Testing of Service Oriented Architectures – A practical approach / APPENDIX V1.0

Schahram Dustdar, Stephan Haslinger

[1] Distributed Systems Group, Vienna University of Technology
dustdar@infosys.tuwien.ac.at
[2] UCS GmbH, Vienna, Austria
stephan.haslinger@ucs.at

This appendix gives an explanation to the XML meta language used by TMSS.

INDEX

---

1 XSD for Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:complexType name="actionType">
        <xs:sequence>
            <xs:element ref="connect_to"/>
            <xs:element ref="commandline"/>
        </xs:sequence>
        <xs:attribute name="action_id" use="required" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="after_testType">
        <xs:sequence>
            <xs:element name="action" type="actionType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="before_testType">
        <xs:sequence>
            <xs:element name="action" type="actionType"/>
        </xs:sequence>
    </xs:complexType>
```

```
<xs:element name="commandline" type="xs:string"/>
<xs:element name="commandline_start" type="xs:string"/>
<xs:element name="commandline_stop" type="xs:string"/>
<xs:element name="connect_to" type="xs:string"/>
<xs:element name="connection_protocol">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="ftp"/>
            <xs:enumeration value="local"/>
            <xs:enumeration value="telnet"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="connection_timeout" type="xs:short"/>
<xs:element name="ip_address">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="listenport" type="xs:long"/>
<xs:element name="log_file_name" type="xs:string"/>
<xs:complexType name="log_file_propertiesType">
    <xs:sequence>
        <xs:element ref="log_file_name"/>
        <xs:element ref="time_waiting_for_log_file"/>
        <xs:element ref="unique_pattern"/>
        <xs:element ref="timestamp_format"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ma_hostType">
    <xs:sequence>
        <xs:element ref="ip_address"/>
        <xs:element ref="listenport"/>
        <xs:element ref="connection_protocol"/>
        <xs:element ref="connection_timeout"/>
        <xs:element ref="port"/>
        <xs:element ref="commandline_start"/>
        <xs:element ref="commandline_stop"/>
    </xs:sequence>
    <xs:attribute name="hostid" type="xs:string" use="required"/>
</xs:complexType>
<xs:element name="ma_jdbc_connection_string" type="xs:string"/>
<xs:complexType name="ma_propertiesType">
    <xs:sequence>
        <xs:element name="ma_host" type="ma_hostType"/>
        <xs:element ref="ma_jdbc_connection_string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="masteragentsType">
    <xs:sequence>
        <xs:element name="ma_properties" type="ma_propertiesType"/>
    </xs:sequence>
    <xs:attribute name="agentid" type="xs:string" use="required"/>
</xs:complexType>
<xs:element name="port" type="xs:short"/>
<xs:complexType name="taType">
    <xs:sequence>
        <xs:element name="ta_properties" type="ta_propertiesType"/>
    </xs:sequence>
    <xs:attribute name="agentid" use="required" type="xs:string"/>
</xs:complexType>
<xs:complexType name="ta_hostType">
    <xs:sequence>
        <xs:element ref="ip_address"/>
        <xs:element ref="connection_protocol"/>
```

