Automated Reliability Testing - ART

Bill Everett SPRE Inc.

wwe@SPRE-Inc.com
tel. (505) 890-7773
fax. (505) 890-2933

High Confidence Software and Systems Workshop
30 March 2001

SPRE

Outline

- **Reliability Testing of Software-based Products.**
  - **Why Reliability Testing?**
  - **Basic Concepts**
- **ART Project**
  - **Objectives/Goals/Approach**
  - **Overview of ART**
  - **Results/Summary**

What is the purpose of testing?

- Finding bugs?
- Assuring quality?

Assuring Quality!

To *assure* quality you must be able to measure it from the perspective of the customer or user.

A Quality Measure

- Customers are most concerned with how often a product will fail and what the *cost* will be when it does.

  Concern (Risk) = Failure Rate * Cost

- Reliability
  - The ability to operate failure free for a specified period of time under specified operating conditions.
  - Is directly related to Failure Rate.

Basic Concepts
Software Reliability

- Failures versus Faults
- Time
  - Execution versus Calendar
- Operational Profile
- Reliability Modeling

Failures versus Faults

- Failures
  - What the customer or user observes.
  - Define and categorize in terms of impact.
- Faults
  - In software, the underlying cause of failures.
  - A fault may manifest itself as one, many or no failures.

