Adapted Statistical Usage Testing: A Case Study

Peter B. Lakey
Cognitive Concepts, St. Louis, September 2002
www.cognitiveconcepts.org, plakey01@earthlink.net

I. Introduction

This paper provides a summary of the experience and lessons learned when applying statistical usage testing (SUT) to the U.S. Army's Crusader program. Even though the program was discontinued in August, 2002, significant results were obtained that are worthy of reporting to the software reliability engineering community.

The cancelled Crusader was to be a technically advanced, fully automated artillery system. Due to the high degree of automation and the substantial portion of the system to be controlled by software, the Army's procurement agency required a software reliability assessment capability.

United Defense Industries (UDLP), Crusader's prime contractor, formed a subcontract with Cognitive Concepts (CC) to lead the software reliability support effort. Based on knowledge and experience, CC recommended the SUT methodology be used to assess the reliability of the tactical software. CC believes that SUT is the most fundamentally sound method available today for certifying the reliability of systems with safety and mission-critical software [1].

In statistical usage testing, behavioral models of the usage of a software system are constructed. Then a large sample of tests are randomly generated from the models. After the sample is executed and results obtained a project can determine, with a high degree of confidence, how reliable the software is as measured during system test.

II. Pilot Application

At the end of 2000 UDLP established a pilot project to evaluate the feasibility and value of applying SUT to the Crusader tactical software. The pilot ran for five months. During this time the pilot team created usage models of the basic administrative function of the self-propelled howitzer (SPH). The functionality included *initialization*, vehicle *movement*, and *shutdown.* Models were constructed by analyzing requirements (use cases) for those operations.

The usage model information was imported into a software program called toolSET_Certify™, which supports the entire SUT process. Random sets of test cases were generated from the tool for each model. Because formal testing was not planned until 2002, the tests were never executed. However, for purposes of the pilot, it was concluded by the team's manager that if the tests had been run, the data required to evaluate software reliability would have been sufficiently available.

The objectives of the pilot were met. Usage models were constructed within a reasonable period and random tests were available to assess reliability. Therefore, the pilot was declared successful and SUT was adopted by UDLP.

III. Full Scale Implementation

Over the subsequent 14 months, starting in June, 2001, Cognitive Concepts worked very closely with UDLP to model the critical tactical functions of the Crusader system software. The objective was to have an initial set of models ready to evaluate Release 6 (R6) software in late 2002. The testing never came to pass due to cancellation of the program. But usage models were constructed and several important lessons were learned along the way.

The tactical usage model development process began with CC participating in detailed reviews of the system software requirements for R6. The majority of the requirements were documented in the form of use case and activity diagrams in Rational Rose®. In November CC started to create models for each major function targeted for R6.

During the pilot all usage states were documented for each model in a Microsoft Word table. The tables were then converted to text files and the text files were imported into toolSET_Certify™. Finally, random test cases were generated from the Certify models. That process worked fairly well for the pilot models which never exceeded a few thousand states, although the state table creation process was very labor intensive as well as error-prone.

Analysis of the scope of the requirements and complexity of the behaviors for R6 revealed that the usage models would be at least an order of magnitude larger. It quickly became clear that the pilot process would fail on a larger scale. Perhaps more importantly it was determined that toolSET_Certify™ would no longer support the needs of the larger usage models. It would simply require way too much time and effort to maintain models in the tool.

An adapted version of the statistical usage testing process was created by CC for the Crusader program in order to handle the high level of complexity associated with the tactical software behavior. Usage models would become so large that the traditional SUT methodology was no longer practical, or even feasible, for documenting models. In addition, a new software program, called the *generator*, was developed by UDLP to take the place of Certify and fully support the new requirements of the adapted SUT.

The adapted SUT process evolved during the construction of usage models for the Crusader *tactical move* operations. During that implementation a usage model architecture was constructed that allowed independent behaviors to be distributed among several interacting usage sub-models. By isolating these behaviors it was possible to make several models instead of a single, unmanageable model. For tactical move this allowed the total number of usage states to be reduced from roughly 100,000 to around 10,000.

IV. ADAPTED SUT PROCESS

The adapted SUT process is defined and illustrated below with an example extracted from the Indirect Fire function.

The first step of the adapted SUT process is to create a usage model architecture for a function to be represented. This task involved identifying all use cases that may pertain to Indirect Fire and their corresponding behaviors. The analysis led to the Figure 1 architecture. The main model was further decomposed until there were a total of 19 independent usage models to be constructed.

With the list of models identified, the traditional SUT steps for usage model construction [2] were followed for each. These steps include identifying usage variables, assigning usage variable values, and identifying stimuli for a model. The usage variables and corresponding values determine the number and composition of usage states. The stimuli dictate the legal transitions between usage states. The more legal transitions, the larger the usage models.

