Raytheon

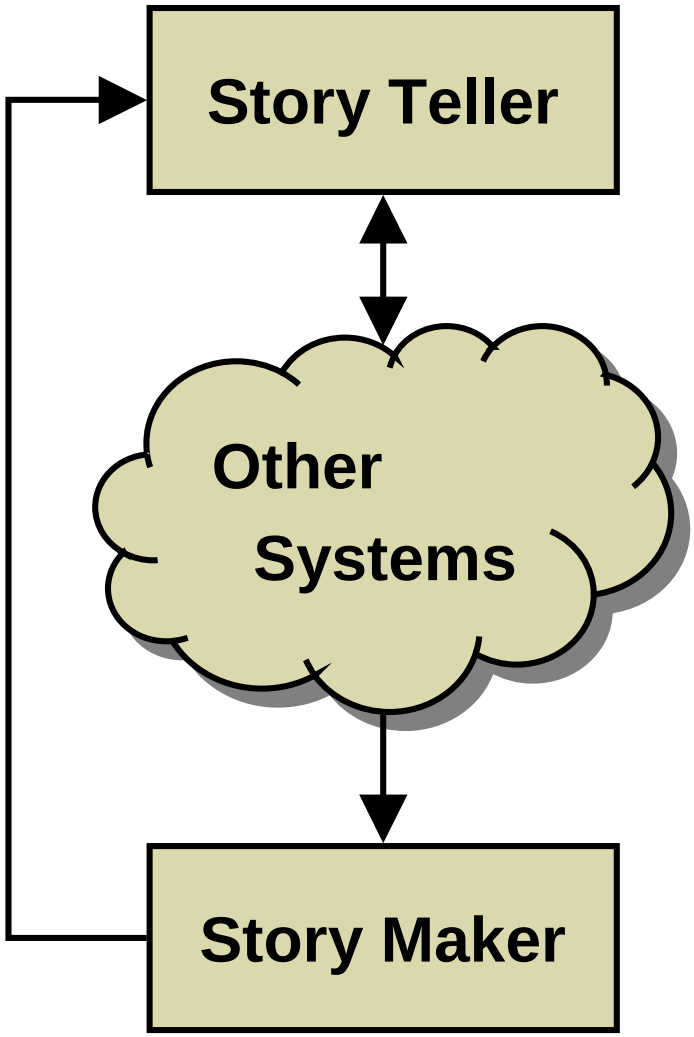
Customer Success Is Our Mission

Agent SMITH:
Evolution of a Test Tool in Tcl/Tk

John J. Seal
Principal Software Engineer
23 October 2008

Background

- Customer has an existing system called Story Teller
- It interacts with other systems
- They asked us to add a new system called Story Maker
  - Gets data from other systems
  - Sends data to Story Teller
  - Well-defined interface (ICD)



The Problem

- Story Maker not available during development!

- Extremely complicated accept/reject logic
  - Messages contain many interrelated fields
  - Teller performs validity and consistency checks
  - Literally hundreds of test cases
  - Certification required to play with those Other Systems