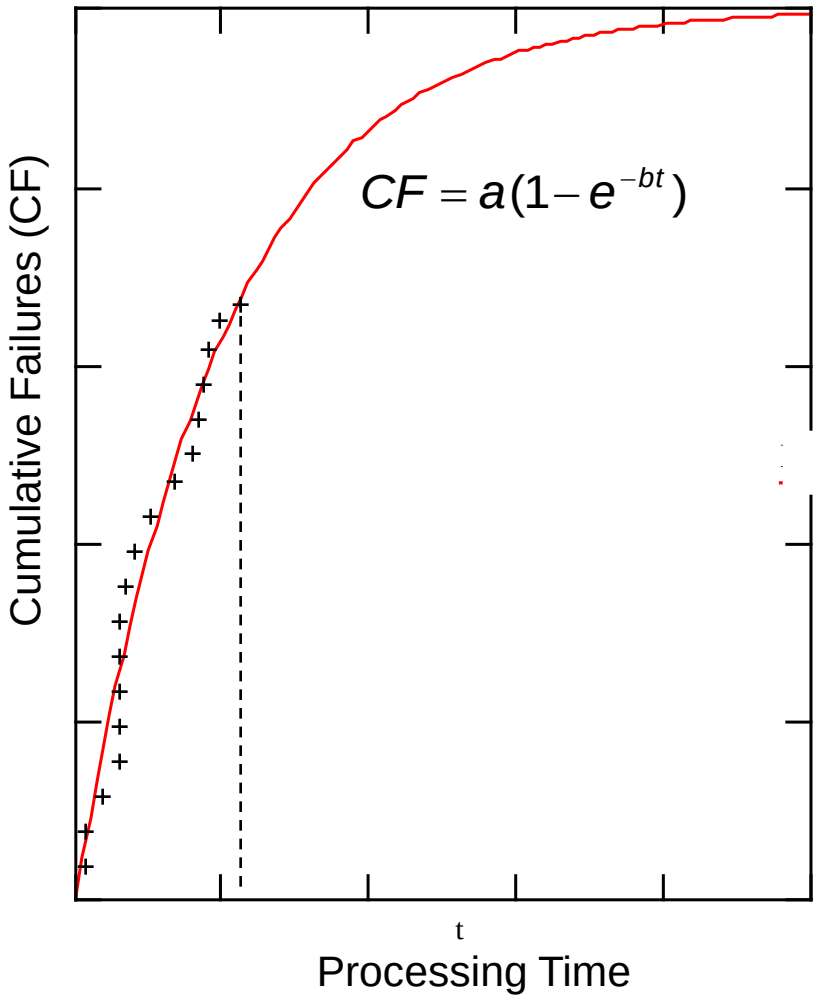
Time

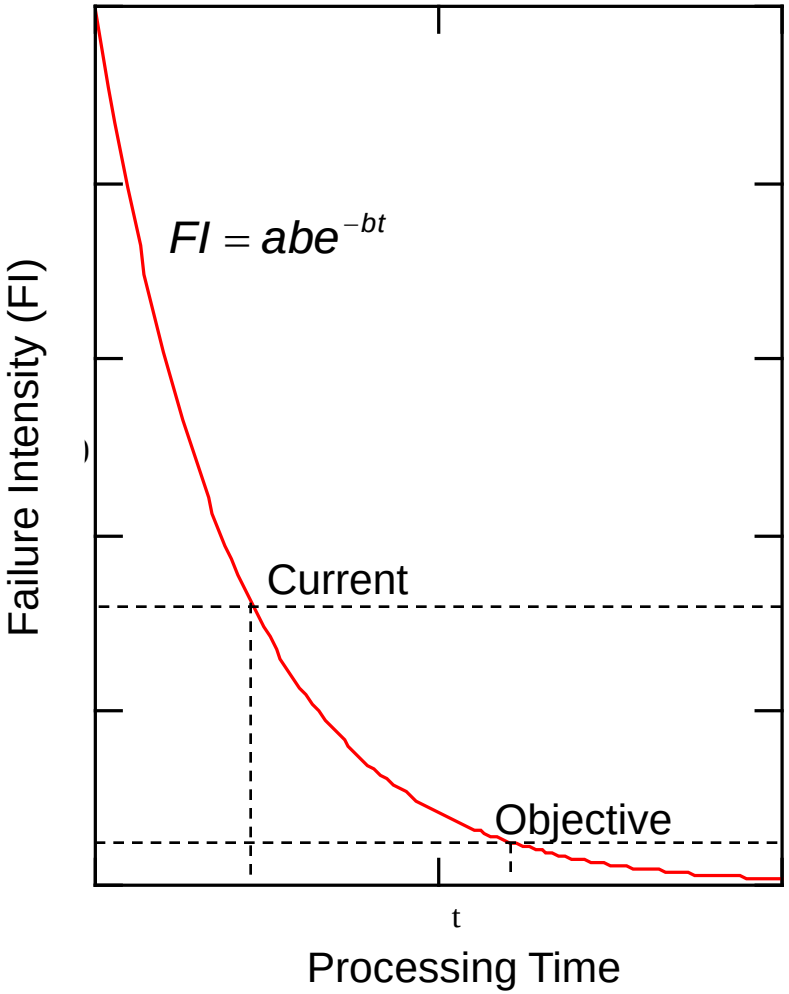
- Execution Time
  - Measure of processing time.
  - Model failure rates in terms of.
  - Use surrogate *natural units* in place of.
- Calendar Time
  - The time that customer/users experience.
  - Must translate reliability measures expressed in execution time back to calendar time.

Operational Profile

- Reliability of software is sensitive to *usage.*
- Operational Profile characterizes *usage* of the software.
  - Features/functions used.
  - Frequency of usage.

Application of SRGM's





Reliability Models

- Software Reliability Growth Models (SRGM)
  - Used during testing.
  - Account for removal of faults.
- Rich source of research and tools.
  - Limit to *descriptive* models.
  - Public domain modeling tools are available.
- Constant Reliability Models
  - Used when there is no fault removal.
  - Used to certify the reliability of existing products.

Reliability and Testing

How can we use reliability measures?

- to guide and manage testing
- to answer the question "*When are we done testing?*"

Outline

- **Software Reliability Engineering**
  - **Basic Concepts**
- **ART Project**
  - **Objectives/Goals/Approach**
  - **Overview of ART**
  - **Results/Summary**

Project Objectives

- Automate the end-to-end support for reliability testing, including
  - Developing Operational Profiles of User Activity
  - Generating Test Cases from the Operational Profile
  - Scheduling, dispatching, archiving of test runs
  - Processing test run output to detect failures
  - Performing reliability analysis from failure data
- Demonstrate applicability to reliability testing of NSA applications
- Make extensive use of COTS and public domain products.

Phase 1 (Prototype) Goals

- ✓Develop a prototype of ART
  - Serves as a “proof of concept”.
- ✓Demonstrate ART with pilot application
  - Pilot is representative of NSA applications.
- ✓Deliver an ART User Manual
  - A guide for using ART commands.
  - A handbook for how to do reliability testing.
- ✓Use COTS and public domain software
  - For Operational Profile modeling: Teradyne’s TestMaster®
  - For reliability analysis: CASRE, SRE Toolkit, Spreadsheet Programs.

Approach

- Select Pilot Application
- Develop ART Prototype
- Demonstrate the Use of ART
  - on ART itself
  - on Pilot Application
- Conduct “User Acceptance Test”

Select Pilot Application

- Criteria for selection
  - Unclassified application
  - Similar to classified applications
  - Developed by NSA staff
- Selected C-Code Analyzer *canz* as pilot
- Reviewed requirements for *canz*
  - Augmented requirements with reliability requirements
    - Set reliability objectives
      - E.g., 1 severe failure per 1,000 runs
    - Specified an operational profile
      - A set of operations performed by a user
      - Their frequency of occurrence.
  - Added interpretations for ambiguous functional requirements

Develop ART Prototype

- Developed by Teradyne and SPRE, Inc.
- Used a documented development process
  - Followed SEI CMM level 3 practices
  - Adopted (where feasible) NSA InfoSec Standards
- Used a distributed environment
  - Sun Solaris and Windows NT

Observations

- Using a formal process kept us on schedule.
- “Practice what you preach!”, use ART on itself.

Demonstrate the Use of ART on ART Itself

- Established reliability requirements for ART
  - Set reliability objectives
    - E.g., 1 failure per 1,000 commands invoked
  - Specified an operational profile
    - E.g., specific ART commands & frequency invoked.
- Provided early “alpha” testing of ART
- Demonstrated ART met its reliability requirements

Observations

- “Using ART on itself” established credibility.
- Helped debug the process of using ART.

Demonstrate the Use of ART on Pilot Application

- Baselined reliability requirements for *canz*
  - Defined failures and severity classes
  - Set reliability objectives
    - E.g., 1 severe failure per 2,000 runs.
  - Documented an operational profile
- Conducted Reliability Testing
  - Reported problems back to developers to verify, identified faults not fixed.