```
            <xs:element ref="connection_timeout" minOccurs="0"/>
            <xs:element ref="port" minOccurs="0"/>
            <xs:element ref="commandline_start" minOccurs="0"/>
            <xs:element ref="commandline_stop" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="hostid" use="required" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="ta_propertiesType">
        <xs:sequence>
            <xs:element ref="type"/>
            <xs:element name="ta_host" type="ta_hostType"/>
            <xs:element name="log_file_properties" type="log_file_propertiesType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="testagentsType">
        <xs:sequence>
            <xs:element name="ta" type="taType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="time_waiting_for_log_file" type="xs:short"/>
    <xs:element name="timestamp_format" type="xs:string"/>
    <xs:element name="tmss_configuration">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="before_test" type="before_testType"/>
                <xs:element name="after_test" type="after_testType"/>
                <xs:element name="testagents" type="testagentsType"/>
                <xs:element name="masteragents" type="masteragentsType"/>
            </xs:sequence>
            <xs:attribute name="test_scenario" type="xs:string" use="required"/>
            <xs:attribute name="test_id" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="type">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="batch"/>
                <xs:enumeration value="online"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="unique_pattern" type="xs:string"/>
</xs:schema>
```

1.1 Example for Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<tmss_configuration test_scenario="MNP_TEST_FOR_ECOWS" test_id="MNP1">
    <before_test>
        <action action_id="ad1">
            <connect_to>host1</connect_to>
            <commandline>nohup /usr/bin/service1/startservice1.ksh >/usr/bin/service1/log.txt</commandline>
        </action>
    </before_test>
    <after_test>
        <action action_id="ad2">
            <connect_to>host1</connect_to>
            <commandline>/usr/bin/service1/stopservice1.ksh</commandline>
        </action>
    </after_test>
    <testagents>
        <ta agentid="1">
            <ta_properties>
                <type>online</type>
```

```
            <ta_host hostid="host1">
                <ip_address>192.168.0.2</ip_address>
                <connection_protocol>telnet</connection_protocol>
                <connection_timeout>10</connection_timeout>
                <port>21</port>
                <commandline_start>/usr/bin/startta.sh -ma 192.168.0.1 -lp 8200</commandline_start>
                <commandline_stop>/usr/bin/stopta.sh -ma 192.168.0.1</commandline_stop>
            </ta_host>
            <log_file_properties>
                <log_file_name>/usr/bin/service1/log.txt</log_file_name>
                <time_waiting_for_log_file>30</time_waiting_for_log_file>
                <unique_pattern>tmss-MESSAGE:</unique_pattern>
                <timestamp_format>YYYY-MM-DD HH24:MI:SS</timestamp_format>
            </log_file_properties>
        </ta_properties>
    </ta>
    <ta agentid="ta_sap_telco_a">
        <ta_properties>
            <type>online</type>
            <ta_host hostid="sap_host1">
                <ip_address>192.168.0.17</ip_address>
                <connection_protocol>telnet</connection_protocol>
                <connection_timeout>10</connection_timeout>
                <port>21</port>
                <commandline_start>/usr/bin/startta.sh -ma 192.168.0.1 -lp 8200</commandline_start>
                <commandline_stop>/usr/bin/stopta.sh -ma 192.168.0.1</commandline_stop>
            </ta_host>
            <log_file_properties>
                <log_file_name>/usr/bin/service1/log.txt</log_file_name>
                <time_waiting_for_log_file>30</time_waiting_for_log_file>
                <unique_pattern>tmss-MESSAGE:</unique_pattern>
                <timestamp_format>YYYY-MM-DD HH24:MI:SS</timestamp_format>
            </log_file_properties>
        </ta_properties>
    </ta>
    <ta agentid="2">
        <ta_properties>
            <type>batch</type>
            <ta_host hostid="host2">
                <ip_address>127.0.0.1</ip_address>
                <connection_protocol>local</connection_protocol>
            </ta_host>
            <log_file_properties>
                <log_file_name>/home/testagent/test.log</log_file_name>
                <time_waiting_for_log_file>30</time_waiting_for_log_file>
                <unique_pattern>tmss-MESSAGE:</unique_pattern>
                <timestamp_format>YYYY-MM-DD HH24:MI:SS</timestamp_format>
            </log_file_properties>
        </ta_properties>
    </ta>
    <ta agentid="test_agent_telco_c">
        <ta_properties>
            <type>batch</type>
            <ta_host hostid="host3">
                <ip_address>10.12.14.12</ip_address>
                <connection_protocol>ftp</connection_protocol>
                <connection_timeout>20</connection_timeout>
                <port>23</port>
            </ta_host>
            <log_file_properties>
                <log_file_name>/home/testagent/test.log</log_file_name>
                <time_waiting_for_log_file>30</time_waiting_for_log_file>
                <unique_pattern>tmss-MESSAGE:</unique_pattern>
                <timestamp_format>YYYY-MM-DD HH24:MI:SS</timestamp_format>
            </log_file_properties>
        </ta_properties>
    </ta>
```

```
        <ta agentid="ta_telco_d">
            <ta_properties>
                <type>batch</type>
                <ta_host hostid="host4">
                    <ip_address>194.232.0.12</ip_address>
                    <connection_protocol>ftp</connection_protocol>
                    <connection_timeout>20</connection_timeout>
                    <port>23</port>
                </ta_host>
                <log_file_properties>
                    <log_file_name>/home/testagent/test.log</log_file_name>
                    <time_waiting_for_log_file>30</time_waiting_for_log_file>
                    <unique_pattern>tmss-MESSAGE:</unique_pattern>
                    <timestamp_format>YYYY-MM-DD HH24:MI:SS</timestamp_format>
                </log_file_properties>
            </ta_properties>
        </ta>
        <ta agentid="ta_telco_e">
            <ta_properties>
                <type>batch</type>
                <ta_host hostid="host5">
                    <ip_address>193.170.151.20</ip_address>
                    <connection_protocol>ftp</connection_protocol>
                    <connection_timeout>20</connection_timeout>
                    <port>23</port>
                </ta_host>
                <log_file_properties>
                    <log_file_name>/home/testagent/test.log</log_file_name>
                    <time_waiting_for_log_file>30</time_waiting_for_log_file>
                    <unique_pattern>tmss-MESSAGE:</unique_pattern>
                    <timestamp_format>YYYY-MM-DD HH24:MI:SS</timestamp_format>
                </log_file_properties>
            </ta_properties>
        </ta>
    </testagents>
    <masteragents agentid="ma_1">
        <ma_properties>
            <ma_host hostid="ma_host1">
                <ip_address>192.168.0.1</ip_address>
                <listenport>8200</listenport>
                <connection_protocol>telnet</connection_protocol>
                <connection_timeout>10</connection_timeout>
                <port>21</port>
                <commandline_start>/usr/bin/startma.sh -lp 8200</commandline_start>
                <commandline_stop>/usr/bin/stopma.sh</commandline_stop>
            </ma_host>
            <ma_jdbc_connection_string>jdbc:oracle:thin:@192.168.0.1:1521:TAS</ma_jdbc_connection_string>
        </ma_properties>
    </masteragents>
</tmss_configuration>
```

