guiding optimization and checking coercion. We have explored extensions of existing abstraction methods such as program slicing, data approximation, behavior reduction and decomposition, [17, 18], and we explored an abstraction based on partial traces [19]. Our study of the adaptation of an abstract bicyclist simulation demonstrated the benefit of using abstraction methods. With a data approximation method, it took three hours to filter out 12.4% of invalid combinations of flexible point values. With partial trace abstraction it took five minutes to filter 31.6% of invalid combinations. With a control abstraction method, our results showed that previously determined optimal flexible point bindings did not extrapolate to similar but different paths for the bicyclist. Of the abstraction tools we have explored, fully-automated program slicing is one of the most promising. We have developed a prototype program slicer based on the program analysis tool, SOOT [20]. The slicer can perform inter-procedural slicing of Java programs using SOOT's program analysis framework.

Future work will employ *impact analysis.* Software impact analysis concerns estimating portions of software that can be affected if a proposed software change is made. The approach we envision will operate as follows. First, a subset of requirements, represented as correctness properties, is identified to help maintain user confidence in changes brought about by an adaptation, and to reduce analysis cost. Once the correctness properties and changes are known, impact analysis is performed to extract an impact set within the software. Built on this impact set, *automated* test case generation techniques are employed to generate test cases targeted specifically at the changes. Then, regression testing is employed using the test cases generated to validate correctness properties. If faults are detected, the adaptation process resumes until another solution is found. One contribution of our work beyond simulation adaptation will be our new methods for automated generation of test cases.

4 Validating Emergent Behaviors

Simulation behavior is emergent if it is unexpected and stems from interactions of underlying model components. Emergent behavior can be beneficial for the insight it provides. Emergent behavior can be harmful if it reflects an error in model construction. Because models often include a great deal of uncertainty, it is important that users have tools available for establishing the validity of emergent behaviors.

Validation of emergent behaviors requires an exploration capability that extends a model beyond its original intended use, so that users can test hypotheses about the characteristics of emergent behaviors. Need for a model extension capability requires adaptation which COERCE supports [1, 19]. We call our adaptation-based exploration process *Explanation Exploration.* Explanation Exploration (EE) allows a user to observe characteristics of emergent behavior as a simulated phenomenon is semi-automatically driven towards *conditions of interest*, as we explain further below.

Multiple advantages arise as a result of using COERCE: 1) COERCE flexible points enable capture of a broader range of model abstraction alternatives (both structural and parametric [19]) than a typical parameterized approach supports, 2) because COERCE employs semi-automated search methods, users can efficiently explore questions they might not have otherwise investigated, and 3) users can explore relationships between simulation behaviors they understand, but do not necessarily know how to induce, directly or indirectly, and emergent behaviors.

What constitutes a behavior can vary. Of importance to us is how a user relates choices about flexible points and knowledge of uncertainty to behaviors, and how behaviors are related to each other. An emergent behavior, *E*, occurs when some subset of observable simulation behaviors exhibits a pattern of unexpected behavior(s) across a set of simulation trials. An emergent behavior in a sailboat simulation may be: "the velocity of the sailboat is sometimes greater than the true wind speed when the sailboat's orientation is near perpendicular to the true wind direction." Given an emergent behavior, a user must establish if expectations regarding simulation behaviors need to be modified to include the emergent behavior. Alternatively the user may decide the emergent behavior is an error and not valid. EE facilitates this decision process. The user generally needs to formulate hypotheses about the relationship between alternatives (arising from flexible points or uncertainties) and variations of *E*, manifested as a function of bindings chosen for the flexible points.

A user will identify either a direct coupling between a set of alternatives (e.g. flexible points) and emergent behaviors, or an indirect coupling. Informally a direct coupling hypothesis is:

Direct Coupling Hypothesis: Within selected sets of bindings for a selected set of flexible points, predictable behavior E_{dc} related to E will be manifested in accordance with user expectations.

Exploration of direct couplings can be conducted in a straight-forward manner: alternatives for flexible point bindings can be tested and impact on emergent behaviors can be analyzed directly. Indirect couplings pose more interesting challenges because a user may not be able to hypothesize a direct link between alternatives (e.g. flexible points), bindings for those flexible points and expectations about an emergent behavior. However, it may be possible to identify instrumentable conditions within the simulation that can be related directly to emergent behavior expectations. If the user can then identify flexible points that relate directly to the intermediate conditions then a composition of the direct relationships yields a direct relationship between the flexible points and the emergent behavior. However, it is often the case that the user does not know how to make the intermediate conditions occur directly. If s/he can offer possible relevant sets of flexible points and bindings then a hypothesis may be testable with the support of search methods, such as COERCE. Informally an indirect coupling hypothesis is:

Indirect Coupling Hypothesis: For a range of allowable sets of bindings for a range of allowable flexible points, there are cases when intermediate condition C arises. When C arises, behaviors E_{ic} related to emergent behaviors E will be manifested in accordance with user expectations.

