Ranking of Architecture Support for Testing (AST) Model Problems

March 1, 2011

The model problems produced by our workshop were ranked by 10 (out of 11) of our practitioner guests by the requested deadline. The SEI workshop facilitators did not vote, nor did our guests from the research community. For each scenario, respondents ranked the scenario according to the following scale:

- VH (Very High) = 5 (meaning that the respondent places a very high value on this capability)
- H (High)= 4
- M (Medium) = 3
- L (Low) = 2
- VL (Very Low) = 1 (meaning that this is a capability that is not at all valuable to the respondent)

Any scenario not ranked by a respondent was scored as "VL (Very Low)" on that respondent's form.

The 29 model problems produced in our workshop are listed in the attached table, in the order they were sent out. Each scenario is annotated with its average ranking according to the scale above. The scores ranged from 4.22 (scenario REQ1) to 2.56 (scenario INT16).

Now these problems and their rankings will be sent to our colleagues Henry Muccini and Antonia Bertolino, where they will serve as input to researcher's workshop in late March.

Thanks once again for your time, energy, and help.

Best regards,

Paul C. Clements

AST Model Problem Ranking

SCORE	4.00 or more 3.00 – 3.99 2.00 – 2.99	Model problem scenario • Source: Which stakeholder initiates the action? • Stimulus: What is the action? • Environment: What's going on at the time? What relevant conditions are in play? • Response: What is the desired outcome of the story? • Response measure: What quantitative measure tells us the activity was successful?

AST and Requirements

4.22	Scenario name	REQ1
	Source	Tester
	Stimulus	A tester chooses a test set to test the system for requirements satisfaction.
	Environment	The architecture is complete. System test has not yet begun.
	Response	The tester uses an architecture analysis tool that identifies the smallest number of tests to run to provide coverage of 98% of the requirements. Redundant tests are eliminated.
	Response measure	Performing the analysis is much less costly and time-consuming to run than the tests it replaces.
	Original notes from workshop	Can we use the architecture to tell us that (out of all of the huge number of tests possible) if we run a small number of tests, we will have covered 98% of the requirements?
3.89	Scenario name	REQ2
	Source	Tester
	Stimulus	A tester chooses a test set.
	Environment	The architecture is complete.
	Response	The tester uses an architecture analysis tool that identifies the smallest number of tests to run to provide coverage of the highest-risk areas of the system. Redundant tests are eliminated.
	Response measure	Performing the analysis is much less costly and time-consuming to run than the tests it replaces.
	Original notes from workshop	Can I use architecture to use tests to cover high-risk or high-probability-of-error areas in the system? • Design for testability… of the architecture to meet *those* requirements (encapsulation, isolation…) • Predictive analysis… to analyze for satisfaction of *those* req's. • Generation of test artifacts… to test for *those* requirements • Conformance… is necessary for the other three to work.
2.89	Scenario name	REQ3
	Source	Unit testers
	Stimulus	Unit testing of a module begins
	Environment	Modules are coded, not yet integrated. Each module's responsibilities with respect to requirements are documented.
	Response	Analysis is run on individual modules to find errors with respect to the requirements.
	Response measure	Cost of performing the analysis is very low, and the results save significant project cost and re-work.
	Original notes from workshop	Can I use AST to help discover design and coding errors earlier in the development cycle (errors w.r.t. the requirements)?
3.11	Scenario name	REQ4

	Source	Stakeholder who provides new requirements
	Stimulus	New requirements have been proposed for a fielded system.
	Environment	Requirements analysis phase of a new version of a system
	Response	Analysis of the requirements yields a report on the architectural changes the new requirements will necessitate, as well as a report on the cost of making those changes and making the corresponding changes to the implementation.
	Response measure	The cost of performing the analysis is very low.
	Original notes from workshop	Requirements and predictive analysis: Predicting the impact of future requirements on the architecture and the system. This is an alternative to making the changes, testing them, and hoping for the best. (This is an alternative to exhaustive testing.)
3.44	Scenario name	REQ5
	Source	Tester
	Stimulus	System testing is about to begin.
	Environment	The system, when fielded, will consist of a very large number of nodes (for example, every soldier in a division may carry a node, as well as all of the division's vehicles), a prohibitively large number for a test and integration lab to handle.
	Response	The system is tested in the test and integration lab using the nodes that are there. Analysis is performed to give high confidence that this testing is sufficient, and the system will perform correctly in the field.
	Response measure	The cost of the analysis is low; the confidence measure it provides is very high.
	Original notes from workshop	Suppose the requirements call for a huge number of (nodes, users, servers, …) and this cannot be lab-tested. How can AST help to set "lab limits" to test what can be tested, and set a level of confidence in the full-scale system?