Next the new steps defined for the adapted SUT process were performed. The primary tasks are creating rules for determining the legal set of model states and rules for the legal application of stimuli from each state in the model. Usage model state tables were created by first generating a Cartesian product of all combinations of values for each usage variable in a Microsoft Access database (some had products larger than 1 million entries). This product was then reduced by applying state rules, stored as queries in Access. After the state reductions, the total number of states for the combined 19 models turned out to be 24,312.

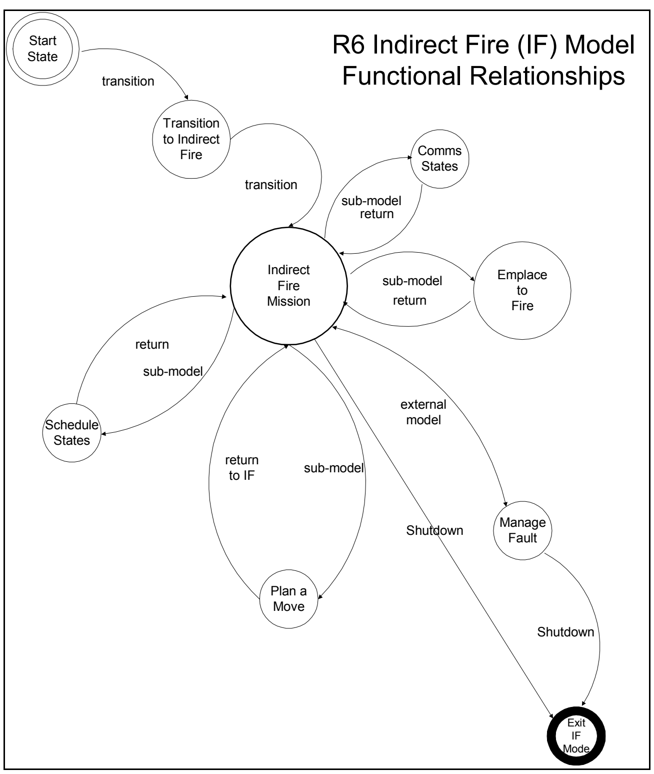

Figure 1. Indirect Fire Function Sub-model Architecture

Stimulus rules, also encoded as Access queries, were then created and applied to each reduced model. The total number of transitions for the 19 models was 303,131. Thus, the Indirect Fire usage models are very large, even after many simplifying assumptions were incorporated.

The accuracy of the IF usage models could not be verified until many random walks from the *generator* tool were evaluated. Analysis of the initial sample revealed a few situations where behaviors were incorrect. These were corrected and 5000 random walks generated without error.

A sample output from the tool is shown below; it shows the sequence of stimuli for the random walk and the relative probability of that stimulus being selected. These frequency values were generated automatically in Access by applying a simple set of stimulus probability rules.

Stimulus	Frequency
<Gunner Selects Planned Fires>	0.037358
<Send K02.12 Command to Fire (FPF)>	0.064918
<Gunner Selects Fire Enable Button>	0.720000
<Load Gun>	0.330000
<Fire Gun>	0.344348
<Load Gun>	0.330000
<Send K02.40 Cease Load>	0.019800
<Critical Fault Detected >	0.005000
<Shutdown Vehicle >	1.000000

V. CONCLUSION

To overcome the inherent problem of developing usage models for software systems with complex behaviors CC was effectively forced to invent new approaches to make the SUT methodology viable. The adapted steps are a major leap forward for usage modeling; the adaptation brings SUT closer to realism for many applications.

During the Crusader project CC has made four advances to the improvement, simplification, effectiveness and efficiency of the usage modeling process: 1) identification of a usage model architecture; 2) transparency of usage variables; 3) automatic generation of state tables; and 4) automatic generation of transition probabilities. To take full advantage of these improvement, UDLP developed the generator tool for random walks through the models.

The adapted SUT process worked flawlessly when applied to the Crusader Indirect Fire function. The complete set of models resulted in roughly 23,000 states and 300,000 arcs. From beginning to end the process required 5 weeks of labor by one analyst. Without the adapted process and the generator tool, creating a set of models this size would have conservatively taken 4 months, with far more errors.

REFERENCES

[1] Poore, J. H. and C. J. Trammell, ``Application of Statistical Science to Testing and Evaluating Software Intensive Systems'' a chapter in Statistics, Testing, and Defense Acquisition,1998.

[2] Software Engineering Institute technical report CMU/SEI-96-TR-022 section 4: Usage Modeling and Test Planning Process.