# Rapid and Adaptive End-to-End T&E of Systems of Systems (SoS)

Dr. Raymond Paul

OASD(C3I)

Washington, DC

# Contents

- E2E Test Background
- Systems of Systems (SoS) and the E2E Test Concept
- E2E Trial: E2E Benefits in the VSoS Exercise
  - Operational Scenario Specification and Analysis,
  - Economical Test Automation for Increased Coverage,
  - Test Reuse and Regression Testing,
  - SoS Simulation.
- Conclusion

# Background of E2E T & E

- DoD Y2K testing applied the Assurance-Based Testing (ABT) process for statistical confidence in test limits
- ABT uses the Howden formula:

$$C = 1 - (1-B)N$$

C: confidence level

B: threshold failure density

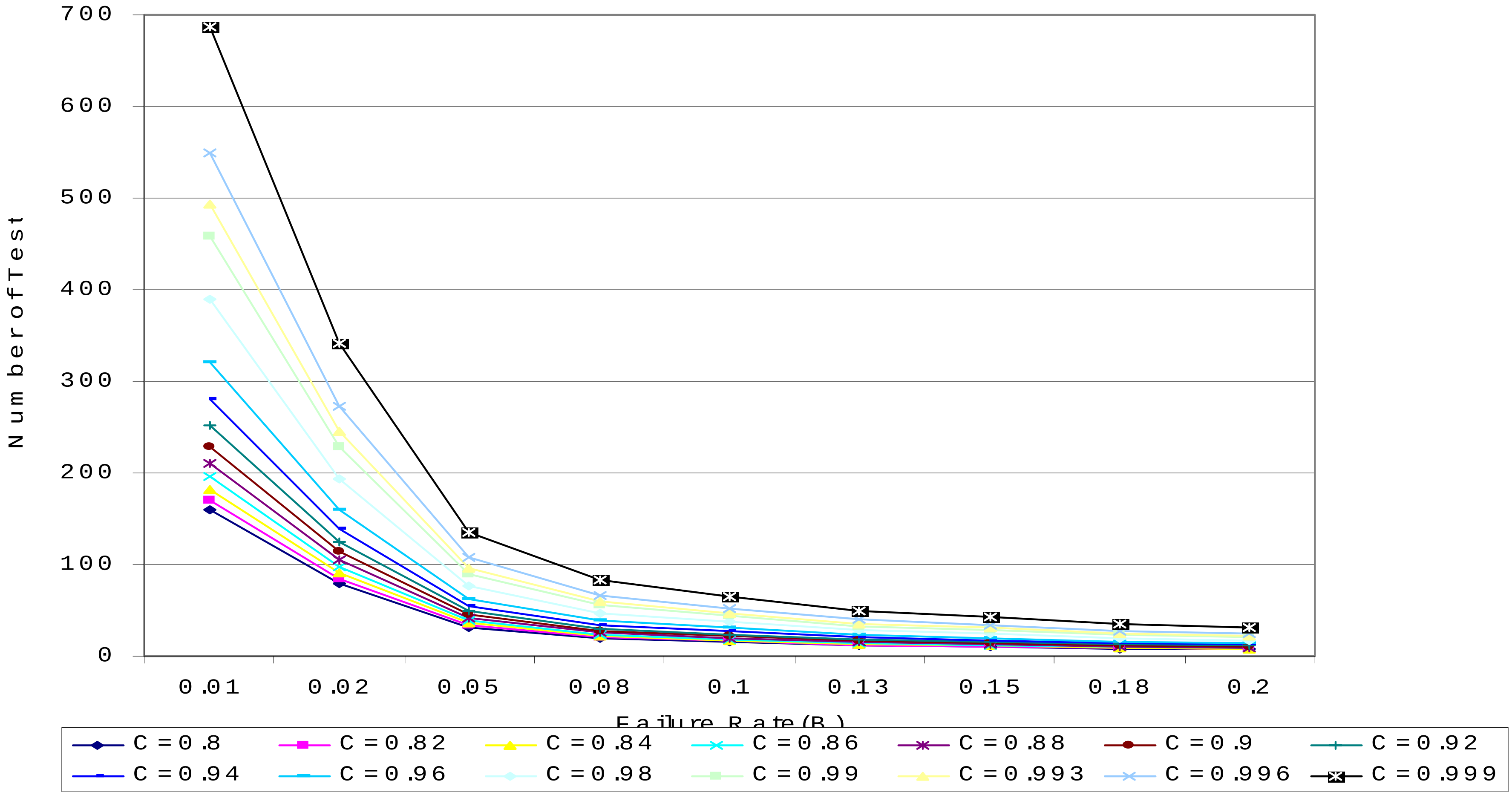
N: number of random cases run without failure

# ABT Statistical Model

$$C = 1 - (1-B)N$$

| Confidence Level(C) | Failure Density(B) | Number of test cases required |
|---|---|---|
| **0.8** | **0.01** | **160** |
| **0.8** | **0.02** | **80** |
| **0.8** | **0.05** | **31** |
| **0.8** | **0.10** | **19** |
| **0.95** | **0.01** | **298** |
| **0.95** | **0.02** | **148** |
| **0.95** | **0.05** | **58** |
| **0.95** | **0.10** | **28** |

# Modeling the Howden Formula



**Confidence is a function of test case numbers and target failure rates .**

# Can DoD T&E Support the SoS User?

*"Don't deliver complex systems until they have been tested under all possible operational scenarios that may occur during my mission."*

This is an impossible task with DoD's current T&E process!

# Additional User Concerns with a SoS

- Interoperability
- Collaboration and Interfaces
- Concurrency & Timing Constraints
- Functionality
- Reliability & Availability
- Fault Tolerance
- Security

# The E2E Test Process

- The E2E Test Process, with E2E Tool support, provides a realistic approach to achieve adequate test coverage of an evolving System of Systems (SoS)
  - System changes are inevitable and frequent.
  - New and legacy systems must be proven to work together before they are sent to the SoS user.

