CLAIM PROCESSING SERVICE SIMULATION DEMO

This demo shows a composite application consisting of 2 SOAP services. It allows demonstrating how to record and simulate the behavior of one of SOAP services.

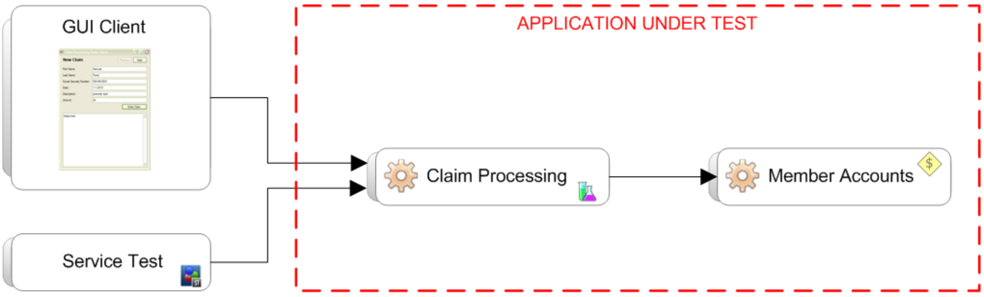

Fig. 1: The Composite Application

The *ClaimProcessing* service can be invoked either by the GUI client or by a Service Test script during the test – see Figure 1. We are going to record and simulate the *MemberAccounts* service since it is of limited accessibility.

THE USE CASE

The GUI client allows filling in a health insurance claim or selecting one of pre-filled claims.

The claim request contains partial identification of the insured person – some of first name, last name and/or social security number. The client sends an *enterClaim()* request to the *ClaimProcessing* SOAP service.

In order to create a new claim, the *ClaimProcessing* service has to complete the insured person information. It does so by invoking the *memberSearch()* on *MemberAccounts* SOAP service.

The *MemberAccounts* service looks up the person from pre-defined list of members and returns 0 or more member records containing member IDs.

The *ClaimProcessing* service creates a new claim if and only if it was possible to identify the person; it means when there was exactly one record returned by the *MemberAccounts* service, returning fault otherwise. From the predefined claim requests, this does not work for Karel Got (not a member) and a "Poirot" without first name (since there are both Mr. Hercule Poirot and his sister registered).

The *ClaimProcessing* calls the *MemberAccounts* service *getMemberPlan()* to get the member insurance plan containing the claim approval limit – the maximum amount of money to claim without going through further approval process. It also calls the *getMemberDetail()* to get the person contact information and other details to complete the claim.

The *ClaimProcessing* service marks the claim approved if the claimed amout was not higher than the limit in the member plan, then returns the claim identifier.

The client then requests the claim detail from *ClaimProcessing* service an displays the result.

USING REAL SERVICES

Use the `run-real.bat` script to launch the GUI client and both services on their default endpoints:

ClaimProcessing	http://localhost:8102/ServiceSimulation/Demo/ClaimProcessingService
MemberAccounts	http://localhost:8101/ServiceSimulation/Demo/MemberAccountsService

RECORDING AND SIMULATING THE MEMBER ACCOUNTS SERVICE

Use the `run-virtualized.bat` script to run the GUI client and both services configuring the *ClaimProcessing* service to use virtual *MemberAccounts* service instead of real one:

ClaimProcessing	http://localhost:8102/ServiceSimulation/Demo/ClaimProcessingService
MemberAccounts	http://localhost:8080/MemberAccounts

Open the prepared simulation project and switch the service into recording mode. The communication between ClaimProcessing and MemberAccounts services is being recorded – see Figure 2.

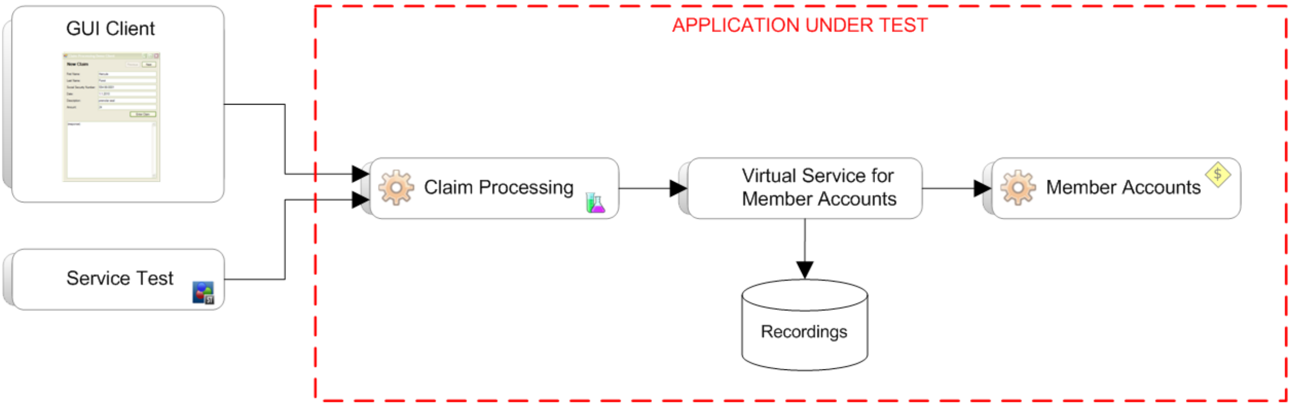

Fig. 2: Recording The Real Member Accounts Service

When you use the recording for simulation, the recording is processed and the resulting simulation model is used to simulate the behavior of *MemberAccounts* service. The real MemberAccounts service can be shut off - See Figure 3.

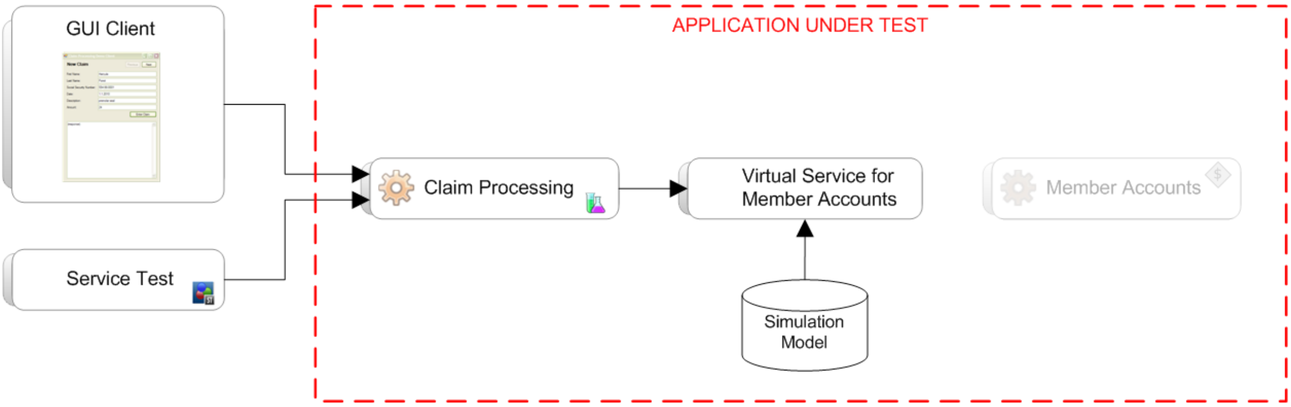

Fig. 3: Simulating The Member Accounts Service