The potential for synergy between certification and insurance

Paul Luo Li	Mary Shaw	Kevin Stolarick	Kurt Wallnau
Institute for Software Research International	Institute for Software Research International	The Sloan Software Center	Software Engineering Institute

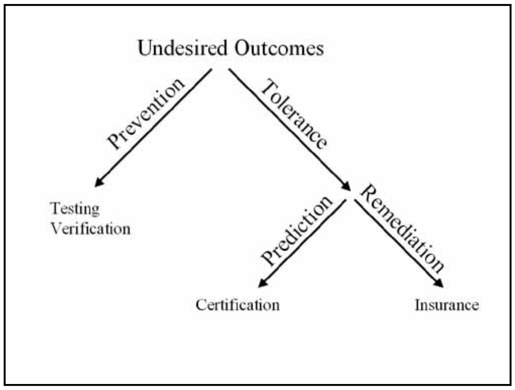
Carnegie Mellon University
Pittsburgh PA

Abstract

Because of their affordability and availability, reusable software components have long been a tantalizing IT investment. However, the risks associated with uncertainties about technical attributes and lack of protection against undesirable behaviors often deters their adoption. Certification and insurance are potential approaches to managing these risks. Probabilistic certification and insurance base their predictions and products on similar kinds of data; this offers the prospect of consistency (by using the same data for both) and cost-effectiveness (by reusing the data). The combined benefits of the two methods in the form of risk reduction and lowering of variance may make software reuse investments more attractive to risk-averse companies.

Introduction

Reuse is often claimed to improve software quality, drive down costs, and speed up development [9], but myriad difficulties hinder its adoption. These risks include technical uncertainties [2] and non-technical issues such as management support [10]. If methods were available to quantify and manage the risks associated with reuse, then IT managers might be more willing to undertake reuse investments.



The first line of defense against undesirable outcomes is prevention during design and development. Since this is not always successful, we explore alternatives to classical fault-tolerance to mitigate undesirable outcomes that could arise. Our two-pronged approach combines certification that predicts the likelihood of an undesirable behavior with insurance that provides financial compensation for the consequences. Certification would offer statistical assurances regarding component behavior, specifically the probability that its behavior would remain within some nominal range. Insurance would complement certification by providing remediation in the event a component behaved outside its specified nominal range.

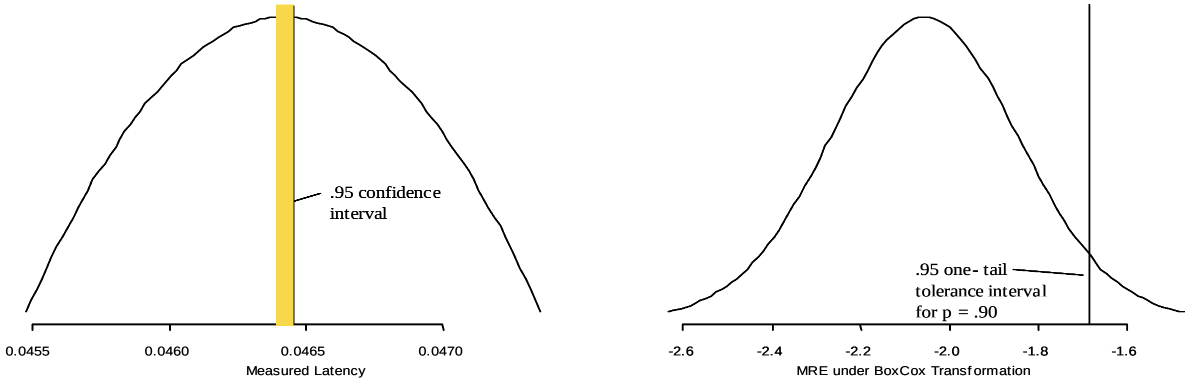
This dual approach is particularly attractive because the probabilistic assurances required for certification appear to rely on the same kinds of information that are required for insurance. In this position paper we explore the data requirements for probabilistic certification and for insurance, showing that we can exploit the data for both purposes and thereby enable the dual risk-reduction mechanism.