AST and Product Lines

3.78	Scenario name	SPL1
	Source	A stakeholder who wants a change to product B.
	Stimulus	A change request, which will result in core assets being modified, is initiated for product B, which is a certified product.
	Environment	The products in the product line must be certified to a specified standard. Core assets have been used as part of the product implementation. Several products have used these core assets. There is a policy that if any core asset changes, all products that depend on that core asset are rebuilt.
	Response	The changes to the core assets are made and the products, A and B, dependent on those assets, are rebuilt using approved tactics. An architecture-based analysis proves that no recertification of A, which only has new implementations of the same core assets, is needed. Evidence accumulated during the changes is used to expedite the recertification of B.
	Response measure	The effort to recertify B is significantly less than the effort to certify B originally while maintaining the level of confidence.

	Original notes from workshop	Testing Certified Product Line Assets : Can AST help to reduce certification costs for products by identifying and managing certifications levels of Core Assets?
		• Change Core or Variant Assets for next Product B without affecting certification of Product A
		  ○ Cost is an issue
		  ○ Risk to product delivery
		  ○ Takes too much time to perform testing for delivery
		• What does it take to certify a parent product having capability and then apply certification to other products?
		• Product A has completed the certification and a problem is found in a separate product. How can we prevent certification process if Product A needs to pick up change?
		• Does early or late binding help in reducing certification costs?

3.78	Scenario name	SPL2
	Source	The testing process
	Stimulus	Integration test suites must be selected for products in the software product line.
	Environment	Products in the product line are defined in terms of features. All core assets are thoroughly unit tested.
	Response	An architecture-based product test suite construction algorithm analyzes the features, combinations of features, and dependencies among features that define each product. The algorithm uses optimization techniques to select the smallest number of products to cover all of the features.
	Response measure	The percentage of products that have to be tested to achieve complete feature coverage is much less than an exhaustive test suite but how much less will vary based on the degree of similarity among the products.
	Original notes from workshop	Configuration Complexity and Management: Given the Core Assets are utilized in 200 products, what is the smallest number of products which can be tested to provide "complete" feature coverage. • With all of the variability supported in the Core Assets, AST should help to identify the possible test combinations given the Core Assets chosen for a product and identify high risk features? • Add configurability and then pay the price! In testing as well • Looking at the superset of the problem space • Trimming the superset to a subset for testing is non-trivial • Feature management difficulties lead to difficulty in managing assets • Slicing tests by feature can be difficult, approach is to make test suites organized by core and by features, not all of us try this
3.56	Scenario name	SPL3
	Source	Stakeholder who initiates new capability
	Stimulus	A new capability, which spans multiple subsystems, is added to the product line.
	Environment	An operating software product line organization has a validated architecture and functioning products. The new capability impacts multiple core assets and requires the addition of new assets.
	Response	An architecture-based unit test suite construction algorithm analyzes the core asset interfaces and their dependencies in the architecture to specify a sufficient set of unit tests to achieve an acceptable level of coverage. An architecture-based integration test suite construction algorithm identifies the smallest set of existing core assets that are needed to combine with each new core asset to provide acceptable integration test coverage.
	Response measure	Both algorithms produce test suites that are significantly smaller than suites selected using non-architecture based methods while providing the same level of coverage.
	Original notes from workshop	Integration of Disruptive Capabilities and Features: A new capability is added to the product line which spans multiple subsystems. Portions of the new function span across several Core Assets. Insuring the new capability functions properly has difficulty because of ownership and integration issues. Need an tools/organizational model which helps facilitate creation and management of those tests. • New Capabilities / Core Assets added which were not part of the Change Scenarios of the original product and not considered in the original architecture. • Overall responsibility for testing features spanning subsystems • A feature applied across many components • New and unknown elements in the system • Test of the Superset (impossible) • Test of the Common Assets all together (impossible)
3.56	Scenario name	SPL4
	Source	Stakeholder who initiates new product

