Microsoft Protocol Engineering Team

- Fred Wurden
- Dave MacDonald
- Keith Stobie
- Chris Kaler
- Nico Kicillof
- Wolfgang Grieskamp
- ... and many more
Part 1: Background
Microsoft’s Interoperability Initiative

Principles [http://www.microsoft.com/interop/principles]:

1. Open Connection to Microsoft Products
2. Support for Standards
3. Data Portability
4. Open Engagement
Microsoft’s Interoperability Initiative

Principles [http://www.microsoft.com/interop/principles]:

1. Open Connection to Microsoft Products
   • Focus of this talk: *Quality Assurance of Protocol Documentation*

2. Support for Standards

3. Data Portability

4. Open Engagement
Microsoft’s Technical Document Quality Assurance Efforts

1. Technical Documents
2. Protocol Parser Development
4. Protocol Test Suite Development
5. Assistance & Document Feedback
6. Document User

Licensees
Microsoft’s Technical Document Quality Assurance Efforts

Technical Documents

Protocol Parser Development

Protocol Test Suite Document Review

Protocol Test Suite Development

Assistance & Document Feedback

Document User
Document Quality Assurance Approach

- Protocol Parsers
- Protocol Models
- Protocol Test Suites

✔ Usability
✔ Accuracy

Microsoft Vendors

Technical Documents

Document Users

4/17/2008
Scope and Constraints

- In close sync with regulatory agencies in US and EU
- 250+ protocols in Windows alone (~ 30,000 pages of documentation)
  - Scope is extending (Office, .Net, ...)
- Clean-room approach
  - Vendors in India and China do test suite development
- Current investment
  - ~280 vendor employees for parser and test suite development
  - ~50 Microsoft employees for management and tools infrastructure
Part 2: Netmon and Protocol Parsers (in a nutshell)
Netmon: a network traffic analysis tool

- Can capture traffic on various interfaces
- Can do “almost real time” parsing
- Rich UI for analysis of traffic
- Based on declarative packet parser language
- Parsers will be moved to open source
  - MS-PL license
Netmon features

- Parsers defined hierarchically in script
  - Easily modified
  - Supports properties
  - Supports reassembly
  - Supports conversations
- Comes with SDK/API
Part 3: Protocol Test Suite Development
Testing of *Documents*

*Developing model and test suite*

- From document alone
- By vendor without previous MS internal knowledge

**Ensures:**

- Usability of document
  - Simulates developer situation
- Accuracy of document
  - Discovers discrepancy between document and implementation
Out of Scope

- Exhaustive implementation testing
- Stress/performance testing
- Certification testing
- Client behavior testing
Protocol Quality Assurance Process (PQAP)

- **Study phase**
  - Strategy review

- **Plan phase**
  - Plan review

- **Design phase**
  - Design review

- **Final phase**
  - Final review

PQA Report
PQAP Deliverables

- **Study**
  - Document study
  - Test strategy
  - Requirements spec (initial)
  - Estimates of Effort

- **Plan**
  - Requirements spec (final)
  - Test plan
    - Architecture
    - Scenarios

- **Design**
  - Test design
    - Model, or
    - Test case descriptions
    - Adapter
    - Design

- **Final**
  - Test case code
    - Generated, or
    - Manual
    - Adapter
    - Implementation
    - Test logs and report
    - Network captures
    - Test setup artifacts

Protocol Quality Assurance Report (PQAR)
Requirement Gathering

- Predefined template and guidelines for
  - Identifying requirements in the spec
  - Classifying them according to
    - verifiability criteria
    - verification strategy (manual, model, adapter)
- Requirement gathered from spec alone
  - Gatherers are not expert in the particular protocol
  - Any ambiguity or clarity issue filed as TDI
  - Resulting Requirement Spec reviewed by the reviewing team
  - Windows-specific behavior listed as separate requirements
- Requirement Spec used as input for model design
### Requirements Specification: SMB2

#### MS-SMB2-1700

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.5.15.7</td>
<td>When the server receives a request with an SMB2 header with a Command value equal to SMB2 IOCTL, and a CtrlCode not listed above, if the operation succeeds, the server MUST then construct an SMB2 IOCTL response with the following values: CtrlCode MUST be set to the CtrlCode of the request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-SMB2-1701</td>
<td>S46</td>
<td>Non-extension</td>
<td>Protocol</td>
<td>Server p0</td>
<td>Normative</td>
<td>Test Case</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### MS-SMB2-1717