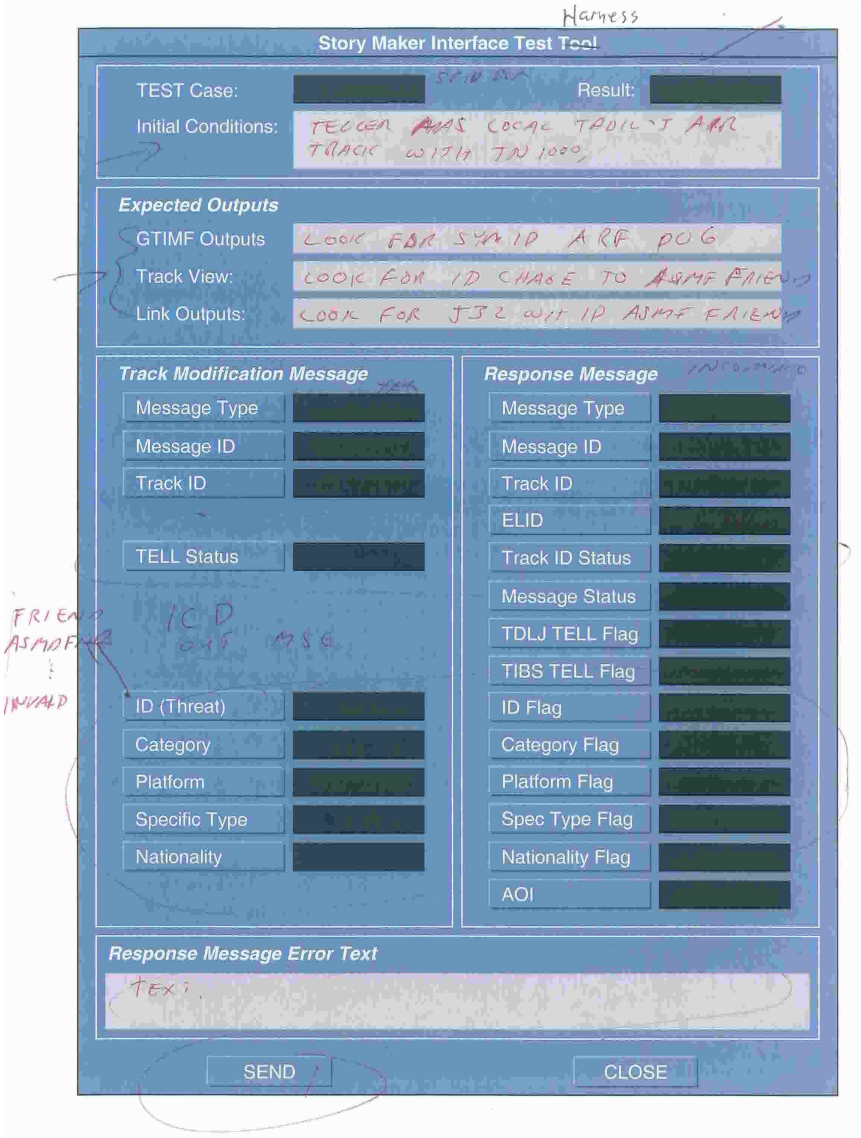
- Several communities with different needs
  - Software Engineers need to perform Unit Testing
  - System Engineers need to perform Integration Testing
  - Test Engineers need to perform Functional Qualification Testing

Risky effort – How can we ensure success?

Proposed Solution

- Story Teller project engineer sketched out an idea for a tool that would let you:
  - Build a test message
  - Describe the required initial conditions and expected outputs
  - Send it to Teller
  - See the response
  - See any error messages
  - Indicate test PASS/FAIL

- How hard would it be?



Tcl/Tk to the Rescue!

- Excellent prototyping environment
  - Tcl makes networking easy
  - Tk excels at building GUIs
- Already used in Story series and related systems
- Development proceeded in four stages
  - 1. Prototype proposed solution for proof-of-concept demo
  - 2. Automate the tool with playback of test cases
  - 3. Hook into Teller to monitor internal and external data
  - 4. Parse the data into human-readable form

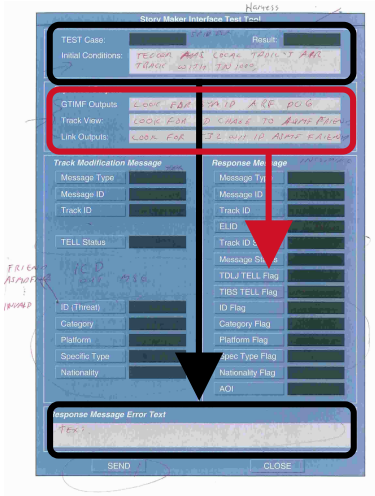
We use Tcl/Tk for prototypes *and* delivered applications

Why “Agent” SMITH?

- A previous project of mine was nicknamed “matrix”
  - It was developed when the first Matrix movie was released
  - It features a prominent matrix of data in the form of a TkTable
- Another project was a code counting tool
  - Count SLOCula
  - Started a trend of including “titles” when naming things
- The acronym SMITH was a natural for this project
  - Someone called it “Agent SMITH”, with a Hugo Weaving drawl
  - The name stuck

1. Prototype Proposed Solution

- Add detailed Expected Response fields
  - Eliminate confusing Expected Outputs fields
  - There are really two kinds of response
    - The actual response message back from Teller
    - Actions taken by Teller and the Other System as a result
  - Can automatically check expected vs. actual response
  - Still need a way to describe external reactions, but how?



- Add generic Event Log pane
  - Scrolling text widget
  - Combines TEST Case, Initial Conditions, Result, and Error Text

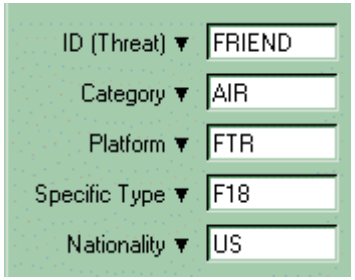
- Add destination host:port fields for message

We made some design changes *before* the demo

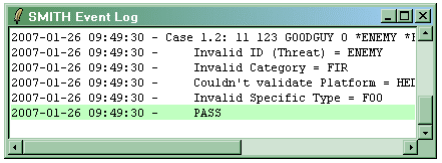
Prototype (continued)

- Demonstrate proof-of-concept to Teller team

- Add menus to aid in data entry
  - First try used tk_optionMenu widgets
    - Not good for labels - they show their current selection
    - Made the window big and cluttered
  - Switch to menubuttons with down-arrow character in name
  - Should we have used combo boxes? (Maybe still cluttered.)



