Using MBT to test Microsoft Lync Client conversation feature

Shirley Tan and Sergio Mera
Microsoft

ETSI MBT User Conference 2011, Berlin, Germany
Outline

• Lync and Spec Explorer (SE)
• Lync Test team and MBT: adoption strategy and blockers
• Model development and integration with Lync Test framework
• Takeaways of the experience and future testing plans using MBT
What is Lync?

• Microsoft Lync is a real time communication system that provides
  – Presence for self and contacts
  – Audio, video and instant messaging conferencing
  – Data collaboration: desktop and application sharing, white-boarding, etc..
• It allows workers to stay connected with others from virtually anywhere
• Lync is part of Microsoft Office and currently being used by millions of enterprise customers
Case study: Lync Client and MBT

• The Lync system has to deal with a high load of concurrent users across the network, interacting in many possible ways.

• This makes model-based testing a good approach for testing critical features of Lync.

• We focused on the Lync Client and used Spec Explorer to model APIs in the conversation feature first.
Adopting MBT in Lync Test team

- Lync Test team had no experience with MBT before
- Adoption is opt-in, so testers are not forced to use it
- We organized a MBT mini-workshop to facilitate adoption:
  - An initial kick-off SE model was developed as a proof of concept
  - A four 2-hour session workshop was carried out, involving other teams with experience in Spec Explorer
  - An online site with videos, meeting notes and training materials prepared
  - We encouraged testers to ask for help using the Microsoft Spec Explorer discussion list
Modeling feature states

- Active conversations and contacts define the states:
  
  ```java
  //Conversations
  public static SequenceContainer<Conversation> ConversationList;
  
  //Contacts
  public static SequenceContainer<Contact> ContactList;
  ```

- A contact contains contact data, as name, presence, email, phone numbers, etc.
- A conversation contains participants and modality type (IM, VoIP, Video, etc.)
Modeling the interactions

• We focused on some specific actions from main scenarios

• The model describes possible user interactions with the Conversation feature:
  – Start conversations with different types of contacts
    • Using contact object, SIP address, phone number, etc.
  – Send/receive an instant message
  – Start/stop a call
  – Mute/unmute a contact

• We used the abstract identifier pattern to map conversation and contact IDs to real entities
Slicing the model

• We restricted all possible sequences to interesting slices

```csharp
machine LynP2PConvoBVT() : Actions
{

    (let ContactIdentity c1, ContactIdentity c2
     where
     {
        ConvoCanvas.SetDomain(c1, c2);
     }
    in
    Initialize();
    StartConversationWithContact(c1, _);
    StartConversationWithContact(c2, _);
    DialPhoneNumber(InputString.Valid, _);
    (StartLyncCall? | SendIM(NormalSpecialString.Normal, _));
    StopCall;
    CleanUp;
)

    |||
    VerifyConversations*
}

• We also restricted the domain to a fixed list of users
Verifying the model

- State graphs were verified and added into test plans for the feature team to review
Integrating test cases with the test framework

- Generated test cases were successfully integrated with Lync Test Framework and run against Lync implementation
Takeaways

• Approximately 15 testers followed the ramp-up process on MBT
• Piloted two test features, and found spec issues and ambiguities
• Generated test cases were part of the Lync test suites
• Collected best practice data and gained experience
Takeaways

• Adoption curve showed to be slow as the mind shift involved is not negligible
• The starting audience was 15 people; only 8 finished the workshop
• People find hard to invest time in learning MBT when their day-to-day work is focused on traditional testing and fast result
• This seems to match other Microsoft groups that already ramped up on MBT and are using it regularly: the learning process takes some time
Future plan

• We plan to continue the adoption process for Spec Explorer within the Lync Team
• We plan to expand models and test coverage
• We identified possible owners for different Lync features who are willing to push MBT adoption within their local test groups
• We will use a set of metrics to be able to evaluate success, including number of DCRs, coverage, time spent on design/spec changes, etc.
Questions

Questions?