Observations

  - Uncovered ambiguous requirements.
  - Generated (automatically) comprehensive, data-intensive test suites (500,000 lines C-code in 9 hours).

Conduct “User Acceptance Test”

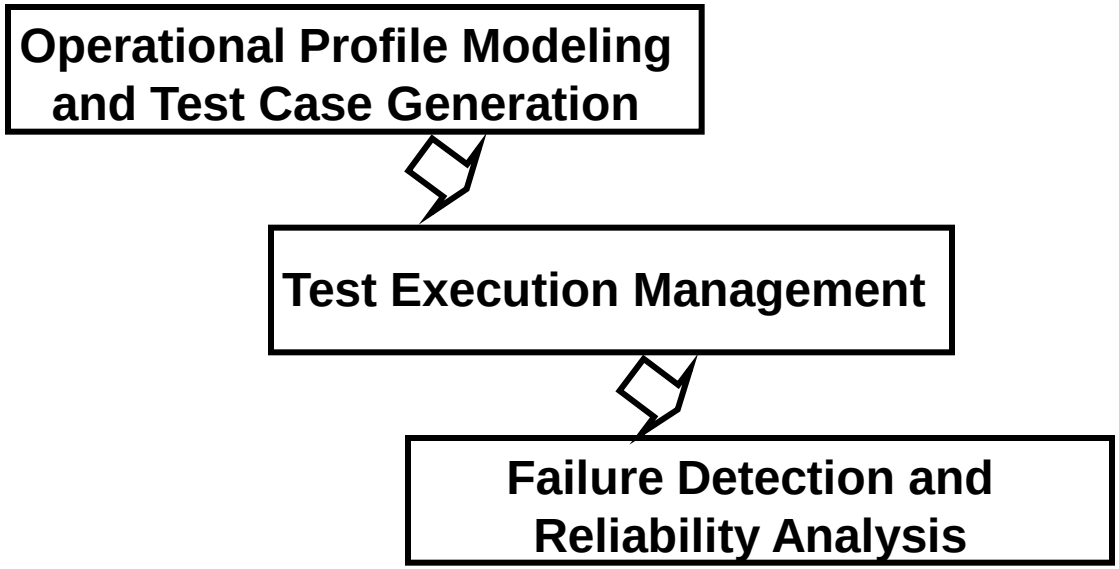
- Demonstrated flexibility of ART with
  1. a different Application (PACS)
     - Personal Access Control System developed under IDEA R&D project.
  2. a different COTS test execution toolset.
- Provided independent testing of ART
  - Testing conducted by University of Maryland.

Observations

- Provided another demonstrated use of ART
- Tested ART installation and user interfaces
- Provided useful input to ART User Guide

