IMMoRTALS Phase III CP I Exemplar Software System (P3CP1-ESS)

Adapting software systems to schema changes

Authored by Jacob Staples (*jstaples@securboration.com*)

Purpose of this document

This document provides an overview of an Exemplar Software System (ESS) developed for the purpose of evaluating the IMMoRTALS repair system in our team's Challenge Problem 1.

Purpose of the ESS

The ESS demonstrates issues that arise when assumptions made by application developers about schema compliance are baked-in to application code and subsequently broken by changing schemas.

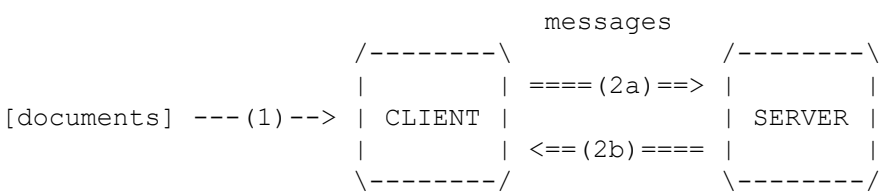
ESS project structure

- Non-code artifacts
    - *schema*
        - *schema/v1* contains a schema used in a WSDL with dependencies on MDL_v0_8_17.xsd. This was derived from SwRI's MDL_v0_8_17_with_Examples.zip.
        - *schema/v2* contains a schema used in a WSDL with dependencies on MDL_v0_8_19.xsd. This was derived from SwRI's MDL_v0_8_19_with_Examples.zip.
        - *schema/v3* currently a copy of schema/v2
    - *messages*
        - *messages/v1* contains several representative XML messages conformant with MDL_v0_8_17.xsd. These were copied from SwRI's MDL_v0_8_17_with_Examples.zip. **NOTE** that one of the original XML documents, VariableLengthPackages.xml, has validation issues and has been disabled for the time being.
        - *messages/v2* contains several representative XML messages conformant with MDL_v0_8_19.xsd. These were copied from SWRI's MDL_v0_8_19_with_Examples.zip.
        - *messages/v3* currently a copy of messages/v3
- Code artifacts
    - *schema-code*
        - *schema-code/v1* contains autogenerated Java types derived from schema/v1
        - *schema-code/v2* contains autogenerated Java types derived from schema/v2
        - *schema-code/v3* contains autogenerated Java types derived from schema/v3
    - *server* is a Java spring-boot app that runs an embedded Tomcat instance that listens for SOAP/XML messages containing MDL-conformant documents
    - *client* is a Java spring-boot client that converses with a server using MDL-conformant documents

Architecture and evolutionary pressure

The dataflows along which adaptation may need to occur are illustrated below.

SOAP/XML



1. This dataflow involves the client reading MDL documents from the local file system of the platform on which it executes. See client:com.securboration.client.ClientRunner (line 101). If the schema version of the documents differs from the client, adaptation will be required along dataflow 1.
2. This dataflow involves (2a) the client transmitting an MDL document to the server and (2b) receiving a response. See client:com.securboration.client.MessageListenerClient (line 44). If the schema version of the client differs from that of the server, adaptation will be required along 2a and 2b.

Building the ESS

Prerequisites

1. Gradle (tested with 4.9) (validate with gradle -version) **OR** Maven (tested with version 3.3.9) (validate with `mvn -version`)
2. Java (tested with 1.8.0_161). Validate with `java -version`
3. Connectivity to the Internet. This is a build-only requirement--executing the ESS does not require Internet access.
4. The ESS codebase in a directory subsequently referred to as `$ESS_HOME`

Procedure (Gradle)

All commands are executed from `$ESS_HOME`

Building the software requires selecting client and server MDL versions (which may be different). The client and server MDL versions are specified by editing `CLIENT_SCHEMA_VERSION` and `SERVER_SCHEMA_VERSION` in `$ESS_HOME/gradle.properties`.