	Stimulus	A new product that was not originally envisioned is added to the product line.
	Environment	An operating software product line organization has a validated architecture and functioning products. The new product may require additional core assets.
	Response	If new core assets are created to contribute to the new product an architecture-based core asset testing algorithm is used to construct a test suite from existing and new tests. Then an architecture-based product test suite construction algorithm analyzes the core assets that make up the new product. It selects existing test cases that apply to the product in a first pass and then specifies additional new test cases, which will have to be constructed, in a second pass until a minimal set is achieved.
	Response measure	The percentage of the total tests that must be newly constructed is very low (but will vary depending on the percentage of core assets that are new to the product line).
	Original notes from workshop	Integration of Disruptive Capabilities and Features: A new capability is added to the product line which spans multiple subsystems. Portions of the new function span across several Core Assets. Insuring the new capability functions properly has difficulty because of ownership and integration issues. Need tools or organizational model to help facilitate creation and management of those tests. • New Capabilities / Core Assets added which were not part of the Change Scenarios of the original product and not considered in original architecture. • Overall responsibility for testing features spanning subsystems • A feature applied across many components • New and unknown elements in the system • Test of the Superset (impossible) • Test of the Common Assets all together (impossible)
3.00	Scenario name	SPL5
	Source	Architect
	Stimulus	A change is needed to achieve a different level of a non-functional requirement
	Environment	An operating software product line organization has a validated architecture and functioning products.
	Response	An architecture-based product test suite construction algorithm identifies and analyzes the changes made to the architecture. The algorithm analyzes the existing test suite and identifies changes that must be made to the test suite to achieve suitable coverage.
	Response measure	The percentage of the test suite that must be changed and executed is very low.
	Original notes from workshop	Changing the Architecture has minimal impacts to Tests: If the architecture changes for non-functional reasons, what are techniques to utilize in test development to minimize test modifications. • Change the variability mechanism to utilize components as opposed to an on/off switch, should not change the test
3.22	Scenario name	SPL6
	Source	Test manager
	Stimulus	A need to understand the effectiveness of the testing process
	Environment	An operating software product line organization has a validated architecture, functioning, tested products, and specified, documented test assets.
	Response	An algorithm queries the test reporting and defect reporting databases. The query retrieves test results and coordinates these with defect reports to determine how many tests are required to locate a defect. The defect yield, a measure of effectiveness, is reported.
	Response measure	The results of the defect yield analysis are sufficiently detailed so that differences in test effectiveness between two testing methods can be discerned.
	Original notes from workshop	Test Coverage of the Architecture and Product Line: An as-tested measure to understand the coverage of Core Assets and features of the architecture on an individual product and aggregated across products.

		• Measure of the effectiveness of the test development process? • Help identify areas where gaps may exist in test coverage • Aggregated is imperative
2.78	Scenario name	SPL7
	Source	Architect
	Stimulus	A design decision involving decomposition is being studied.
	Environment	An operating software product line organization has a validated architecture and functioning products.
	Response	An architecture-based analysis of the as-is architecture and the will-be architecture compares the testability of the two architectures and gives guidance on what decision to make.
	Response measure	The analysis is sufficiently sensitive that it gives a clear indication of which design is preferred a high percentage of the time.
	Original notes from workshop	Analysis of Test Complexity Incorporating process and tools: "Complexity of Test" Analysis tool and process given the Core Assets, code, interfaces, and the various options provided. • Test impact of architectural decisions to incorporate into the trade-off analysis of decomposition • Vary the decisions and be able to understand the impact on test ◦ The decisions being made are decomposition, asset selection, variability point selection, features ◦ Impact to multiple products must be considered

Scope of Architecture and AST

3.00	Scenario name	SCO1: Requirements with cross-cutting impacts
	Source	Users
	Stimulus	New requirements are precipitated that increase system load by adding more users and/or increasing transactions.
	Environment	System is under normal (runtime) operation. Large distributed database with many users (e.g. 10<x<5000), each user has a local workstation. Database is common to all users OR System is in the development phase (Where can 'testing' of artifacts effectively be done? On what artifacts? At what degree of confidence?)
	Response	AST techniques are used to analyze the new requirements for their impact on the architecture and its ability to support the new load while still maintaining required quality attributes.
	Response measure	The proper feedback occurs within an acceptable time.
	Original notes from workshop	• Requirements (e.g. multiuser, complex scheduling, deployment) that affect multiple quality attributes has to be reflected into the architecture. • How can specifications across multiple (model) representations be aligned (synchronized) and used as the basis for AST? • How can we analyze/test the architecture early with respect to the requirements and analyze the impact on quality attributes of interest (response time, integrity, etc)? • How can generation of test cases touch on all the components (interfaces?) that are affected by the requirements for multiple views throughout the lifecycle?
3.11	Scenario name	SCO2: Changed requirements with cross-cutting impacts
	Source	Requirements team or user community
	Stimulus	Make the system multiuser (with up to 1000 users), necessitating a change to the architecture.
	Environment	System is in the design phase
	Response	AST techniques are used to plan regression testing.
	Response measure	The proper feedback occurs within a specified acceptable time.