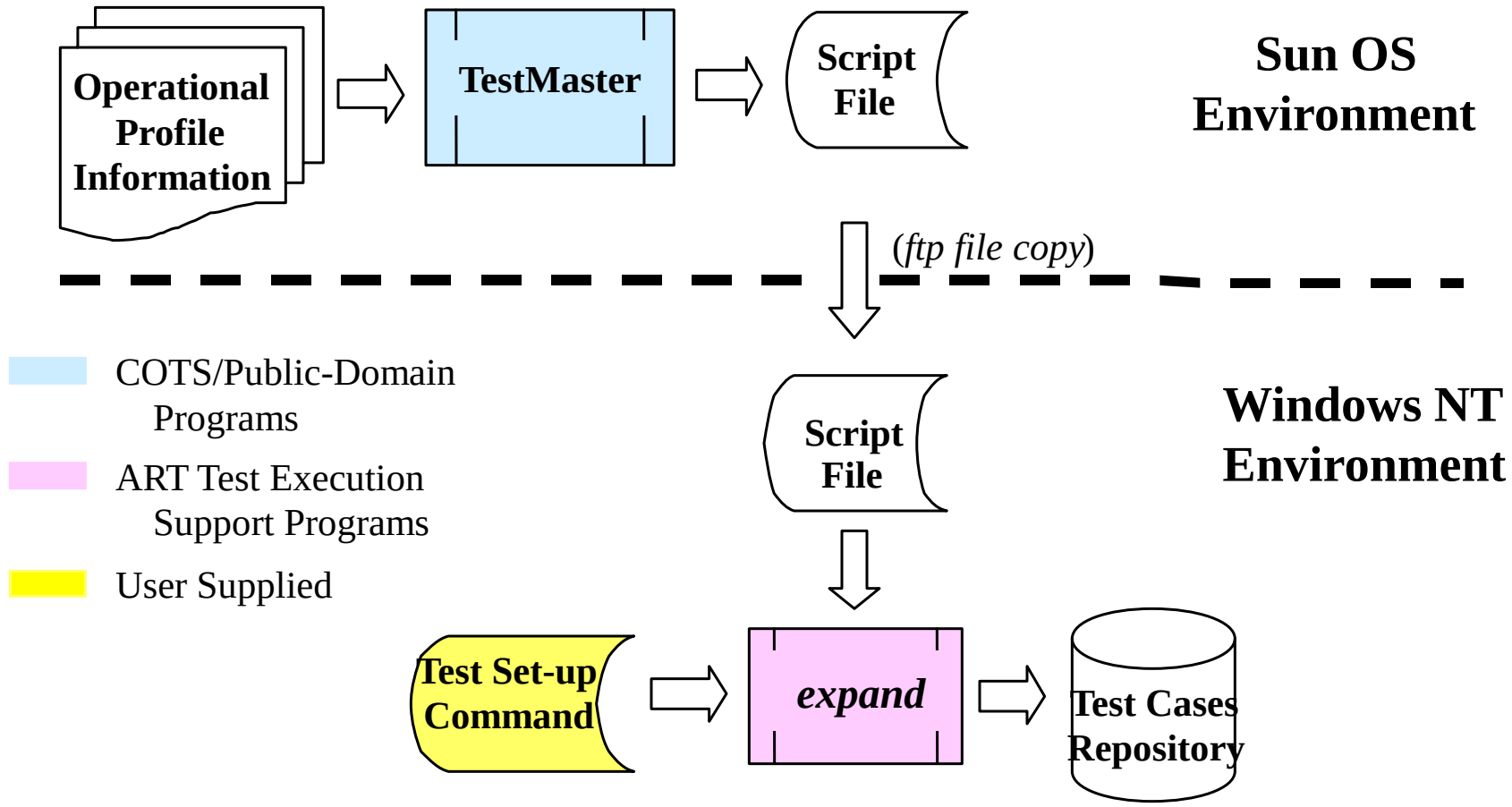
Overview of ART

- ART Architecture

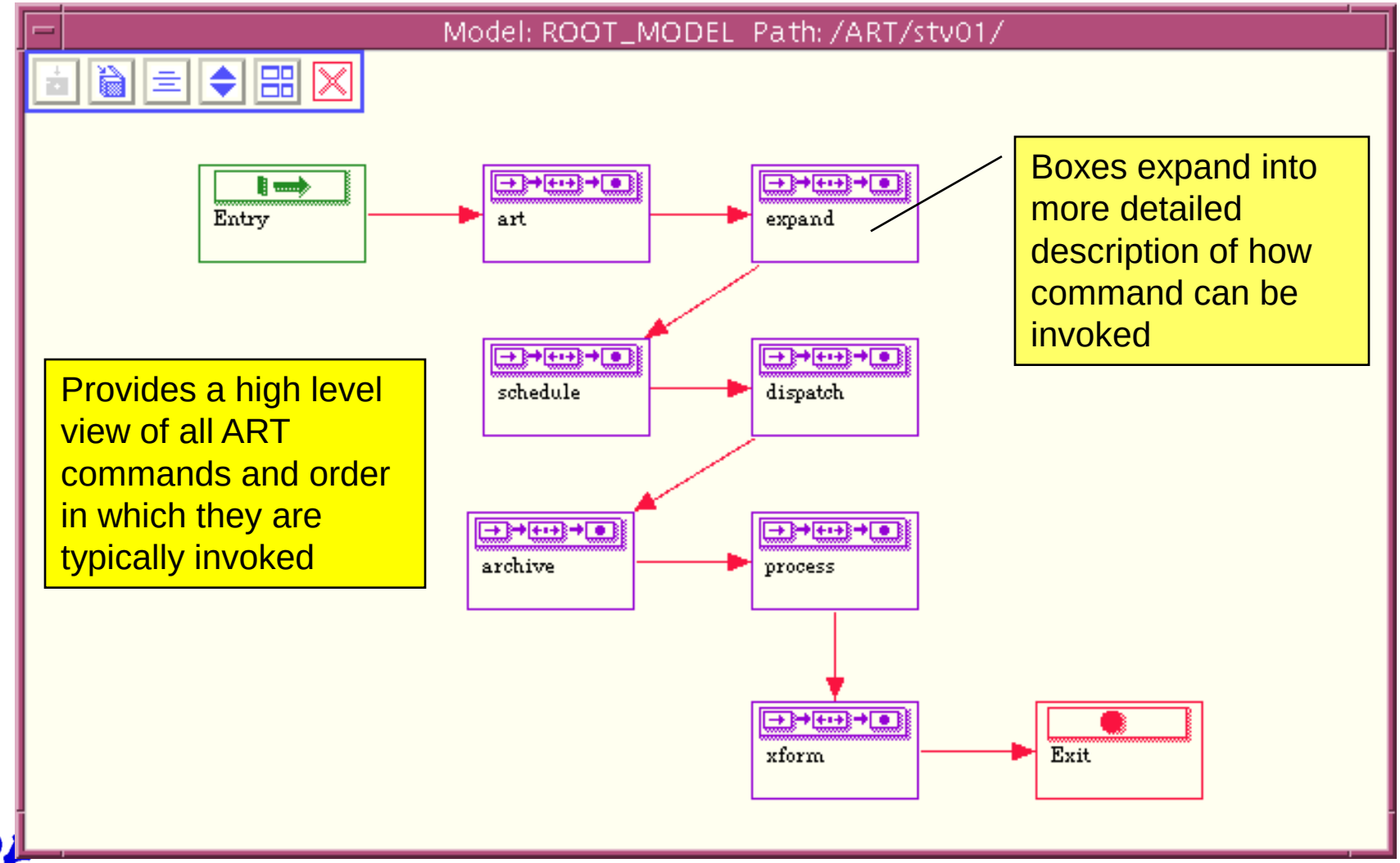


- Examples of using ART
  - An Operational Profile of ART
  - A Reliability Analysis of ART