# End-to-End (E2E) T&E

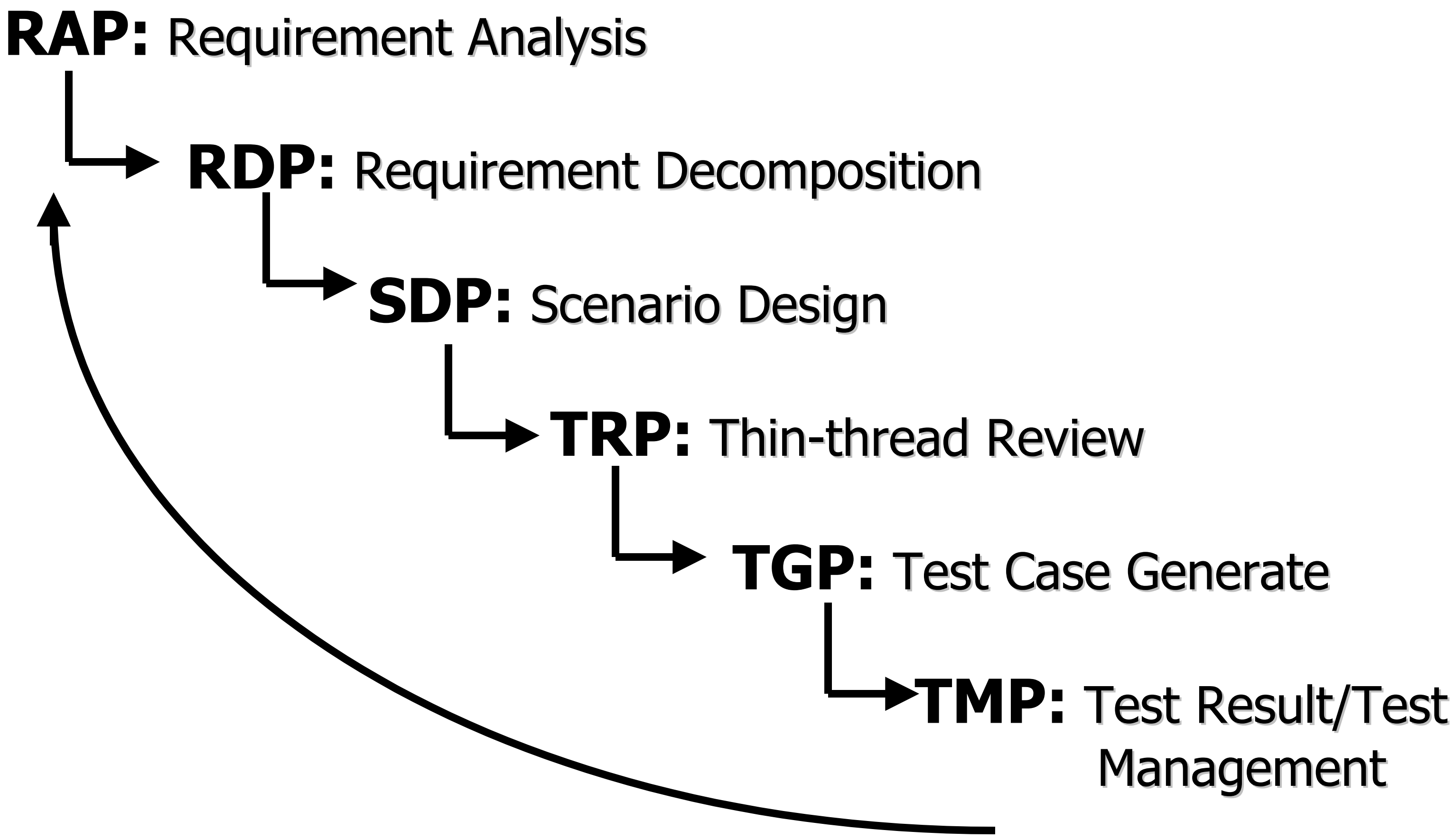
- E2E Test verifies *operational utility of the system to the user,* to ensure *operational and mission success* on the battlefield.
- A *partial* transaction (one or more systems executing within a SoS) is *not* an E2E function.
- Users need SoS that have been tested for *end-to-end* mission performance before delivery.
- A complete *end-to-end* transaction execution trace is called a *thin thread*.

# Thin thread definition:

A complete trace (E2E) of data/message using a minimally representative sample of external input data transformed through an interconnected set of system (architecture) to produce a minimally representative sample of external output data. The execution of a thin thread demonstrates a method to perform a specified function.

(U.S. Department of Defense Year 2000 Management Plan)