For example:

- Server uses V2 and client uses V1: `CLIENT_SCHEMA_VERSION = v1` and `SERVER_SCHEMA_VERSION = v1`
- Server uses V3 and client uses V2: `CLIENT_SCHEMA_VERSION = v2` and `SERVER_SCHEMA_VERSION = v3`
- Server uses V3 and client uses V3: `CLIENT_SCHEMA_VERSION = v3` and `SERVER_SCHEMA_VERSION = v3`

To build the software after setting these variables, execute the following command: `gradle clean build`

NOTE that your first build may be quite time consuming as dependencies are downloaded and cached (subsequent builds should be much faster).

Procedure (Maven)

All commands are executed from `$ESS_HOME`

Building the software requires selecting client and server MDL versions (which may be different). The client and server MDL versions are specified using maven profiles indicated via the `-P` flag. For example, to use a v1 client

and v1 server, the software would be built using `mvn clean install -P clientSchemaV1,serverSchemaV1`. Evolutionary pressure is applied by using different schema versions.

The general format for an ESS Maven build is therefore `mvn clean install -P clientSchemaX,serverSchemaY` where X and Y are one of {v1, v2, v3}

Other examples:

- Server uses V2 and client uses V1: `mvn clean install -P clientSchemaV1,serverSchemaV2`
- Server uses V3 and client uses V2: `mvn clean install -P clientSchemaV2,serverSchemaV3`
- Server uses V3 and client uses V3: `mvn clean install -P clientSchemaV3,serverSchemaV3`

NOTE that your first build may be quite time consuming as dependencies are downloaded and cached (subsequent builds should be much faster).

Running the ESS

All commands are executed from `$ESS_HOME`

Prerequisites

1. Java 8 (tested with 1.8.0_161). Java 9 and Java 10 are not currently supported. Validate with `java -version`

Running the server

```
java -jar ./server/target/immortals-cp3.1-server-1.0.0.jar
```

By default the server starts on port 8080. If another port is desired, it can be specified like so:

```
java -Dserver.port=8081 -jar ./server/target/immortals-cp3.1-server-1.0.0.jar
```

Running the client

```
java -DMESSAGES_TO_SEND_DIR=./messages/v1 -DREPORT_DIR=./report -DSERVER_ENDPOINT_URL=http://
```

The MESSAGES_TO_SEND_DIR property specifies a directory containing XML documents compliant to V1, V2, or V3 as specified for the client during the build (the use of a directory containing a V1 document with a V2 client would be considered an incorrect use of the software and not evolutionary pressure). In the example above, V1 messages will be used.

The client traverses this directory and recursively visits any XML documents found. Each document is modified by the client and transmitted to the server (note that for builds with a different client and server version, this will cause the code to break). Each document is treated as a separate test case and the number of passing test cases are emitted to ./report (specified by -DREPORT_DIR) after processing all documents.

The SERVER_ENDPOINT_URL property specifies the endpoint of the server. You may need to modify the host and/or port.