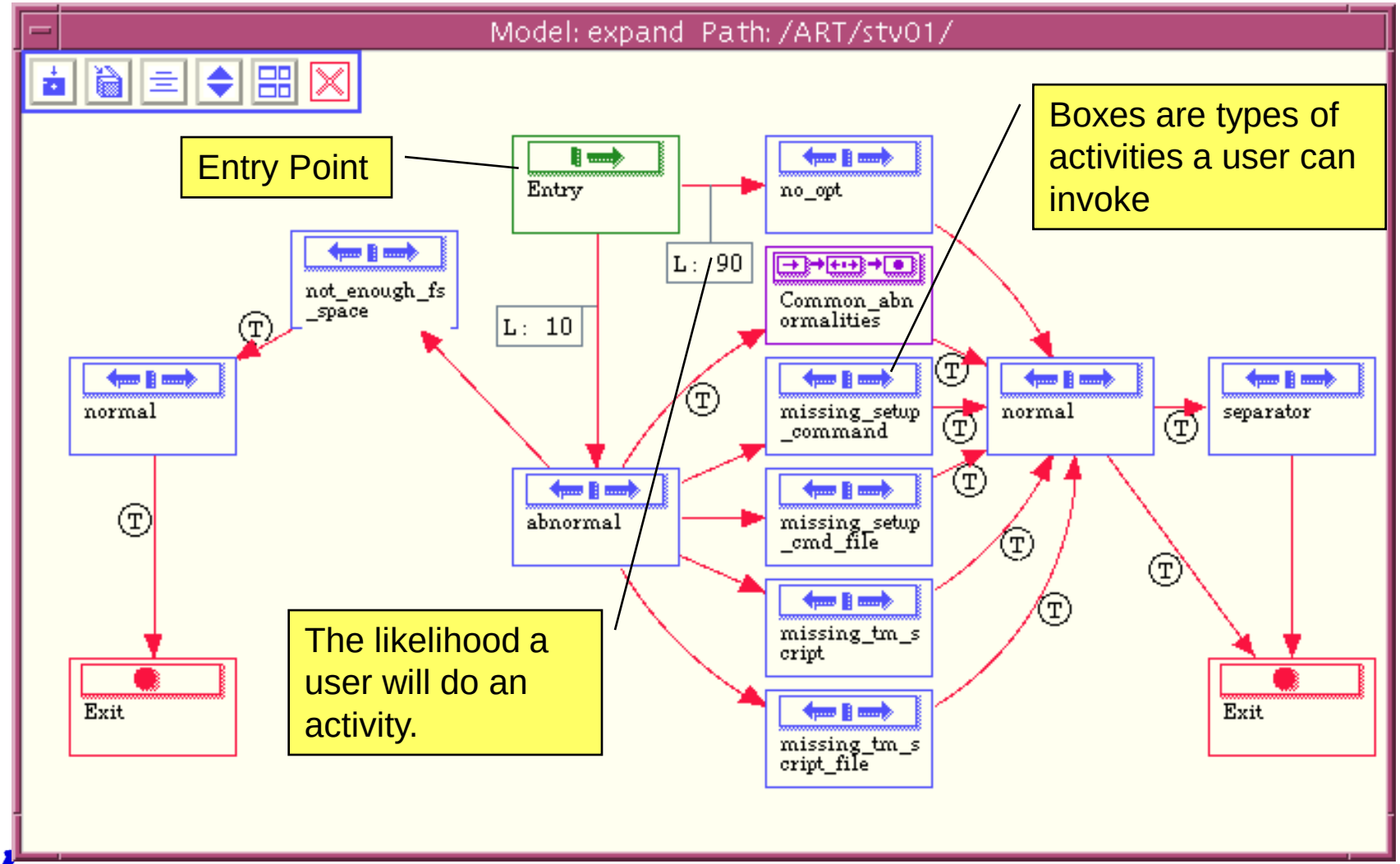
Operational Profile Modeling and Test Case Generation