1.2 Explanation for Configuration

tmss_configuration	Head element for the configuration XML file.	
	Attributes	
	test_scenario	Unique name, showing in which test this configuration will be used. A reference to the unique attribute “test_scenario”, which is used in the head element “tmss-test” in the test XML file.
	test_id	Unique id, showing in which test this configuration will be used. A reference to the unique attribute “test_id”, which is used in the head element “tmss-test” in the test XML file.

before_test	This element holds necessary information which programs should get executed before starting the test. With this element it is possible to indicate programs and scripts, which should be started in advance.
> action	An element indicating which action should be taken. *Attributes* action_id — Unique id for this action
>> connect_to	The machine where the script has to be executed. It is a reference to the element “ta_host”.
>> command_line	The command line for the script to get started.

test_agents	Head element for test agents.
> ta	Group element for test agents. *Attributes* agent_id — Unique id for this test agent.
>> ta_properties	Properties for a certain test agent.
>>> type	Shows the type of the agent. *Possible Values* Online — If the agent is able to connect to the master agent Batch — If the agent is not able to connect to the master agent. In this case the log files have to be transferred manually or automatically via ftp.
>>> ta_host	Host, where the test agent is running. *Attributes* host_id — Unique id for this machine.
>>>> ip_address	IP address of the machine. Besides the IP address also the name of the machine is possible, if it can be resolved by a DNS server.
>>>> connection_protocol	Possible Values telnet — This will be the case if the type of the agent is “online” ftp — This will be the case if the type of the agent is “batch” and the log file can be retrieved via ftp local — If the file gets retrieved manually. The “log_file_name” then has to point to the log file on th e local machine. This is the machine where the test deamon is running
>>>> connection_timeout	Timeout in seconds for this connection.
>>>> port	Port for the connection.
>>>> commandline_start	The command line for starting the test agent. This is just used if the agent is of type “online”.
>>>> commandline_stop	The command line for stopping the test agent. This is just used if the agent is of type “online”.
>>> log_file_properties	Properties for a certain log file.
>>>> log_file_name	The name if the log file, including the absolute path.
>>>> time_waiting_for_log_file	After the agent got started it is possible that the log file is not in place, as the service may begin to create the file later. This gives the time how long the agent should wait for the log file until it raises an error.
>>>> unique_pattern	The pattern the agent should search for in the log file. We assume to have “TMSS-MESSAGE”. With this element it is configurable.