# The E2E Process Includes Various Activities



**RAP:** Requirement Analysis

**RDP:** Requirement Decomposition

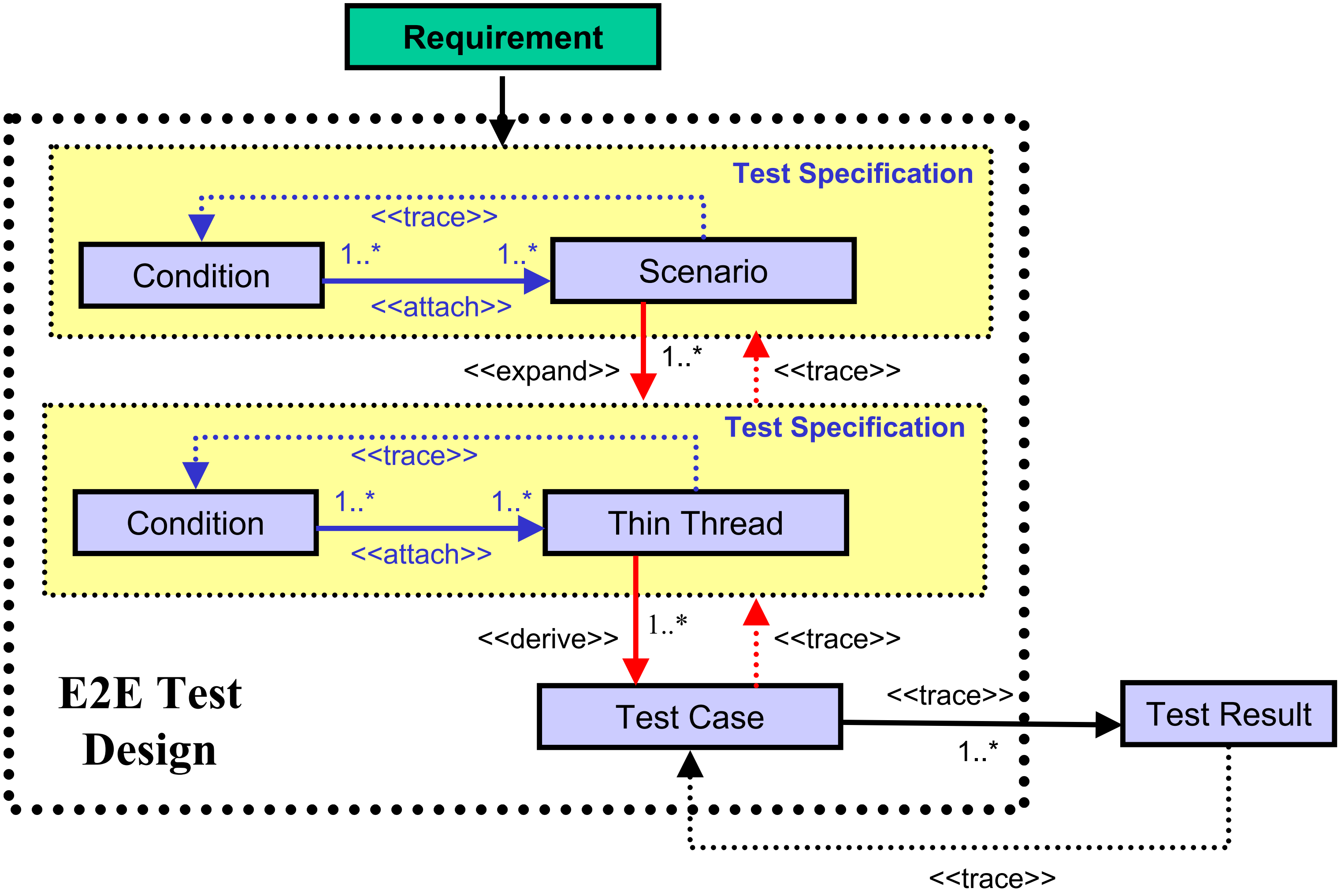
**SDP:** Scenario Design

**TRP:** Thin-thread Review

**TGP:** Test Case Generate

**TMP:** Test Result/Test Management

# E2E Test Design



# E2E Test Trial – The VSoS Exercise

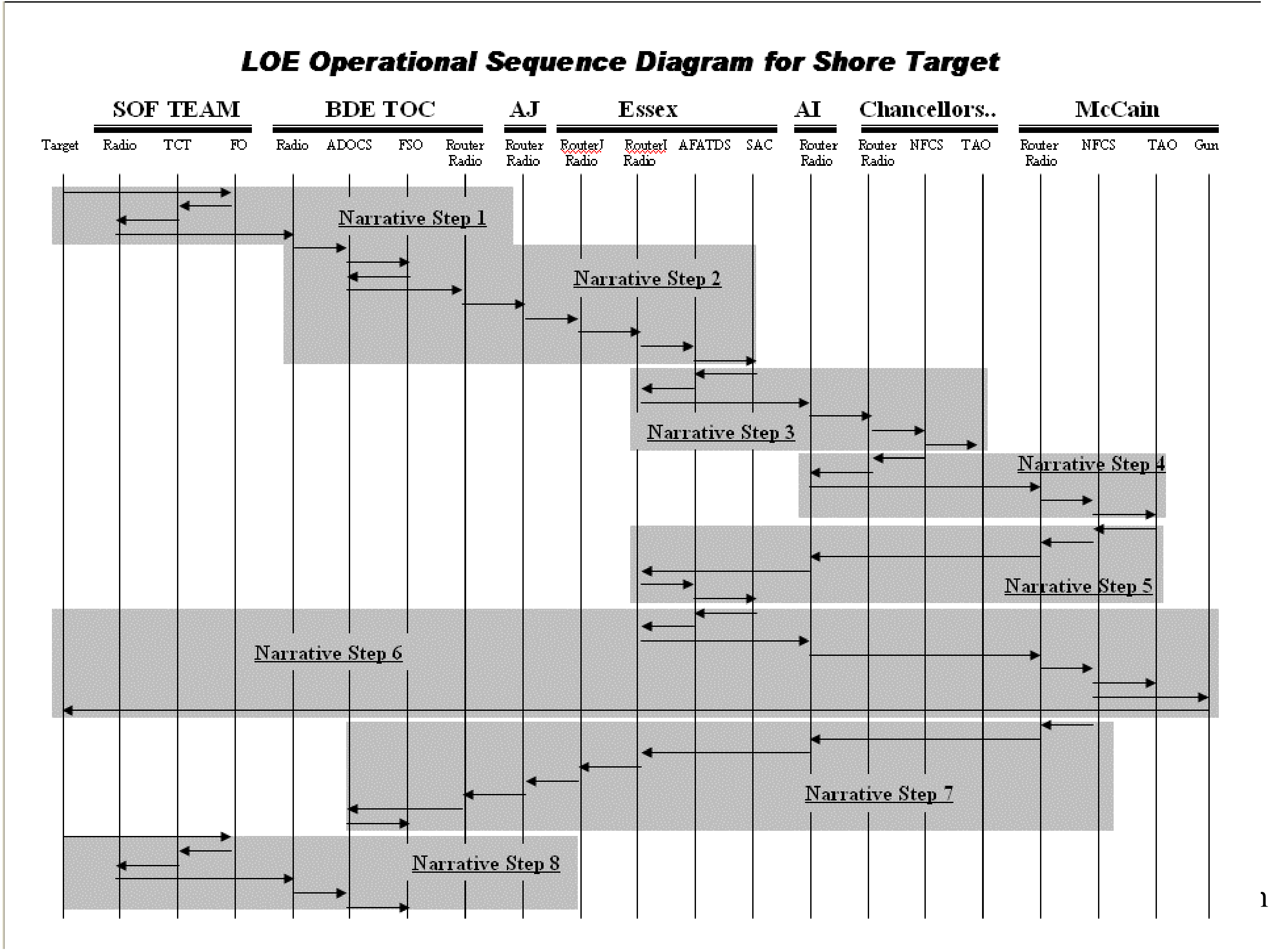
- The Limited Objective Experiment (LOE) for the network project "VSoS" was chosen to demonstrate the usefulness of the E2E T&E.
- VSoS is a network-centric SoS that must coordinate multiple sea, air, and land forces.
- E2E Testing and E2E Tool support provided extensive benefits:
  - Operational Scenario Specification and Analysis,
  - Economical Test Automation for Increased Coverage,
  - Test Reuse and Regression Testing,
  - SoS Simulation.

# E2E Support of VSoS Specification

- VSoS LOE requirements were constrained by the use of “standard” Natural language Descriptions and Sequence Diagrams
  - Labor intensive rework for every SoS change
  - Lack of timing constraints, concurrency issues, and condition definitions.
- The E2E Tool provided automated and reusable VSoS specifications
  - Decomposition of SoS components (Actor, Condition, Action, Event)
  - Requirements Templates
  - Test-Driven specification to identify errors before SoS test.
  - Prototype code generation to define a preliminary design
  - Detection of timing constraints and concurrency conflicts

Operational Scenario Specification and Analysis

# Example of a “Standard” LOE Sequence Diagram



# Decomposition of VSoS Narrative Specifications into E2E Components

- The VSoS Operational Specifications for the LOE were decomposed into:
  - 3 high-level scenarios,
  - 17 atomic scenario groups,
  - 42 scenarios,
  - 103 conditions,
  - 42 events,
  - 103 actions, and
  - 85 thin threads.

