Model-Based Quality Assurance of The SMB2 Protocol Documentation

Wolfgang Grieskamp
Architect, Protocol Engineering Team, Microsoft Corporation
Part of the company’s interoperability initiative

Addresses compliance requirements

Publicly available
Protocol Engineering Team

- Quality assurance of protocol documentation
  - Ensure accuracy and usability

- Unique challenge
  - Testing documents (not software)
  - Volume
    - 250+ protocols in Windows alone (25,000+ pages of documentation)
      - Scope is extending (Office, .Net, …)

- Requires innovative methods and tools
**Test-Driven Document Analysis**

- **Technical Document**
  - Normative statements

- **Formalization**
  - Validate consistency
  - Simulate implementation process

- **Test Suite**
  - Assertions (checked against the protocol)
Document Testing Approach

**Developing model and test suite**
- From technical document (TD) alone
- Black box
  - Only data on the wire is controlled and observed
- Clean-room approach
  - ~300 vendors in China and India
  - No previous knowledge of implementation details
  - ~50 Microsoft employees
    - Management and tools infrastructure

**Ensures:**
- **Usability of document**
  - Simulates protocol development conditions
- **Accuracy of document**
  - Discovers discrepancy between document and implementation
Out of Scope

- Exhaustive implementation testing
- Stress/performance testing
- Certification testing
- Documentation completeness testing
- Client behavior testing
Document Testing Process

Study phase
- Read TD
- Elicit requirements
- Suggest approach

Plan phase
- Classify requirements
- Plan model
- Suggest architecture

Design phase
- Create model
- Implement adapters

Final phase
- Generate tests
- Deploy test suite
- Execute and capture

PQA Report
Running Example: SMB2
Server Message Block Version 2

Model-Based Quality Assurance of the SMB2 Protocol Documentation
Requirements

- Gathered from TD alone
  - Gatherers are not expert in the particular protocol
  - Windows-specific behavior listed as separate requirements
- Predefined template and guidelines for
  - Identifying requirements (statements) in the TD
  - Classifying them according to
    - verifiability criteria
    - verification strategy (test suite, adapter)
- Requirements Specification (RS)
  - Reviewed by independent reviewers
  - Main input for model design
- Traceability
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
<th>Normative</th>
<th>Test Case</th>
<th>Adapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-SMB2-1700</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>
<td>Non-extension</td>
<td>Protocol</td>
<td>Server p0</td>
</tr>
<tr>
<td>MS-SMB2-1701</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 OpenFileId.</td>
<td>Non-extension</td>
<td>Protocol</td>
<td>Server p0</td>
</tr>
<tr>
<td>MS-SMB2-1702</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: InputOffset MUST be set to the offset, in bytes, from the beginning of the SMB2 header to the Buffer[] field of the response.</td>
<td>Non-extension</td>
<td>Protocol</td>
<td>Server p0</td>
</tr>
<tr>
<td>MS-SMB2-1717</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>
<td>Non-extension</td>
<td>Protocol</td>
<td>Server p0</td>
</tr>
<tr>
<td>MS-SMB2-1718</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 OpenFileId.</td>
<td>Non-extension</td>
<td>Protocol</td>
<td>Server p0</td>
</tr>
<tr>
<td>MS-SMB2-1719</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: InputOffset MUST be set to the offset, in bytes, from the beginning of the SMB2 header to the Buffer[] field of the response.</td>
<td>Non-extension</td>
<td>Protocol</td>
<td>Server p0</td>
</tr>
</tbody>
</table>
Protocol Quality Assurance Report (PQAR)

- Template-based document
- Incrementally produced
- Central point of documentation of progress
- After finalization, turns into test suite documentation for sustained engineering
4.3 TDIs Filed

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

The chart below depicts SMB2 TDIs filed in different phases.

![Chart showing TDIs filed in different phases]
Model-based testing (MBT)
A “light-weight” formal method

Spec/Reqs

Model

Test Cases

Test Oracle

Implementation

Generate

Control

Observe

Verdict

Feedback

Feedback
Mature technology from Microsoft Research
  - First version, 2002

Multiple modeling styles and languages
  - Programs, patterns, diagrams

State machine extraction from model program

Data generation (pairwise combinations, stochastic selection, etc.)

Test code generation from state machine

Model composition

IDE integration
Test Representation

- **Problem:** getting data on the wire and back
  - Known problem for protocol testing
  - E.g. TTCN-3

- **Our solution:** extension to Visual Studio Test Framework
  - Unit tests: concise test case representation
  - Test adapters written in a managed language (C#)

- **Protocol Test Framework (PTF)**
  - Custom support for dealing with protocols
  - Automatic data packet (de)serialization
    - Based on declarative definitions
  - Protocol-specific logging capacity (beacon packets)
Adapter interface: SMB2

```csharp
public interface ISmb2SetupAdapter : IAdapter
{
    void AssumeShareExists(int shareId, ShareType type);
    void AssumeShareDoesNotExist(int shareId);
    ...
}

public interface ISmb2Adapter : IAdapter
{
    void TreeConnectRequest(
        int relativeMessageId,
        int creditRequest, int shareId);
    event TreeConnectResponseHandler TreeConnectResponse;
    ...
}

public delegate void TreeConnectResponseHandler(int relativeMessageId,
    int creditResponse, int treeId, ShareType shareType);
```

- Methods represent stimuli (test control)
- Events represent responses (test observation)
Uses rich (infinite) model state

Exploration slices an FSM
Test Selection: SMB2

```plaintext
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
}
```
Conclusions

- Our process provides confidence in 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
  - Proof that MBT scales in industry testing
- Making technologies available to the community
  - Various publications
  - Network Monitor freely released
  - Spec Explorer to follow
The End (Thanks!)

Q&A