>>>> *timestamp_format*	The format of the timestamp.

master_agents	Head element for master agents.
> *ma_properties*	Properties for a certain master agent.
>> *ma_host*	Host, where the master agent is running. *Attributes* agent_id — Unique id for this master agent.
>>> *ip_address*	IP address of the machine. Besides the IP address also the name of the machine is possible, if it can be resolved by a DNS server.
>>> *listenport*	The port on which the master agent is listening for connection requests of the test agents.
>>> *connection_protocol*	Possible Values telnet — The deamon will connect to the machine via telnet and start the master agent.
>>> *connection_timeout*	Timeout in seconds for this connection.
>>> *port*	Port for the connection.
>>> *commandline_start*	The command line for starting the master agent.
>>> *commandline_stop*	The command line for stopping the master agent.
>> *ma_jdbc_connection_string*	The connection string for the test database.

2 XSD for Testing

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:complexType name="beforeType">
        <xs:sequence>
            <xs:element name="flow" type="flowType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="configuration_file" type="xs:string"/>
    <xs:complexType name="dependenciesType">
        <xs:sequence>
            <xs:element name="order_dependency" type="order_dependencyType"/>
            <xs:element name="time_dependency" type="time_dependencyType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="endpointsType">
        <xs:sequence>
            <xs:element name="service_endpoint" type="service_endpointType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="fixed_orderType">
        <xs:sequence>
            <xs:element name="message" type="messageType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="flowType">
        <xs:sequence>
            <xs:element ref="flow_name" minOccurs="0"/>
            <xs:element ref="flow_type" minOccurs="0"/>
```

```
            <xs:element name="sources" type="sourcesType" minOccurs="0"/>
            <xs:element name="targets" type="targetsType" minOccurs="0"/>
            <xs:element ref="message_subject" minOccurs="0"/>
            <xs:element ref="message_id" minOccurs="0"/>
            <xs:element name="message_order" type="message_orderType" minOccurs="0"/>
            <xs:element name="before" type="beforeType" minOccurs="0"/>
            <xs:element name="next" type="nextType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="flow_id" use="required" type="xs:string"/>
    </xs:complexType>
    <xs:element name="flow_name">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="Customer_sucessfully_ported"/>
                <xs:enumeration value="port_Customer_Request"/>
                <xs:enumeration value="port_Customer_Response"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:complexType name="flow_orderType">
        <xs:sequence>
            <xs:element name="flow" type="flowType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="flow_type">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="point_to_point"/>
                <xs:enumeration value="publish-subscribe"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:complexType name="from_invocation_flowType">
        <xs:attribute name="flow_id" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="from_termination_flowType">
        <xs:attribute name="flow_id" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="latencyType">
        <xs:sequence>
            <xs:element name="from_invocation_flow" type="from_invocation_flowType" minOccurs="0"/>
            <xs:element name="from_termination_flow" type="from_termination_flowType" minOccurs="0"/>
            <xs:element name="to_termination_flow" type="to_termination_flowType"/>
            <xs:element ref="max_seconds"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="latency_from_source_to_targetType">
        <xs:sequence>
            <xs:element ref="max_seconds"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="max_seconds" type="xs:long"/>
    <xs:complexType name="messageType">
        <xs:sequence>
            <xs:element name="source_service_endpoint" type="source_service_endpointType"/>
            <xs:element name="target_service_endpoint" type="target_service_endpointType"/>
            <xs:element name="latency_from_source_to_target" type="latency_from_source_to_targetType" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="message_id" type="xs:long"/>
    <xs:complexType name="message_orderType">
        <xs:sequence>
            <xs:element name="fixed_order" type="fixed_orderType"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:element name="message_subject">
```