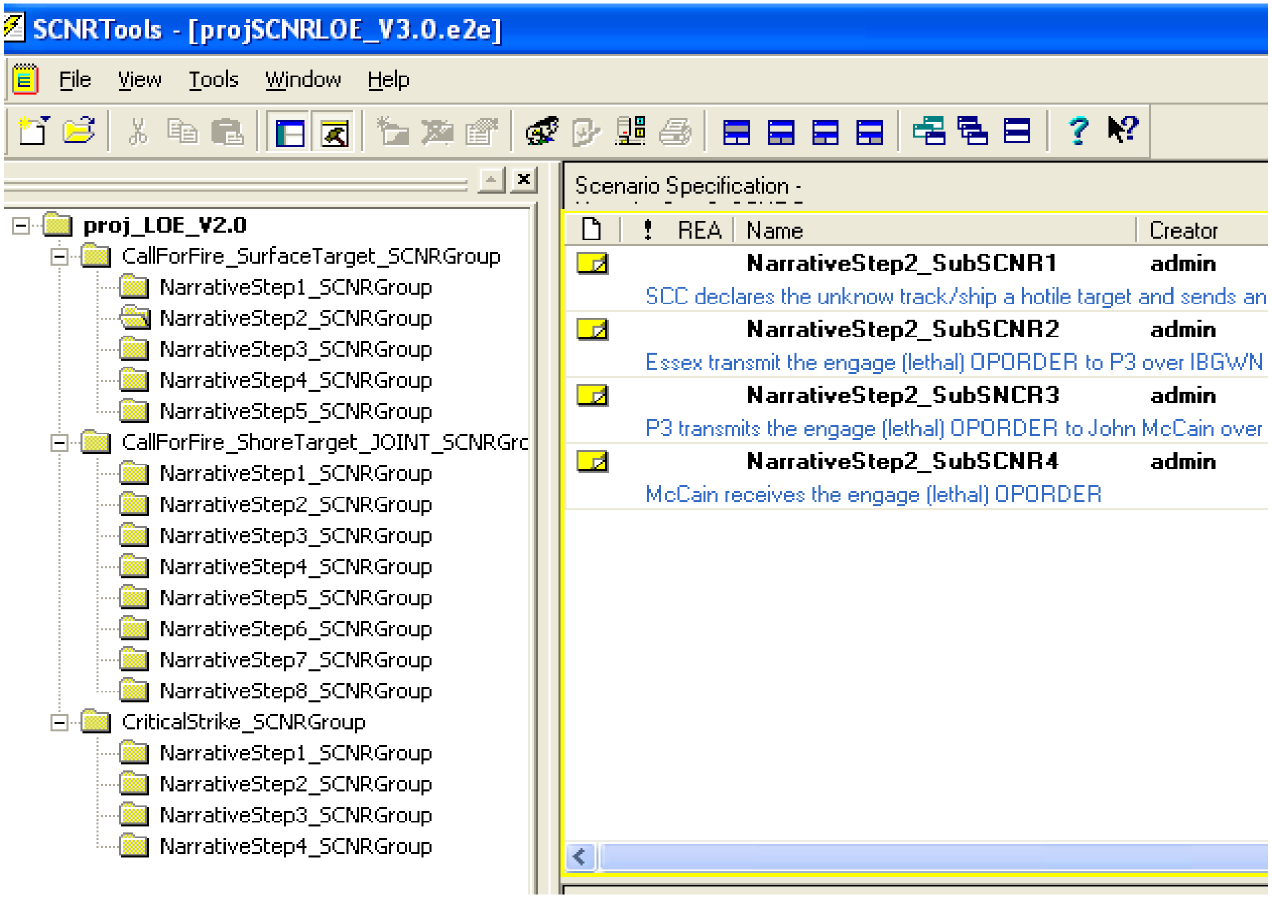
# E2E Tool Support of SoS Scenario Specification and Analysis

- Most SoS design scenarios [Object-oriented] usually cannot be used in formal End-to-End analysis due to lack of information.
- The E2E Tool scenarios provide adequate information for formal analysis:
  - Timing constraints
  - Logical/physical/functional constraints
  - Concurrency constraints
  - Synchronization/Asynchronous operations

Operational Scenario Specification and Analysis

# The E2E Tool Creates SoS Scenario Specifications for SoS Requirements Analysis

Example screen shot of  the “Call for Fire” Surface to Surface Scenario Group



- proj_LOE_V2.0
  - CallForFire_SurfaceTarget_SCNRGroup
    - NarrativeStep1_SCNRGroup
    - NarrativeStep2_SCNRGroup
    - NarrativeStep3_SCNRGroup
    - NarrativeStep4_SCNRGroup
    - NarrativeStep5_SCNRGroup
  - CallForFire_ShoreTarget_JOINT_SCNRGrc
    - NarrativeStep1_SCNRGroup
    - NarrativeStep2_SCNRGroup
    - NarrativeStep3_SCNRGroup
    - NarrativeStep4_SCNRGroup
    - NarrativeStep5_SCNRGroup
    - NarrativeStep6_SCNRGroup
    - NarrativeStep7_SCNRGroup
    - NarrativeStep8_SCNRGroup
  - CriticalStrike_SCNRGroup
    - NarrativeStep1_SCNRGroup
    - NarrativeStep2_SCNRGroup
    - NarrativeStep3_SCNRGroup
    - NarrativeStep4_SCNRGroup

Scenario Specification -

| | ! | REA | Name | Creator |
|---|---|---|---|---|
| | | | **NarrativeStep2_SubSCNR1** | **admin** |
| | | | SCC declares the unknow track/ship a hotile target and sends an | |
| | | | **NarrativeStep2_SubSCNR2** | **admin** |
| | | | Essex transmit the engage (lethal) OPORDER to P3 over IBGWN | |
| | | | **NarrativeStep2_SubSNCR3** | **admin** |
| | | | P3 transmits the engage (lethal) OPORDER to John McCain over | |
| | | | **NarrativeStep2_SubSCNR4** | **admin** |
| | | | McCain receives the engage (lethal) OPORDER | |

# Automated Thin Thread Generation

- The E2E tool can generate thin threads from system scenarios automatically.
  - A thin thread is complete end-to-end execution path
  - One scenario will generate at least one path, but often it has more thin thread
  - Each thin thread can be considered as an execution trace.