	Original notes from workshop	• Requirements ***change*** (e.g. from single to multiuser, change in scheduling, new deployment) that affect multiple quality attributes has to be reflected into the architecture. • How can changes across multiple (model) representations be captured and used as the basis for regression AST? • How can we analyze/test the new architecture satisfies the requirements and analyze the impact on quality attributes of interest (response time, integrity, etc)? • How can selection of existing test cases and the generation of new test cases be driven by the requirement changes and their impact on the architecture?

AST and Integration Testing

3.22	Scenario name	INT1
	Source	Test Designer
	Stimulus	A test designer is creating tests for the system
	Environment	The architecture is complete. The system has multiple modes of operation.
	Response	The test designer uses a tool to generate a set of tests that adequately exercise the different system modes
	Response measure	Adding a test to the generated set doesn't uncover any new faults, but with confidence that the generated test set covers all modes.
	Original notes from workshop	Assuming an architecture can represent different system modes, can that representation be used to determine an "optimal" set of integration tests?
3.22	Scenario name	INT2
	Source	Architect
	Stimulus	An architect is structuring the system's mission threads to support incremental integration
	Environment	The architecture is complete. The system is being integrated incrementally
	Response	The system's mission threads are tagged with the components necessary to support the thread
	Response measure	A tester can use a definition of what has been integrated to determine the subset of mission threads that can be tested
	Original notes from workshop	How can architecture help express threads so that the system can be integrated incrementally?
3.67	Scenario name	INT3
	Source	Test Designer
	Stimulus	The test designer must determine the order in which to test the system's mission threads
	Environment	The architecture and system integration are complete
	Response	The mission threads are prioritized according to criticality
	Response measure	Testers can use the prioritized list to ensure that all critical threads are tested even if integration test time is cut short
	Original notes from workshop	Can the architecture be used to identify critical threads and predict a sequence of integration tests?
3.67	Scenario name	INT4
	Source	Tester
	Stimulus	The tester seeks to uncover performance bottlenecks in the integrated system as quickly as possible
	Environment	The architecture and system integration are complete
	Response	An analysis of the system architecture identifies the components and threads most likely to cause the system to fail in terms of its performance thresholds. The analysis could also be used to predict the components that will cause performance issues when integrated into the system.

	Response measure	Performance problems are discovered early and not late in the testing cycle
	Original notes from workshop	• Can the architecture be used to predict and define the bottlenecks within the system and then be used as a basis for integration testing? • Utilization and resource availability can be bottleneck predictors • Capturing sufficient performance information to determine high transaction components • Predicting integration risk
3.11	Scenario name	INT5
	Source	Tester
	Stimulus	The tester is determining whether the system meets performance requirements
	Environment	The architecture and system integration are complete. The architect has defined a complex relationship between individual performance parameters
	Response	The tester is able to analyze the relationship determining appropriate loadings of the separate performance parameters in order to provide confidence that overall system performance will meet the defined relationship
	Response measure	Testing is accomplished with a small number of test sets ensuring, with confidence, that the performance parameters can be achieved.
	Original notes from workshop	• How can the architecture represent the detail of n messages and m calls into a general throughput capability (n and m may vary, but their combination should be within some value)? • Can the architecture be used to predict maximum capabilities?
3.44	Scenario name	INT6
	Source	Test designer
	Stimulus	The designer is creating cases to provide confidence in the system integration
	Environment	The architecture is complete
	Response	The designer uses some form of architectural analysis to develop a definition of observable (and controllable) entities within the system.
	Response measure	The analysis takes less time than developing tests (providing similar confidence in the system integration) manually
	Original notes from workshop	How does the architecture define what things should be observed?
3.78	Scenario name	INT7
	Source	Test designer
	Stimulus	The test designer must create a set of tests for system integration
	Environment	The architecture is complete
	Response	The designer uses an automated tool that analyzes the architectural descriptions of the components and generates lists of observable and controllable entities
	Response measure	The tool generates 100% of the observable entities provided by the components within the system
	Original notes from workshop	How can a module definition describe what can be observed?
3.33	Scenario name	INT8
	Source	Test designer
	Stimulus	The designer determines what observation points to add to the system components
	Environment	The architecture is complete but development has not begun or is only partially complete
	Response	An automated (preferably) analysis of the mission threads determines observation points in specific components that will provide most benefit to the testers. The resultant test points are then included in system development
	Response	Testers get the data necessary to ensure that the system components are appropriately