Certification

Certification consists of having a trusted authority attest that a component possesses a particular property [3]. We take a descriptive rather than the more common normative view of certification. In the descriptive view, the certified component property will be of the form c.X, where X is a property of component c, rather than the form c.satisfies(X), where X is a norm that must be satisfied (e.g., [11]). Our goal is to certify such descriptive properties and to use these properties to predict the behavior of the components when they are used in assemblies.

Descriptive component properties fall into two broadly defined classes: formal properties and empirical properties. A model of component behavior, such as might be specified in Milner's π-Calculus [6] is a formal component property. Our concern in this position paper is with empirical properties. Empirical properties are characterized by a measure of component behavior, such as throughput, responsiveness, latency, and reliability. Empirical properties inherently require experimental observation and measurement.

We illustrate our point by using component latency as a certifiable property. We believe the following statistical models can apply to any empirical component property. Hissam et al. provide further details on the illustration in [5]. In this example, we certify the latency of components and the predictive model used to relate component latency to assembly latency, which we believe will be of engineering value for insurers.



- The left model describes the measured latency of a particular component in terms of a confidence interval. It states that we can have .95 confidence that the mean latency of a sample of measurements of this component will fall within 0.046 ± 3.4E-05 seconds.
- The model on the right describes the quality of predictions in terms of a one-tail tolerance interval. It states[1] that, for this particular prediction model, the mean magnitude of relative error (MRE) of latency predictions is less than ~2%, and that the relative error of 9 out of 10 predictions will be less than ~6.3%, and that we can have .95 confidence in this upper limit.

The details of how these models are constructed and interpreted can be found in various texts on statistical measurement theory [4,8]. For certification we are interested in the regions that lie within the confidence and tolerance intervals, which provide a measure of confidence in component properties and related design predictions. For insurance we are interested in the regions that lie outside of these intervals, for these regions quantify the probability of a failure in our prediction.

Insurance

Insurance is a means of recovering the financial losses of policyholders after the occurrence of an undesirable event [1]. This is a system that has been shown to be a useful risk management tool in other areas. The insurance companies have methods such as pooling and reinsurance that might help to make software insurance systems viable. Insurance principally for software is not available today except to the extent that a business or system policy subsumes it. As companies are purchasing and reusing software components for business critical services, risk and

[1] The interpretation of this model requires taking the inverse BoxCox transformation of the data depicted in the figure. This transformation was needed to convert the data to normal form for interval calculation.

accountability issues regarding software might become dominant. We suggest that an insurance system specifically addressing the risks associated with software systems is needed.

Insurance is based on mathematical models for on random variables. The first is the occurrence rate and the second is the loss amount [7]. The latter can be dealt with by asking decision makers to quantify their losses. The former is more challenging and the focus of this paper. Traditional occurrence models are based almost entirely on empirical data, by observing a large number of events over time. The same process seems daunting for software for several reasons:

- Observation time – Software's rapid release cycles limits the amount of useful observational data that can be collected in a timely fashion and the duration in which it is relevant.
- Environmental flux – The computing environment and myriad interacting systems are so variable and changing that observations made in one specific setting are of little predictive value.
- Deterministic behavior – Code is deterministic. If it fails, it will always fail. This takes away the stochastic occurrence properties which insurance depends on.

Suppose we wanted to insure the latency of a component. The component might have been used in several projects, but not enough to provide a statistically significant sample size. Even if enough samples are available, the setting (e.g., usage case, hardware, operating system, other installed software, etc) might differ so drastically between samples that knowing a component's latency in a prior sample might not contribute to the confidence in its behavior in the current sample.

Data

We suggest that by conducting testing in controlled environments and measuring MREs of different assemblies, certification may address the technical concerns of insurance. Computer components, unlike traditional targets of insurance, can base predictions on the number of executions. Certification can construct and test many different assemblies, providing statistically significant data regarding the influence of various environmental factors on the behavior of the component and the relative errors between the assemblies. A prospective IT consumer can then run an analysis tool in their own setting, with the analysis tool identifying the specific setting of usage, to obtain a certified prediction of the normative behavior of the component in the environment of the prospective customer. This active method of generating data accounts for environmental flux and solves the observation time problem by producing the necessary data.