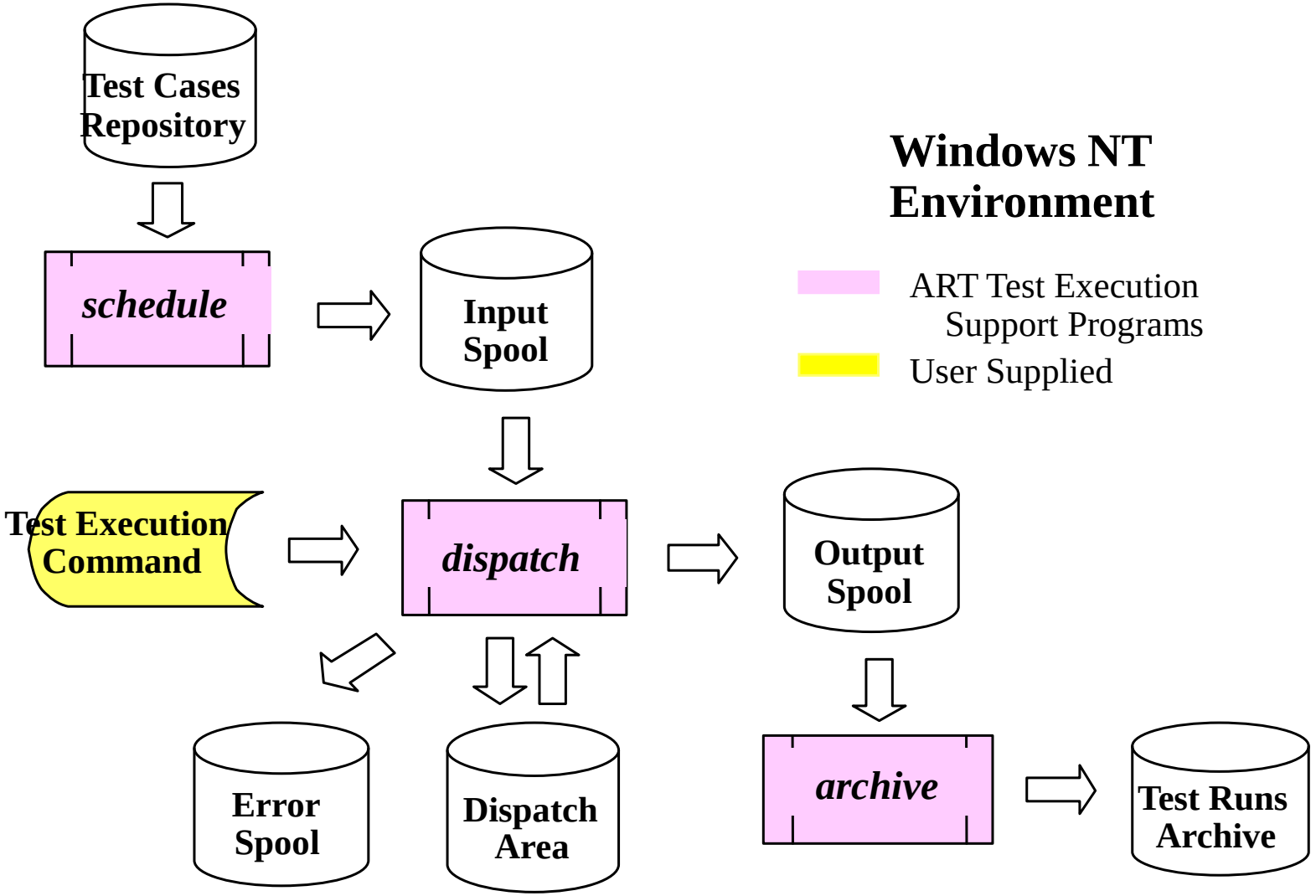
An Operational Profile Model of ART
An ART Session



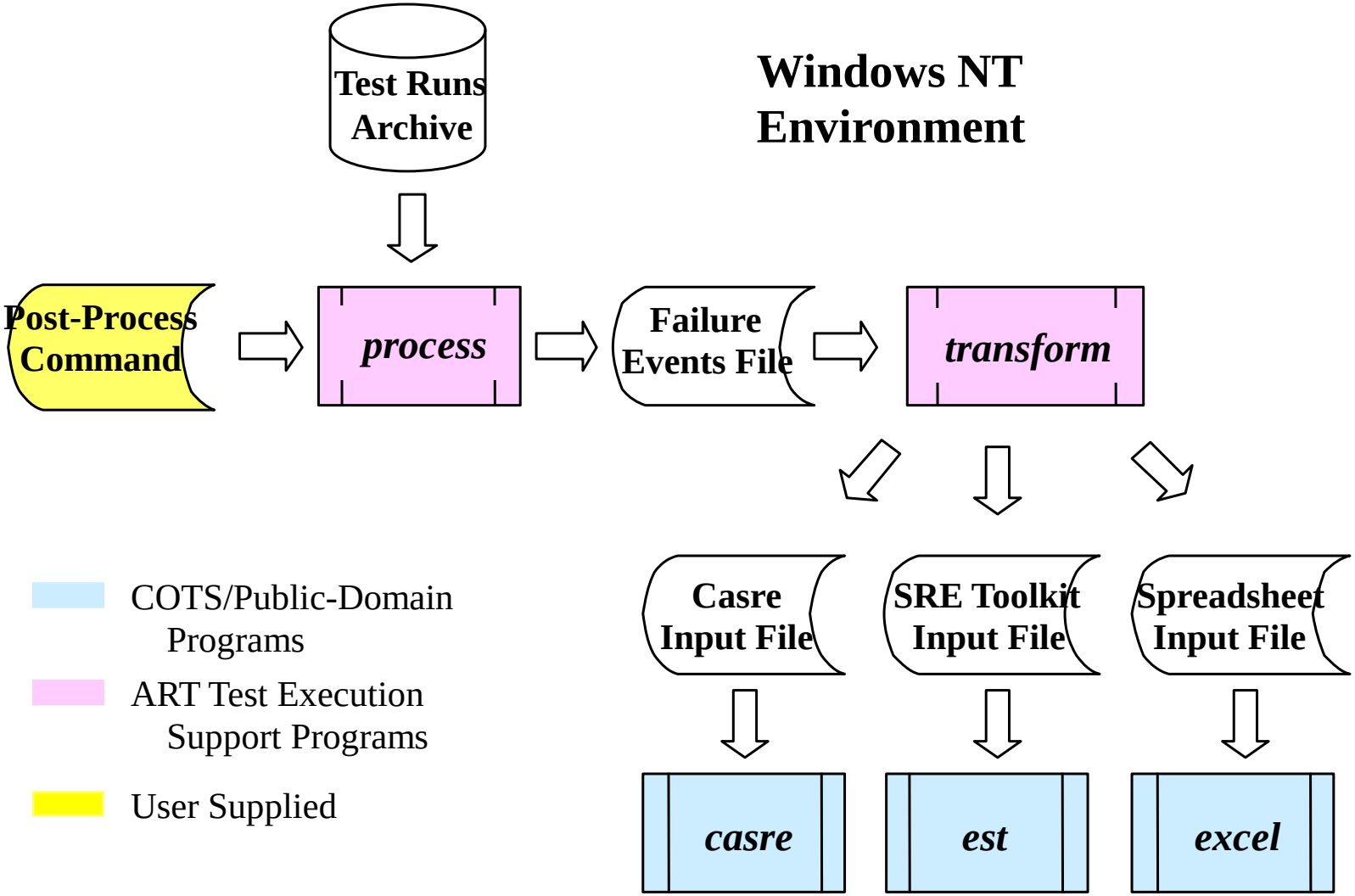
An Operational Profile Model of ART
Different ways of invoking expand command