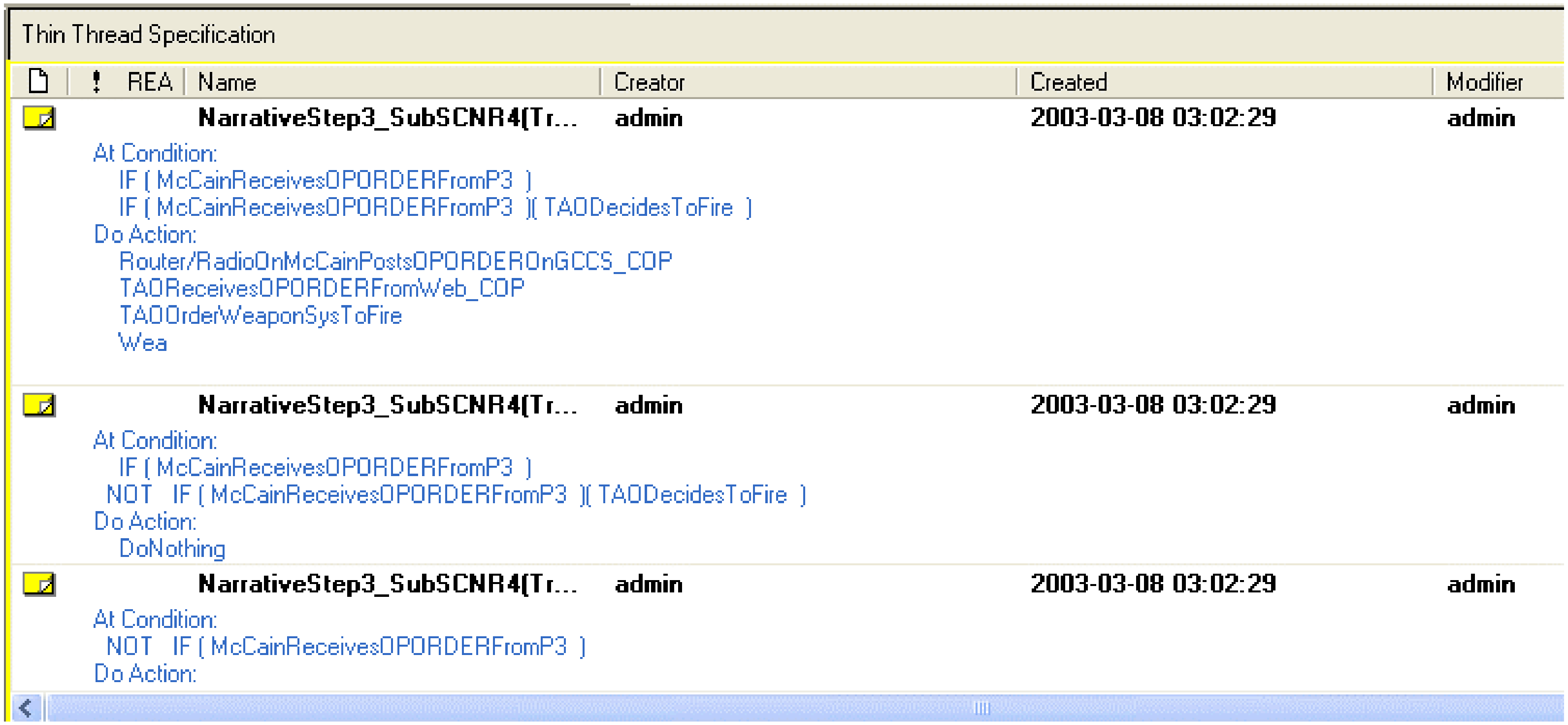
# E2E Tool Translation of Scenario Specifications to XML

```
"<ATOMIC_SCENARIO>
<IF_CLAUSE>
<IF>
<CONDITION>
<COND_ITEM>
<VALUE>189</VALUE>
<T1>-1</T1>
<T2>-1</T2>
</COND_ITEM>
</CONDITION>
</IF>
<THEN>
<ACTION>
<VALUE>195</VALUE>
<T1>-1</T1>
<T2>-1</T2>
</ACTION>
<ACTION>
<VALUE>210</VALUE>
<T1>-1</T1>
<T2>-1</T2>
</ACTION>
<IF_CLAUSE>
<IF>
<CONDITION>
<COND_ITEM>
<VALUE>211</VALUE>
<T1>-1</T1>
<T2>-1</T2>
</COND_ITEM>
</CONDITION>
</IF>
<THEN>
<ACTION>
<VALUE>212</VALUE>
<T1>-1</T1>
<T2>-1</T2>
</ACTION>
<ACTION>
<VALUE>213</VALUE>
<T1>-1</T1>
<T2>-1</T2>
</ACTION>
<ACTION>
<VALUE>214</VALUE>
<T1>-1</T1>
<T2>-1</T2>
</ACTION>
</THEN>
<ELSE>
<ACTION>
<VALUE>30</VALUE>
<T1>-1</T1>
<T2>-1</T2>
</ACTION>
</ELSE>
</IF_CLAUSE>
</THEN>
<ELSE>
<ACTION>
<VALUE>30</VALUE>
<T1>-1</T1>
<T2>-1</T2>
</ACTION>
</ELSE>
</IF_CLAUSE>
</ATOMIC_SCENARIO>
"
```

- “Call for Fires” Surface to Surface Scenario for VSoS - Sub Scenario 4 of Narrative Step 3

# SoS Specifications by Thin Threads

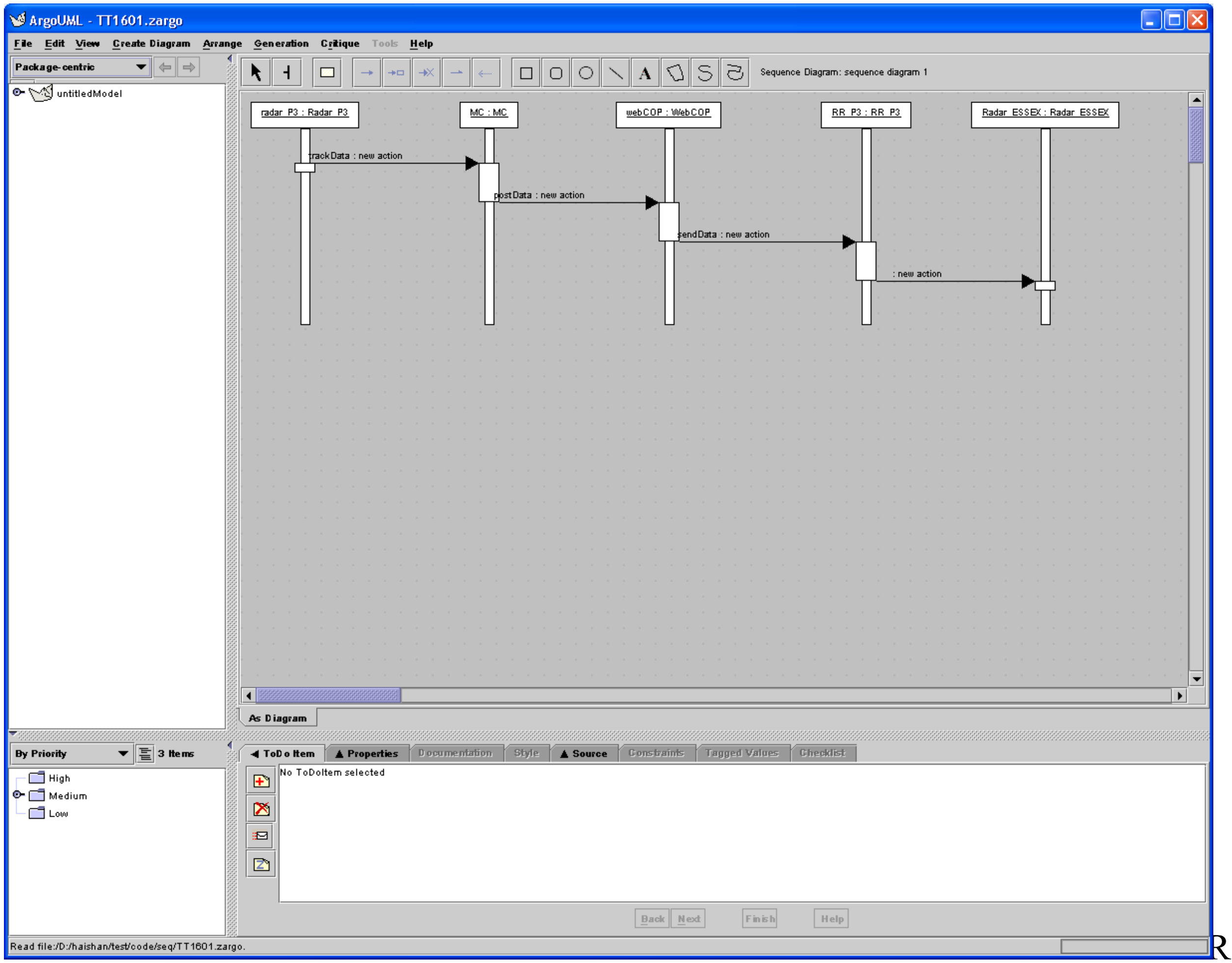
- The previous XML scenario is expanded to a thin-thread description.