Sample report output with client=v1 and server=v1 (everything works)

```
server URI: http://localhost:8080/ws
server ping (millis): 7
expected MDL schema version: MDL_v0_8_17.xsd
starting test run with input dir: ./messages/v1
test1 input: ./messages/v1/AssetAssociations.xml
test1 result: test PASSED
test2 input: ./messages/v1/Bandpass-Measurement.xml
test2 result: test PASSED
test3 input: ./messages/v1/Data-Stream-Ch10.xml
test3 result: test PASSED
test4 input: ./messages/v1/Extract-Bus-Measurements.xml
test4 result: test PASSED
test5 input: ./messages/v1/Mapping-24bit-measurement-into-16bit-fields.xml
test5 result: test PASSED
test6 input: ./messages/v1/Measurements-and-Data-Operations.xml
test6 result: test PASSED
test7 input: ./messages/v1/Measurements-with-TMATS.xml
test7 result: test PASSED
test8 input: ./messages/v1/MissionQoS.xml
test8 result: test PASSED
test9 input: ./messages/v1/MultipleAppsPerNetworkNode.xml
test9 result: test PASSED
test10 input: ./messages/v1/Network-Example.xml
test10 result: test PASSED
test11 input: ./messages/v1/PackagingMeasurements.xml
test11 result: test PASSED
test12 input: ./messages/v1/PackagingPCMStream.xml
test12 result: test PASSED
test13 input: ./messages/v1/PCM-Example.xml
test13 result: test PASSED
test14 input: ./messages/v1/PCM-with-TMATS.xml
test14 result: test PASSED
test15 input: ./messages/v1/Select-Bus-Measurements.xml
test15 result: test PASSED
test16 input: ./messages/v1/Standalone-Example.xml
test16 result: test PASSED
done with tests
# tests performed: 16
# tests that passed: 16
# tests that failed: 0
pass rate: 1.0
fail rate: 0.0
score: 1.0
overall elapsed time (millis): 5562
```

Sample report output with client=v1 and server=v2 (the client barely works)

```
server URI: http://localhost:8080/ws
server ping (millis): 9
expected MDL schema version: MDL_v0_8_17.xsd
starting test run with input dir: ./messages/v1
test1 input: ./messages/v1/AssetAssociations.xml
test1 result: test FAILED due to an exception of type org.springframework.ws.soap.client.SoapFa
test2 input: ./messages/v1/Bandpass-Measurement.xml
test2 result: test FAILED due to an exception of type org.springframework.ws.soap.client.SoapFa
test3 input: ./messages/v1/Data-Stream-Ch10.xml
test3 result: test FAILED due to an exception of type org.springframework.ws.soap.client.SoapFa
test4 input: ./messages/v1/Extract-Bus-Measurements.xml
test4 result: test FAILED due to an exception of type org.springframework.ws.soap.client.SoapFa
test5 input: ./messages/v1/Mapping-24bit-measurement-into-16bit-fields.xml
test5 result: test FAILED due to an exception of type org.springframework.ws.soap.client.SoapFa
test6 input: ./messages/v1/Measurements-and-Data-Operations.xml
test6 result: test FAILED due to an exception of type org.springframework.ws.soap.client.SoapFa
test7 input: ./messages/v1/Measurements-with-TMATS.xml
```

```
test7 result test FAILED due to an exception of type org.springframework.ws.soap.client.SoapFa
test8 input ./messages/v1/MissionQoS.xml
test8 result test FAILED due to an exception of type org.springframework.ws.soap.client.SoapFa
test9 input ./messages/v1/MultipleAppsPerNetworkNode.xml
test9 result test FAILED due to an exception of type org.springframework.ws.soap.client.SoapFa
test10 input ./messages/v1/Network-Example.xml
test10 result test FAILED due to an exception of type org.springframework.ws.soap.client.SoapF
test11 input ./messages/v1/PackagingMeasurements.xml
test11 result test FAILED due to an exception of type org.springframework.ws.soap.client.SoapF
test12 input ./messages/v1/PackagingPCMStream.xml
test12 result test FAILED due to an exception of type org.springframework.ws.soap.client.SoapF
test13 input ./messages/v1/PCM-Example.xml
test13 result test FAILED due to an exception of type org.springframework.ws.soap.client.SoapF
test14 input ./messages/v1/PCM-with-TMATS.xml
test14 result test FAILED due to an exception of type org.springframework.ws.soap.client.SoapF
test15 input ./messages/v1/Select-Bus-Measurements.xml
test15 result test FAILED due to an exception of type org.springframework.ws.soap.client.SoapF
test16 input ./messages/v1/Standalone-Example.xml
test16 result test PASSED
done with tests
# tests performed 16
# tests that passed 1
# tests that failed 15
pass rate 0.0625
fail rate 0.9375
score 0.0625
overall elapsed time (millis) 1793
```