Test Execution Management



Failure Detection and Reliability Analysis



A Reliability Analysis of ART

Reliability Objective

```
                    SOFTWARE RELIABILITY ESTIMATION
                      EXPONENTIAL (BASIC) MODEL
            ART SYSTEM TEST - ANALYSIS OF COMPOSITE FAILURE DATA
BASED ON SAMPLE OF                  15 TEST FAILURES
TEST EXECUTION TIME IS              7104 COMMANDS
FAILURE INTENSITY OBJECTIVE IS      1 FAILURES/1000-COMMANDS
CURRENT DATE IN TEST                991001
TIME FROM START OF TEST IS          1 DAYS
                         CONF. LIMITS          MOST          CONF. LIMITS
                  95%    90%    75%    50%  LIKELY   50%    75%    90%    95%
                ------ ------ ------ ------ ------ ------ ------ ------ ------
TOTAL FAILURES      15     15     15     15     15     15     15     16     16
     *********** FAILURE INTENSITIES (FAILURES/1000-COMMANDS) ************
INITIAL           5.60   6.45   7.84   9.24  11.28  13.36  14.83  16.37  17.36
PRESENT         0.0047 0.0070 0.0133 0.0242 0.0554  0.123  0.211  0.361  0.500
```

Most likely value of failure intensity (rate) is 0.055 failures/Kcmd.

95% confidence that failure intensity is 0.5 failures/Kcmds

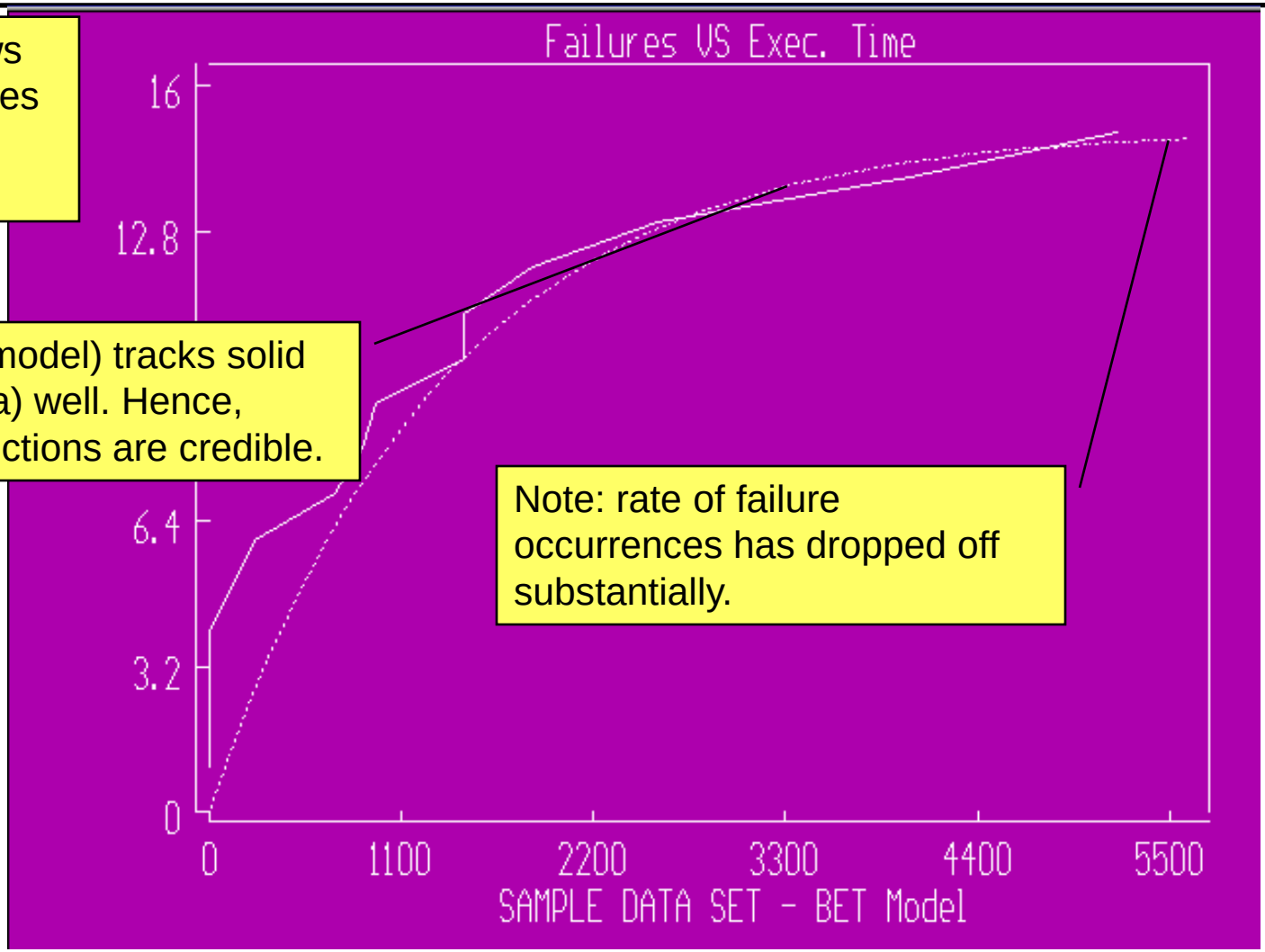
Both fall below our reliability objective.
Can stop testing!!