- Reduce size of main window
  - Make the Event Log a separate resizable window
  - Move the Exit button into a File menu
  - Move the destination specification into a Server menu



We made some design changes *after* the demo

2. Automate the Tool

Raytheon Technical Services Company LLC

- Add entry to File menu to Open a test case file
  - Sourced by Tcl
  - Uses a simple Domain-Specific Language (DSL)

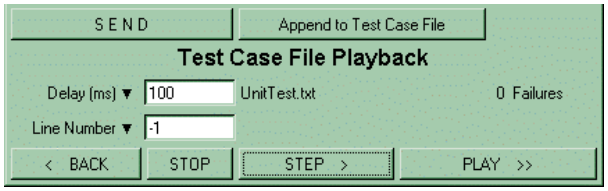
- Sample test case file

```
# Case  Msg Msg     Track  TELL ID                          Spec
#  Num  Typ ID      ID     Stat Threat      Cat     Plat    Type    Nat
case 1  11  123     JJS     0   ASMDFRND    AIR     FTR     F18     US
get msg "Enter the desired Message ID:" ;# get value of msg from operator
case 2  11  $msg    JJS     0   FRIEND      AIR     FTR    *F-18    US
    Illegal specific type
note "Special setup required for next test case!" ;# stops automatically
case 3  11  123     JJS     0   FRIEND      AIR    *BMR    *F18     US
    Inconsistent platform & specific type
```

It's easy to create Domain Specific Languages with Tcl

Automate (continued)

- Add “transport controls” for playback
  - BACK Go back to previous test case
  - STOP Stop playback
  - STEP Execute one test case
  - PLAY Begin executing test cases in sequence
  - Delay Between test cases (speed control)
  - Display of current test case, line number, and file name



- Add button to “record” new test cases
  - The testers are not programmers
  - They didn’t like preparing test case files with a text editor
  - They saw the blindingly obvious solution that I missed

Always listen to feedback from the user community

3. Hook Into Teller

- Teller uses a peculiar homegrown IPC scheme internally
  - Source code was available, but…
  - Functions are compiled into static libraries
  - What are my options?

Possible Solution	Done it?	Fear it?
CriTcl	No	Yes
Ffidl	No	Yes
Custom C extension in shared library	Yes	No

  - Build shared library with wrappers for Teller IPC functions
  - Use it to register for Teller internal messages

- Open socket to receive other messages intended for Maker

It's easy to extend Tcl with other languages

Hooks (continued)

- Extend the test case DSL
  - Specify outgoing messages we expect to be generated
  - Specify incoming messages we expect to come back

- Add Log menu
  - What to log
    - Commands — To Teller
    - Responses — From Teller
    - Outgoing — Messages to Other Systems
    - Incoming — Messages from Other Systems
  - How to log it
    - Event summary
    - Hex/ASCII dump