Thin Thread Specification

```
 🗋 | ! | REA | Name                          | Creator | Created               | Modifier
 ▨                NarrativeStep3_SubSCNR4(Tr...  admin     2003-03-08 03:02:29     admin
          At Condition:
             IF ( McCainReceivesOPORDERFromP3  )
             IF ( McCainReceivesOPORDERFromP3  )( TAODecidesToFire  )
          Do Action:
             Router/RadioOnMcCainPostsOPORDEROnGCCS_COP
             TAOReceivesOPORDERFromWeb_COP
             TAOOrderWeaponSysToFire
             Wea
 ▨                NarrativeStep3_SubSCNR4(Tr...  admin     2003-03-08 03:02:29     admin
          At Condition:
             IF ( McCainReceivesOPORDERFromP3  )
           NOT   IF ( McCainReceivesOPORDERFromP3  )( TAODecidesToFire  )
          Do Action:
             DoNothing
 ▨                NarrativeStep3_SubSCNR4(Tr...  admin     2003-03-08 03:02:29     admin
          At Condition:
           NOT   IF ( McCainReceivesOPORDERFromP3  )
          Do Action:
```

# Sequence Diagrams can be Automated and Reused for SoS Changes



No ToDoItem selected

Read file:/D:/haishan/test/code/seq/TT1601.zargo.

# E2E Tool Automation of SoS Testing

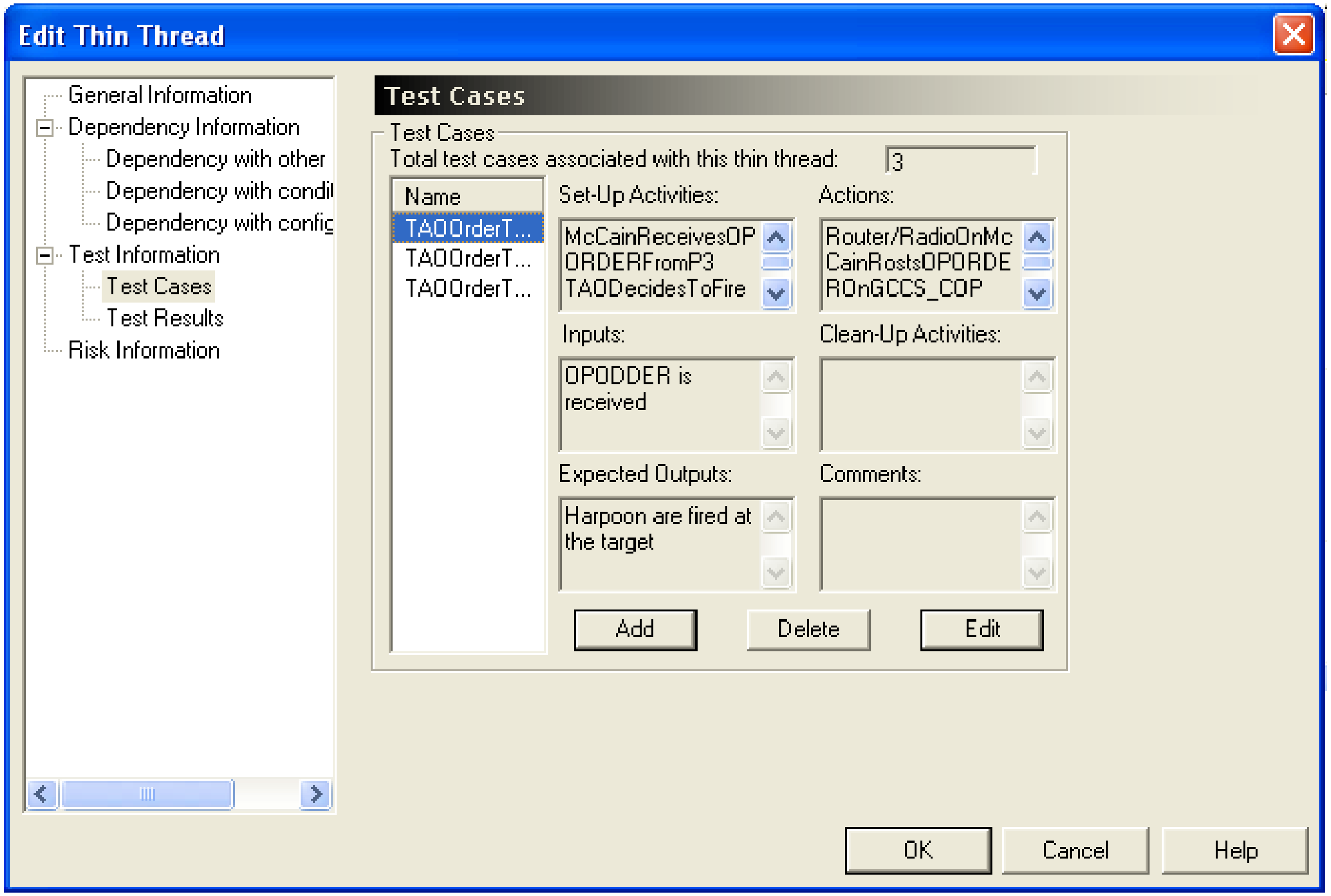
- Specification of the VSoS with Thin Threads allowed:
  - Automated Generation of Test Cases
  - Risk Analysis
  - Use of Patterns for Expanded Test Coverage
  - Test Reuse and Regression Testing

# Automated Generation of Test Cases

- Test cases can be easily derived by attaching data to the thin threads.
- Each test case consists the pre-condition, input, expected output, and post-condition information.
- Multiple test cases can be designed for each thin thread by attaching different data.
- The E2E tool thin threads can be used in various testing strategies
  - Random testing
  - Partition testing and
  - Boundary value testing

# Example Test Cases

The E2E Tool generated 3 test cases from a single thin thread.