Because the user does not know which specific flexible point sets or bindings will cause condition C to arise, search will be employed. The relationship between C and E_{ic} is conjecture on the part of the user, to be established by the outcome of testing the indirect coupling hypothesis.

5 Summary

Model uncertainty presents a significant challenge to model and simulation designers, implementers and users and the policy and decision makers who often depend on their product. Model adaptation to satisfy new requirements, whether of a static (before execution time) nature, or a DDDAS dynamically adapting nature, compounds the challenge. Without methods for harnessing uncertainty and managing user confidence, particularly when model adaptation takes place, simulation designers will continue to face stiff challenges to the validity of their models. Here we have presented our approach to formal representation of uncertainty in model descriptions and simulations and the methods we have designed (and partially implemented) for maintaining user confidence in a model. Our analysis is not complete, but our objectives are clear and our designs are mature, as reflected in the work presented here.

Acknowledgments. The authors gratefully acknowledge the support of the NSF under grant 0426971.

References

1. Waziruddin, S., Brogan, D.C., Reynolds, P.F.: Coercion through optimization: A classification of optimization techniques. In: Proceedings of the Fall Simulation Interoperability Workshop. (2004)
2. Davis, P.K. "New Paradigms and Challenges", *Proceedings of 2005 Winter Simulation Conference*, IEEE, Piscataway, NJ, 2005, pp. 293-302.
3. OCRWM. Yucca mountain science and engineering report REV 1. DOE/RW-0539-1. U.S. Department of Energy, Office of Civilian Radioactive Waste Management, Las Vegas, Nevada, 2002.
4. IPCC. Ipcc fourth assessment report climate change. Intergovernmental Panel on Climate Change, 2007.
5. OMB. Circular a-4, regulatory analysis. Office of Management and Budget, September 2003.
6. Sentz, K and Ferson, S. Combination of evidence in dempster-shafer theory. Sandia National Laboratories, 2002.
7. Kaplan, S. and Garrick, B. On the quantitative definition of risk. Risk Analysis, 1(1):11–27, 1981.
8. Apostolakis, G.. The concept of probability in safety assessment of technological systems. Science, 250:1359–1364, December 1990.
9. Walley, P.. Statistical Reasoning with Imprecise Probabilities. Chapman/Hall, 1991.
10. Ferson, S. and Ginzburg, L.. Different methods are needed to propagate ignorance and variability. Reliability Engineering and System Safety, 54:133–144, 1996.
11. Ferson, S. and Hajagos, J. Arithmetic with uncertain numbers: rigorous and best possible answers. In: Reliability Engineering and System Safety, 85:135–152, 2004.
12. Regan, H., Sample, B. and Ferson, S.. Comparison of deterministic and probabilistic calculation of ecological soil screening levels. Environmental Toxicoloy and Chemistry, 21(4):882–890, 2002.
13. Shafer, G.. A mathematical theory of evidence. Princeton University Press, 1976.
14. Helton, J., Oberkampf, W. and Johnson, J.. Competing failure risk analysis using evidence theory. Risk Analysis, 25(4):973–995, 2005.
15. Easterling, R. and Berger, J. Statistical foundations for the validation of computer models. In: V&V State of the Art: Proceedings of Foundations '02. 2002.
16. Liu, X., Reynolds, P. and Brogan, D.. Using abstraction in the verification of simulation coercion. In: Proceedings of the 2006 Conference on Principles of Advanced and Distributed Simulation (PADS). 119-128, 2006.
17. Holzmann, G.. The SPIN Model Checker: Primer and Reference Manual. Addison-Wesley, Boston, Massachusetts, 2004.
18. Lynch, N., Segala, R., and Vaandrager, F. Hybrid, I/O Automata, In: Technical Report: MIT-LCS-TR-827d, MIT Lab. for Computer Science, Jan. 2003.
19. Carnahan, J., Reynolds, P., and Brogan, D. Language constructs for identifying flexible points in coercible simulations. In: Proceedings of the 2004 Fall Simulation Interoperability Workshop. Simulation Interoperability Standards Organization, Orlando, Florida. September 2004
20. Vallee-Rai, R., Co, P., Gagnon, E., Hendren, L., Lam, P., and Sundaresan, V.,. Soot - a Java bytecode optimization framework. In: Proceedings of the 1999 Conference of the Centre for Advanced Studies on Collaborative Research, 1999.