4. Parse the Data

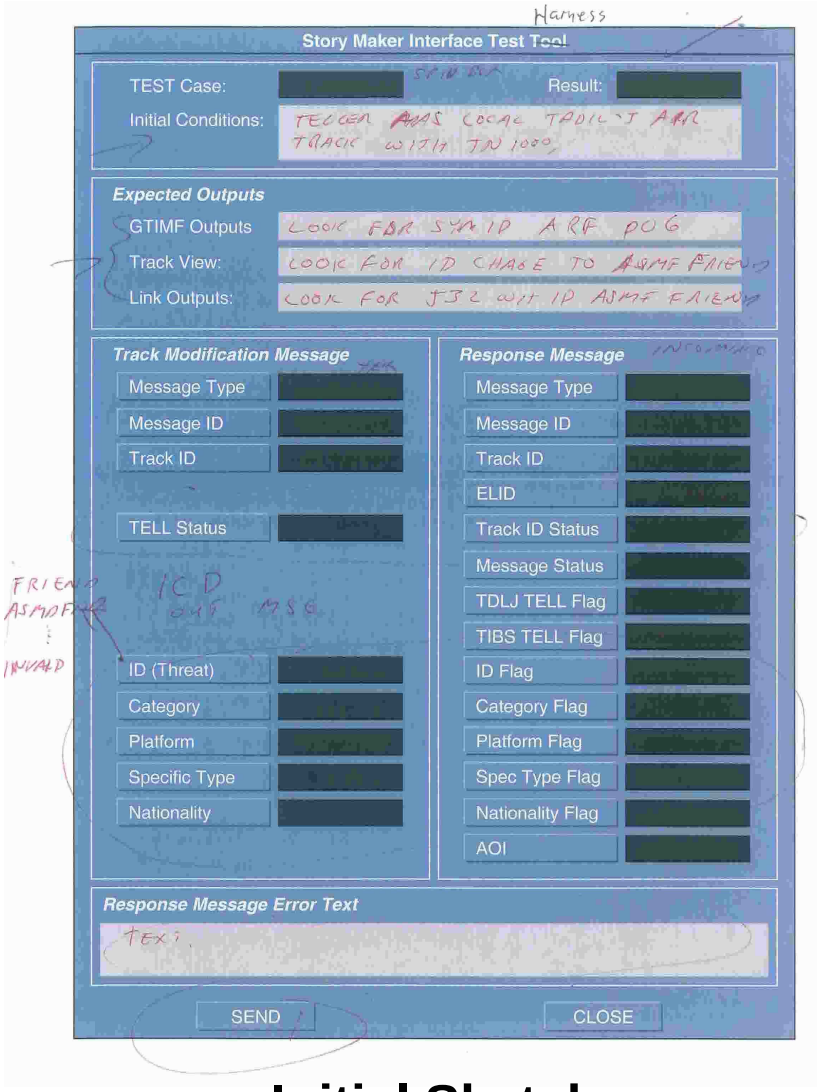
- Generic Tactical Information Message Format (GTIMF)

- Teller already has a C tool to dump GTIMF messages
  - Parses fields and displays them numerically
  - Doesn't interpret values, so not truly human-readable
  - Hard to correlate with Agent SMITH Event Log

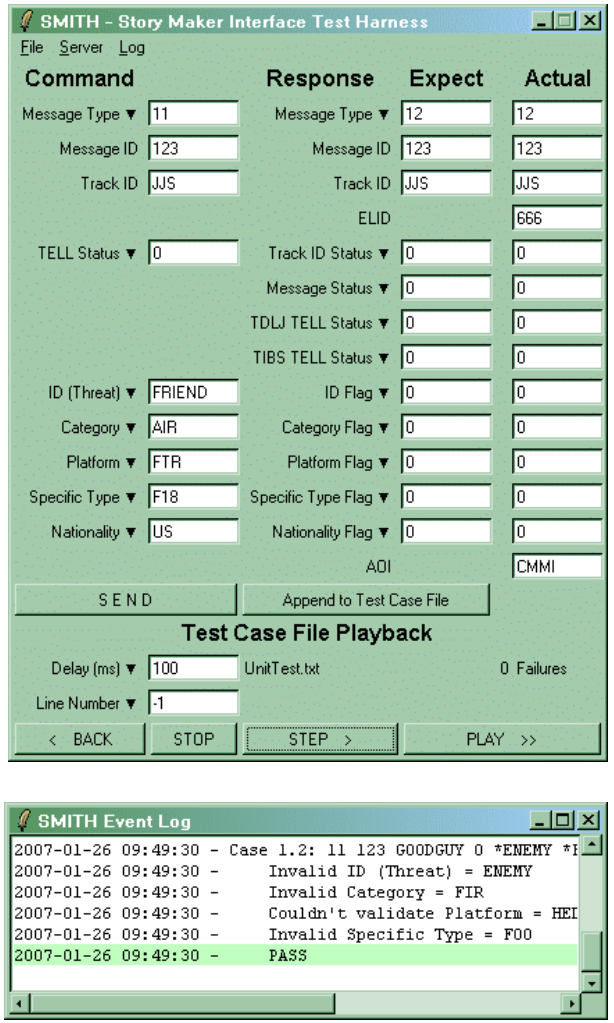
- Write GTIMF parser in Tcl
  - Message structure from ICD encoded in simple data structures
  - Helper procedures parse message hierarchically
  - Could easily be extended to other protocols
  - Less than a week to design, code, test, and document!

Once again, Tcl excels at network programming

Side-By-Side Comparison



Initial Sketch



Final Tool

Summary

- Tcl/Tk helped ensure success of initial effort
  - Simple enough for rapid prototyping
  - Flexible enough for changing requirements
  - Powerful enough for delivered applications

- Agent SMITH has been well received by multiple user communities and used in some unexpected ways
  - System Engineers use it to explore “What if?” scenarios
  - Software Engineers use it to debug other Teller changes
  - Test Engineers use it to provide repeatable stimulus

Tcl/Tk is ideal for developing tools and test harnesses