	measure	integrated with fewer test points than expected without mission thread analysis.
	Original notes from workshop	Can the threads be used to determine the optimal location of observation points?
2.89	Scenario name	INT9
	Source	Tester
	Stimulus	The testers need to ensure that the integrated system isn't violating privacy policies
	Environment	The architecture is complete and contains details with respect to data privacy
	Response	An analysis of the architecture determines a set of tests with respect to data privacy; such tests would define roles and their access to sensitive data
	Response measure	Tests generated from the architecture lead to greater confidence that the system will not inadvertently disclose private data
	Original notes from workshop	• How do we express the privacy concerns so that data is only exposed to appropriate people? • How can privacy policies be communicated?
2.67	Scenario name	INT10
	Source	Test designer
	Stimulus	The test designer wants to test that certain behaviors cannot occur within the system
	Environment	The architecture is complete and contains information on threads and behaviors that the system must not be able to perform
	Response	Analysis of the architecture derives tests for the known to be forbidden behaviors
	Response measure	The test suite tests for all behaviors known to be forbidden as well as those the system should exhibit.
	Original notes from workshop	How can architecture be used to define what cannot be permitted? example, opened up communication paths for one application, but not for all of them
3.22	Scenario name	INT11
	Source	Tester
	Stimulus	The tester needs to test a variety of similar interfaces
	Environment	The architecture and system integration are complete
	Response	The tester is able to automatically generate tests for a variety if instances of a given interface using a single test set and generated test stubs
	Response measure	The test generator can generate 90% of the tests needed to test the different interfaces
	Original notes from workshop	How can the architecture describe the variety of possible interfaces from the legacy systems in one area?
3.00	Scenario name	INT12
	Source	Tester
	Stimulus	The tester must test the integrated system in a variety of different environments
	Environment	The architecture is complete and the system is integrated; the architecture describes known variations in the operational environment
	Response	An analysis of the architecture enables the tester to simulate the environmental variations (differences in communicating systems, platforms, …) and test the system behavior in a sufficient number of environments in order to gain confidence in correct execution. Ideally, the test set will be optimized to provide sufficient coverage without testing all environmental variations
	Response measure	Confidence that the integrated system will operate correctly is obtained without testing all environmental variations
	Original notes from	Can the architecture describe the existing environment well enough to understand the complexity?

	workshop	
3.33	Scenario name	INT13
	Source	Budget team
	Stimulus	The budget team need to prepare estimates for the cost of integration testing
	Environment	The architecture is complete
	Response	Automated analysis of the architecture predicts cost and schedule associated with integration testing
	Response measure	Predicted costs are within 10% of the actual costs of integration testing
	Original notes from workshop	Can the architecture be used to predict the cost of integration testing? Can the architecture description of the business environment/business drivers be used to prioritize and assign effort to integration testing? Can value based testing be applied to the architecture? (Merged with INT 14, as they were nearly identical.)
N/A	Scenario name	INT14: Deleted. Merged with INT13.
3.89	Scenario name	INT15
	Source	Tester
	Stimulus	The tester needs to develop integration tests
	Environment	The architecture of the system is complete and detailed
	Response	Automated tools use the architecture to generate test cases and test subs for the components within the system
	Response measure	The tester asserts that 90% of the generated tests should be run
	Original notes from workshop	How can architecture be used to drive integration testing? Generation of test stubs from high level descriptions (Common Information Model CIM) Generation of test cases from more detailed descriptions (Multispeak) Keyword-based test automation – use of domain specific language. Can we use the domain specific language in the architecture models so that we can test sequences of the keywords… basis for auto-generation of tests
2.56	Scenario name	INT16
	Source	Testers
	Stimulus	A tester wants to perform integration testing for a component into a running system
	Environment	The architecture of the runtime environment (system of systems) is complete and up to date. The operational system must continue to operate
	Response	The tester adds the component to the running system and performs necessary tests on it
	Response measure	Production instances of component respond to the tests without polluting live data with the test data from the component in question
	Original notes from workshop	• How can the architecture help us create an environment where new components are integrated in production/operation? • Data can be tagged as test, meaning that some, but not all actions are taken

(end)