Test Cases

Total test cases associated with this thin thread: 3

Name:
TAOOrderT...
TAOOrderT...
TAOOrderT...

Set-Up Activities:
McCainReceivesOP
ORDERFromP3
TAODecidesToFire

Actions:
Router/RadioOnMc
CainRostsOPORDE
ROnGCCS_COP

Inputs:
OPODDER is received

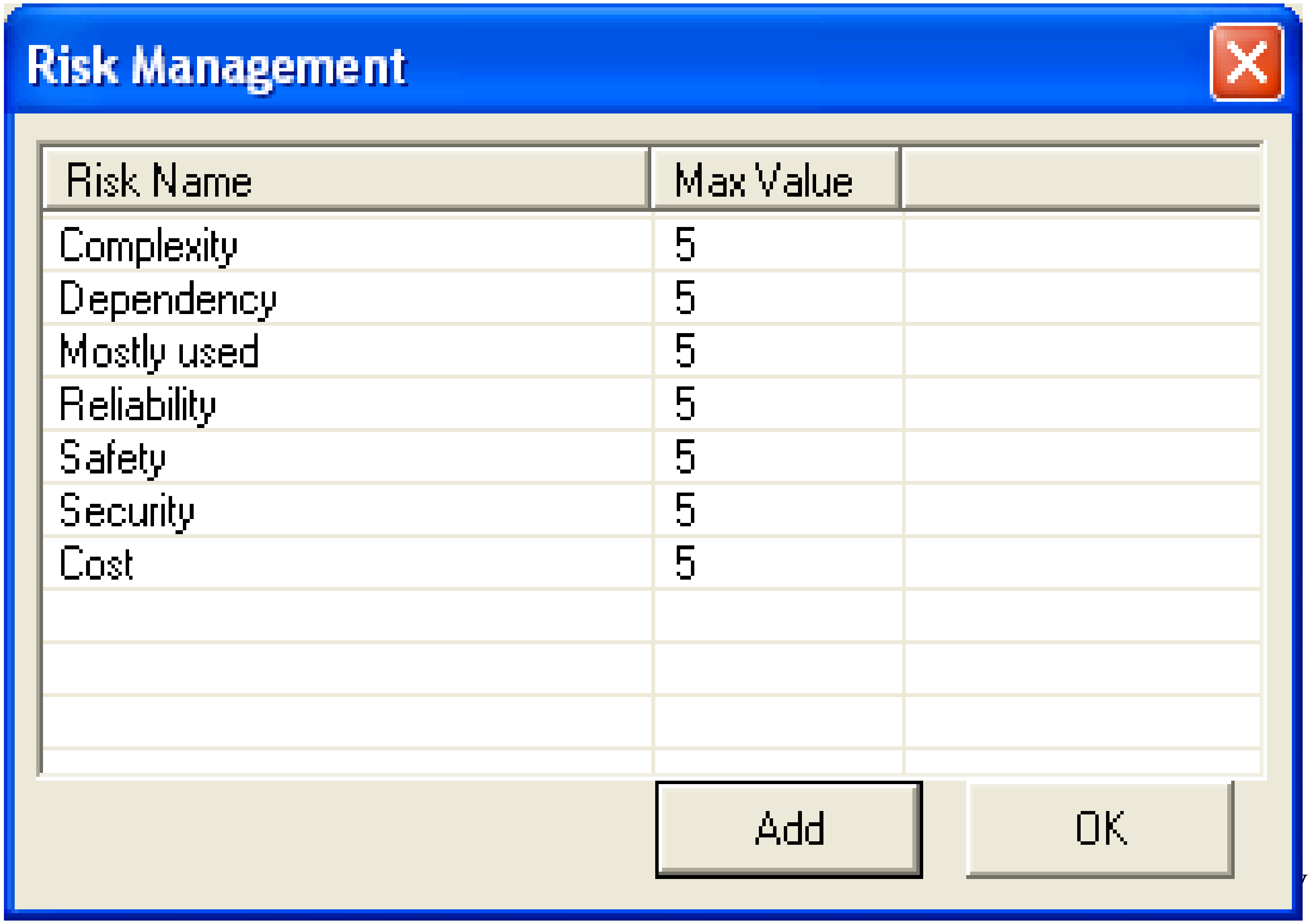
Clean-Up Activities:

Expected Outputs:
Harpoon are fired at the target

Comments:

# Seven Risk Categories were Defined for the VSoS Exercise

Each thin thread was assigned to a risk category



| Risk Name | Max Value | |
|---|---|---|
| Complexity | 5 | |
| Dependency | 5 | |
| Mostly used | 5 | |
| Reliability | 5 | |
| Safety | 5 | |
| Security | 5 | |
| Cost | 5 | |

# E2E Patterns for Test Economy

E2E patterns allow testing of multiple, recurring functions and requirements in an SoS, using a much smaller set of test cases.

- A **verification pattern** is a pre-defined verification mechanism that can be used to verify a group of behavioral requirements (temporal patterns or cause-effect relations),
- A verification pattern is normally associated with a **requirement pattern** and used to verify a group of requirements (Scenarios, Communication, Interactions, Concurrency, Timing, etc.)