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.5.15.7</td>
<td>When the server receives a request with an SMB2 header with a Command value equal to SMB2 IOCTL, and a CtrlCode not listed above, if the operation succeeds, the server MUST then construct an SMB2 IOCTL response with the following values: FileId MUST be set to Open.FileId.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-SMB2-1718</td>
<td>S46</td>
<td>Non-extension</td>
<td>Protocol</td>
<td>Server p0</td>
<td>Normative</td>
<td>Test Case</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### MS-SMB2-1719

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.5.15.7</td>
<td>When the server receives a request with an SMB2 header with a Command value equal to SMB2 IOCTL, and a CtrlCode not listed above, if the operation succeeds, the server MUST then construct an SMB2 IOCTL response with the following values: IOCTLOffset MUST be set to the offset, in bytes, from the beginning of the SMB2 header to the Buffer[] field of the response.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-SMB2-1720</td>
<td>S46</td>
<td>Non-extension</td>
<td>Protocol</td>
<td>Server p0</td>
<td>Normative</td>
<td>Adapter</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Protocol Quality Assurance Report (PQAR)

- Template based document which is incrementally produced
- Central point of documentation of progress
- After finalization, turns into test suite documentation for sustained engineering
4.3 TDI's Filed

SMB2_TDL_Final.psql

SMB2 team at ATC filed 90 TDI's totally, 71 are closed and 19 are active. Note that other teams outside ATC also contributed to file TDI's against SMB2.

The chart below depicts SMB2 TDI's filed in different phases.
PQAP Review Process

- Certified reviewers sign-off on phases of the PQAP
  - Internal and external industry experts which act independently
  - Formalized certification process via apprenticeship model with existing CRs

- Dispositions: re-review, conditional (after changes), accept
  - Quality not deadline oriented
  - Quality bar contains measurements like initial state of document, estimated relevance for document users, etc.
Part 4: Model-Based Testing
Test Suite Development with MBT

- Model
  - Test Cases
  - Test Oracle
    - Implementation
      - Generate
      - Control
      - Observe
      - Verdict

Feedback
MBT Process for Protocols

Define Messages
Basic validation and automatic adapter

Model Behavior
State machines and extended validation

Analyze Behavior
Visualization and model checking

Adapter / Test Execution

Test Suite

Generate Test Cases/Oracle
MBT Technology: Spec Explorer 2007

- Matured technology from Microsoft Research (first version 2002)
- Multiple modeling styles and languages (programs, patterns, diagrams)
- Extraction of state machine from infinite model
- Generation of test code from state machine
- Model composition
- Integration into Visual Studio
Model-Based Test Suite Development Process Drilldown

1. Understand Protocol
2. Define Actions
3. Define Adapters
4. Model & Explore
5. Generate Tests
6. Implement Adapters
7. Run Tests

Potential dependency on data types
Model-Based Test Suite Development Process Drilldown

1. Understand Protocol
2. Define Actions
   - Define Adapters
     - Potential dependency on data types
3. Model & Explore
4. Generate Tests
5. Implement Adapters
6. Run Tests
Test Adapters

Test Cases (traditional or MBT)

- Adapter Interface
- Adapter Implementation

Test Development

Adapter Development

SUT (System under test)

- Adapts SUT functionality
- Contract between teams
  - Test case team
  - Adapter team

- Pluggable
  - Different server setups
  - Different transports
- Choice of adapter implementation flavors
  - Interactive (automatic)
  - Script (set of commands)
  - Managed (.NET code)
  - RPC (automatic)
Methods represent test control

Events represent test observation
Model-Based Test Suite Development Process Drilldown

1. Understand Protocol
2. Define Actions
3. Model & Explore
4. Generate Tests
5. Define Adapters
6. Implement Adapters
7. Run Tests

Potential dependency on data types
Contract Model: SMB2

- Uses rich (infinite) model state
- Exploration slices an FSM
Test Selection: SMB2

```csharp
machine StateMachine() : Actions
{
    construct model program from Actions where namespace = "SMB2.Model"
    // construct contract model from C#
}

machine AllSync() : Actions
{
    // compose contract model with test purpose
    ( AssumeShareExists(1, ShareType.DISK); // assume one share
    SetupConnectionAndSession(1); // setup session (window=1)
    ...
    // wildcard from here
}
|| StateMachine
}

machine TestsForAllSync() : Actions
{
    // construct test cases
    construct test cases where strategy = "longtests" for AllSync
}```
Part 4: Conclusions
Summary

- Comprehensive measurements to ensure document quality
  - But: testing can only prove the presence of errors!
- Using advanced technologies and processes
  - Driving the state of the art in the area
  - Proving that MBT scales in industry testing
- Making technologies available to the community
  - Many papers published
  - Netmon freely released soon, Spec Explorer to followup
The End (Thanks!)

Q&A