A Reliability Analysis of ART

This graph shows cumulative failures per command executed.

Dotted curve (fitted model) tracks solid curve (observed data) well. Hence, failure intensity predictions are credible.

Note: rate of failure occurrences has dropped off substantially.



Results

- Demonstrate the Use of ART
  - on ART itself
  - on Pilot Application
- Conduct “User Acceptance” Test

ART met its Reliability Objectives!

- Reliability Objective
  - 1 failure per 1,000 commands invoked.
- Results of Reliability Analysis
  - 0.055 failures per 1,000 commands invoked

Demonstrate the Use of ART on ART Itself

- Summary
  - Problem Reports
    - Uncovered 34 faults (19 minor, 15 major).
    - Identified 12 enhancements.
  - Operational Profile Characteristics
    - Modeled over 150 invocations of ART commands.
      - 80% normal invocations, 20% abnormal
  - Test Cases
    - Each test case represents an ART session
    - 6.5 commands invoked per test case
    - Checked 780 conditions (about 5 per command invoked)
  - Test runs
    - Processed 1,200 test cases over 3 days
    - Invoked over 8,000 commands.
    - Checked over 40,000 conditions.

Demonstrate the Use of ART on Pilot Application

- Reliability Objectives

Severity	Description	Reliability Objective
Severe	Program hangs	< 1 failure / 2,000 runs
Major	Error in computed metric > 10%	< 1 failure / 200 runs
Minor	Error in computed metric < 10%	< 1 failure / 20 runs

- Reliability Analysis
  - No severe failures detected
  - Application as tested did not meet documented reliability requirements for major or minor failures.
  - Most failures were computational errors.

Demonstrate the Use of ART on Pilot Application

- Reliability Analysis (cont'd)
  - Failure types and their rates of occurrence

Major Failures
Objective: 1 per 200 runs

Computed Metric	Observed Failures per 200 Runs
# Exits	199*
PCM Count	110
# Comment Lines	30
# Statements	18
# Total Lines	0.4

Minor Failures
Objective: 1 per 20 runs

Computed Metric	Observed Failures per 20 Runs
# Comment Lines	17
# Statements	16
# Total Lines	15
PCM Count	8
# Exits	0.4

  - Root causes of faults triggering failures
    - Interpretation of ambiguous requirements
    - * Implementation changes not reflected back in requirements

Demonstrate the Use of ART on Pilot Application

- Summary
  - Generated 2,000 test cases.
    - Over 500,000 lines of C-code generated in 9 hours.
  - Made 998 test runs over 2 days.
    - Number of runs deemed sufficient to do reliability analysis.
  - Uncovered 26 problems with *canz* when developing or running test cases.
  - Excluded rare situations associated with 12 problems during reliability testing.
    - 4 caused severe failures, 8 caused major failures

Conduct “User Acceptance” Test

- ART passed “user acceptance test”.
  - 5 problems reported during initial testing, all problems resolved
  - After problem resolution, over 2,000 test cases run without problems
  - Users Guide updated to include issues identified during User Acceptance

Summary

- Demonstrated feasibility of Automated Reliability Testing using the ART prototype system
- Modeling user interactions in ART allowed
  - Early detection of defects in specifications
  - Focused testing for debugging an application
  - Determination of a system's reliability
- Large potential value for real agency applications

Next Steps

- Trial ART with In-house Application
- Develop a Pre-production version of ART