# Requirement and Verification Patterns

| Pattern | Functional Test Coverage (%) |
|---|---|
| Basic requirement pattern | 40 |
| Key-event driven requirement pattern | 15 |
| Timed key-event driven requirement pattern | 5 |
| Key-event driven time-sliced requirement pattern | 7 |
| Command-response requirement pattern | 8 |
| Lookback requirement pattern | 6 |
| Mode-switch requirement pattern | 8 |
| Interleaving requirement pattern | 6 |
| **Total** | **95 %** |

# Analysis of Test Results

- The tool provides this information on test results:
  - Tested date
  - Tester
  - Test result for each test case (Pass or Fail)
  - Total number of test cases associated with this thin thread
  - Number of executed test cases
  - Number of passed test cases

# SoS Test Reuse and Regression Testing use Thin-Thread Dependency

- The E2E tool uses the scenario specifications to automatically define these dependencies between each thin thread, condition, and configuration:
  - Functional
  - Input
  - Output
  - Input/Output
  - Persistent data
  - Execution path
  - Pre-condition/post-condition
  - Control

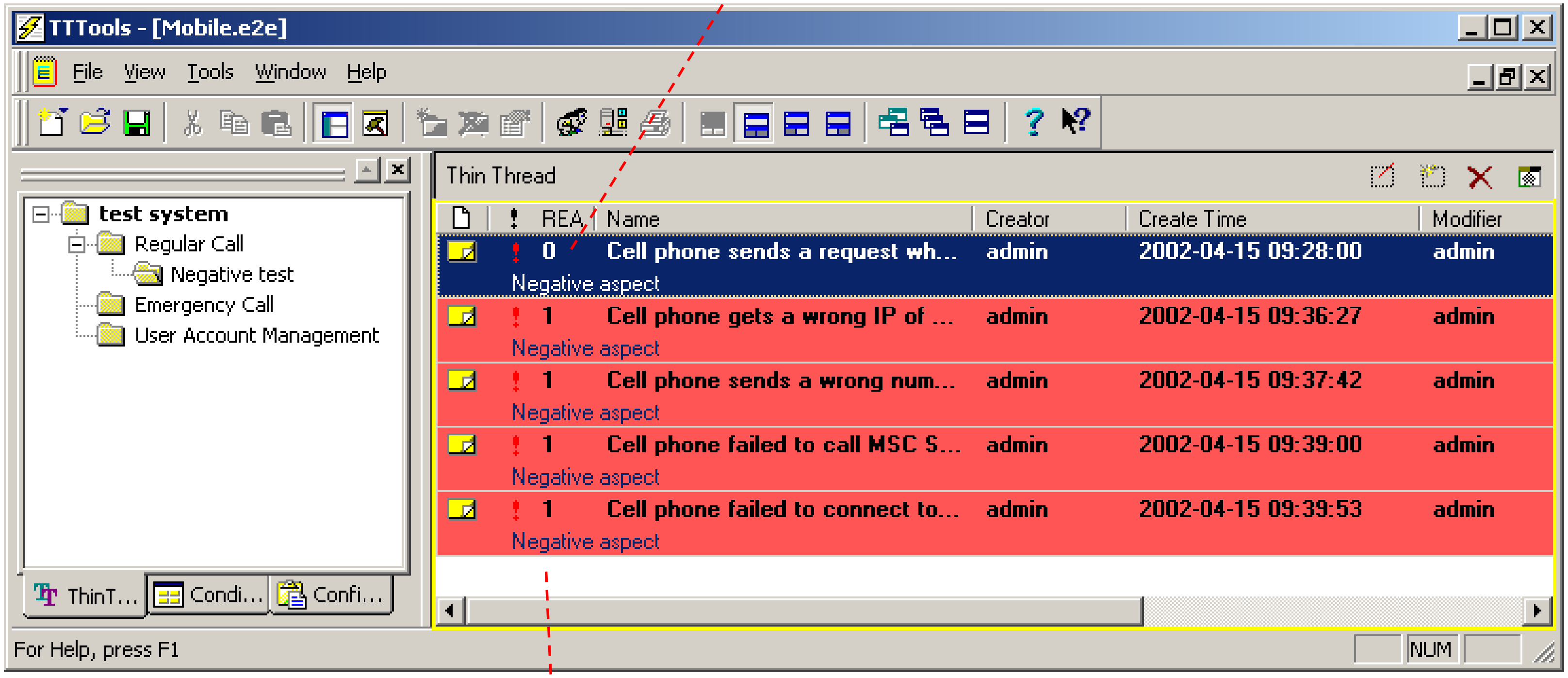
# E2E Tool Support of Regression Testing

| SoS Change | E2E Tool Dependency Identification | Support of Regression Testing |
|---|---|---|
| Thin-thread | All the thin-threads dependent on that thin-thread. | All changed thin-threads *and dependent thin-threads* are identified as candidates for regression testing. |
| Condition | All the thin-threads dependent on that condition. | |
| Configuration | All the thin-threads dependent on that configuration. | |

Test Reuse and Regression Testing

# Automated Ripple Effect Analysis

If this thin-thread is changed,



| ! | REA | Name | Creator | Create Time | Modifier |
|---|---|---|---|---|---|
| ! | 0 | Cell phone sends a request wh...<br>Negative aspect | admin | 2002-04-15 09:28:00 | admin |
| ! | 1 | Cell phone gets a wrong IP of ...<br>Negative aspect | admin | 2002-04-15 09:36:27 | admin |
| ! | 1 | Cell phone sends a wrong num...<br>Negative aspect | admin | 2002-04-15 09:37:42 | admin |
| ! | 1 | Cell phone failed to call MSC S...<br>Negative aspect | admin | 2002-04-15 09:39:00 | admin |
| ! | 1 | Cell phone failed to connect to...<br>Negative aspect | admin | 2002-04-15 09:39:53 | admin |

the four red-colored thin-threads may be affected due to dependency.

# E2E Tool Simulation Supports SoS Design

Analysis of the E2E Tool simulation results will help designers to correct conflicts and detect incomplete design functions by:

- Simulating all possible thin threads or one particular thin thread from a given state
- Verifying all the timing constraints in all possible thin threads
- Detecting all the concurrency conflicts between any thin threads

# Summary – The Challenge

- "The paradigm for the future should be based on putting a prototype system out quickly, then adapting and improving it as it is fielded." Admiral Blair
- To meet this challenge, the DoD Acquisition and T&E  process must become more rapid and adaptive.
  - Systems must be acquired and tested incrementally with a reduced cycle time of about 18 months (as recommended by CCA 1996).
  - Continuous changes in a SoS must be allowed.
  - The quality of delivered systems must be maintained by constant T&E during and after each delivery.

# The E2E Process Achieves Speed Though Automation

The E2E Tool accelerates design and test with:

- Dependency analysis to identify changed scenarios and test scripts
- E2E Tool simulation of scenarios for analysis of SoS behavior
- Verification patterns to reduce the number of test cases for adequate coverage
- Template support for automated design
- Automated generation of test scripts from scenarios
- Automated test execution by distributed agents

# The E2E Process Achieves Speed Though Reuse

- The E2E process allows comprehensive testing of an evolving SoS through reusability of design and test functions.
  - As requirements change, the E2E tool identifies the thin-thread dependencies to ensure quick re-design and adequate regression testing
- Many of the E2E tool components are reusable, including:
  - Scenarios and Thin-Threads
  - Conditions and configurations
  - Automated test scripts and test cases
  - Thin-Threads dependencies to derive regression test cases
  - Verification patterns for economical test coverage
  - Simulations