The final concern of deterministic behavior is a non-factor. While code is certainly fixed, the execution path and system state are not. In real systems the complexity of interactions, especially timing conditions, produce something very close to non-determinism Therefore, reaching the exact set of circumstances necessary for the fault to be duplicated, we argue is still a stochastic process.

To be a useful attribute, certified and predicted values (latency, in the example) must sufficiently approximate actual values. Thus, certifiable properties are specified using a confidence interval. For example, latency for a component could be found to range from $X-\alpha$ to $X+\beta$ with 99% confidence. Insurance actually uses the inverse of the probability expressed by certification. While certification specifies how often a value will be in a specified range, insurance is for those times when the value falls outside of the specified range, i.e. the 1% of the time when the value is less than $X-\alpha$ or greater than $X+\beta$.

An insurance system can provide feedback for certification. One of the issues in certification is to identify factors that affect predictions. The same is true for insurance. Since insurance claims provide samples of values outside of the desired range, analysis can identify factors that contribute to deviations. In addition to identifying risk pools for insurance, the same information could assist certification to classify different assemblies and to produce models that better predict system behavior.

Conclusion

Certification and insurance are two mechanisms that offer prospects for controlling risks of software reuse investments. Certification describes the attributes of the component in an assembly, while insurance provides compensation should a component fail to behave properly. The two processes compliment each other. The data produced by a certification process may be useful for producing the necessary model for an insurance system, and insurance claims could find characteristic to analyze in the certification process. If these methods were in place, a company could conceivably better analyze and manage its reuse investments.

While normative certification is not a new idea, descriptive certification is largely unexplored, and software insurance is non-existent. A major roadblock to certification is the cost associated with the collection and analysis of data. One impediment to a software insurance system seems to be the lack of a predictive model. We see the prospect of a symbiotic relationship between these two areas. Certification alone might not justify the costs of analyzing components and collecting data, but if a profitable insurance system could also result from the analysis conducted, then the joint benefit of the two methods might justify the investment in analysis.

Acknowledgements

This research was supported by the National Science Foundation under Grand CCR-0086003, by the Carnegie Mellon Sloan Software Center, and by the Software Engineering Institute at Carnegie Mellon University.

The authors wish to thank Gabriel Moreno for his help with statistical models for descriptive certification.

Bibliography

1. Newton L Bowers Jr., Hans U. Gerber, et al. *Actuarial Mathematics*. The Society of Actuaries, Schaumburg, Illinois. 1997.
2. Christine L. Braun. “A Lifecycle Process for the Effective Reuse of Commercial Off-the-Shelf (COTS) Software.” *Proceedings of the Fifth Symposium on Software Reusability*, May 1999. Pp29-36
3. Janet Flynt and Laura Elan. “Software Conformity Assessment.” Underwriter Laboratories , see < http://www.ul.com/pscs/SoftwareConformityAssessment.pdf>
4. Gerald Hahn, William Meeker, *Statistical Intervals A Guide for Practitioners*. John Wiley & Sons, 1991.
5. Scott Hissam, Gabriel Moreno, Judy Stafford, and Kurt Wallnau. “Packaging and Deploying Predictable Assembly.” to appear in the *Proceedings of the First International IFIP/ACM Working Conference on Component Deployment*, Berlin, Germany, June 20-21, 2002.
6. Robin Milner. *Communicating and mobile systems: the π-calculus*. Cambridge University Press, 1999.
7. Harry H. Panjer.(Editor). “Actuarial Mathematics.” *Proceedings of the Symposia in Applied Mathematics*, Volume 35, 1985. American Mathematical Society.
8. Semon G. Rabinovich. *Measurement Errors and Uncertainties Theory and Practice*. Springer AIP Press, 1999.
9. David Rine. “Success Factors for Software Reuse that are Applicable Across Domains and Businesses.” *Proceedings of the 1997 ACM Symposium on Applied Computing*, April 1997. Pp182-186
10. Karma Sherif, Ajay Vinze. “A qualitative model for barriers to software reuse adoption” *Proceeding of the 20th international conference on Information Systems*, January 1999. Pp47-63
11. TCSEC 85 : Dept. of Defense Standard, Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200.28-STD, GPO 1986-623-963,643 0, Dec. 26, 1985.