```
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="Customer_sucessfully_ported"/>
                <xs:enumeration value="port_Cust_Request"/>
                <xs:enumeration value="port_Customer_Request"/>
                <xs:enumeration value="port_Customer_Response"/>
                <xs:enumeration value="port_Request_from_other_telco"/>
                <xs:enumeration value="port_Response"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="name" type="xs:string"/>
    <xs:complexType name="nextType">
        <xs:sequence>
            <xs:element name="flow" type="flowType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="order_dependencyType">
        <xs:sequence>
            <xs:element name="flow_order" type="flow_orderType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="service_endpointType">
        <xs:sequence>
            <xs:element ref="name"/>
            <xs:element ref="ta_agent_id"/>
        </xs:sequence>
        <xs:attribute name="service_id" use="required" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="source_service_endpointType">
        <xs:sequence>
            <xs:element ref="message_subject" minOccurs="0"/>
            <xs:element ref="message_id" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="service_id" use="required" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="sourcesType">
        <xs:sequence>
            <xs:element name="source_service_endpoint" type="source_service_endpointType" maxOccurs="unbounded"/>
            <xs:element ref="message_subject" minOccurs="0"/>
            <xs:element ref="message_id" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ta_agent_id" type="xs:string"/>
    <xs:complexType name="target_service_endpointType">
        <xs:sequence>
            <xs:element ref="message_subject" minOccurs="0"/>
            <xs:element ref="message_id" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="service_id" use="required" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="targetsType">
        <xs:sequence>
            <xs:element name="target_service_endpoint" type="target_service_endpointType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="time_dependencyType">
        <xs:sequence>
            <xs:element name="latency" type="latencyType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="tmss-test">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="configuration_file"/>
                <xs:element name="endpoints" type="endpointsType"/>
                <xs:element name="workflow" type="workflowType"/>
```

```
                    <xs:element name="dependencies" type="dependenciesType"/>
                </xs:sequence>
                <xs:attribute name="test_scenario" type="xs:string" use="required"/>
                <xs:attribute name="test_id" type="xs:string" use="required"/>
            </xs:complexType>
        </xs:element>
        <xs:complexType name="to_termination_flowType">
            <xs:attribute name="flow_id" use="required" type="xs:string"/>
        </xs:complexType>
        <xs:complexType name="workflowType">
            <xs:sequence>
                <xs:element ref="workflow_name"/>
                <xs:element name="flow" type="flowType" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="wf_id" type="xs:string" use="required"/>
        </xs:complexType>
        <xs:element name="workflow_name" type="xs:string"/>
</xs:schema>
```

2.1 Example for Testing

```
<?xml version="1.0" encoding="UTF-8"?>
<tmss-test test_scenario="MNP_TEST_FOR_ECOWS" test_id="MNP1">
    <configuration_file>/usr/testsample/configuration_example.xml</configuration_file>
    <endpoints>
        <service_endpoint service_id="1">
            <name>telco_a</name>
            <ta_agent_id>1</ta_agent_id>
        </service_endpoint>
        <service_endpoint service_id="2">
            <name>telco_b</name>
            <ta_agent_id>2</ta_agent_id>
        </service_endpoint>
        <service_endpoint service_id="3">
            <name>telco_c</name>
            <ta_agent_id>test_agent_telco_c</ta_agent_id>
        </service_endpoint>
        <service_endpoint service_id="4">
            <name>telco_d</name>
            <ta_agent_id>ta_telco_d</ta_agent_id>
        </service_endpoint>
        <service_endpoint service_id="5">
            <name>telco_e</name>
            <ta_agent_id>ta_telco_e</ta_agent_id>
        </service_endpoint>
        <service_endpoint service_id="6">
            <name>sap_telco_a</name>
            <ta_agent_id>ta_sap_telco_a</ta_agent_id>
        </service_endpoint>
    </endpoints>
    <workflow wf_id="wf1">
        <workflow_name>MNP_TEST_WITH_EXTERNAL_TELCOS</workflow_name>
        <flow flow_id="fl1_wf1">
            <flow_name>port_Customer_Request</flow_name>
            <flow_type>point_to_point</flow_type>
            <sources>
                <source_service_endpoint service_id="1"/>
                <message_subject>port_Cust_Request</message_subject>
                <message_id>454987624</message_id>
            </sources>
            <targets>
                <target_service_endpoint service_id="2">
                    <message_subject>port_Request_from_other_telco</message_subject>
```

```
                    <message_id>003453550</message_id>
                </target_service_endpoint>
                <target_service_endpoint service_id="3"/>
                <target_service_endpoint service_id="4"/>
                <target_service_endpoint service_id="5"/>
            </targets>
            <message_subject>port_Customer_Request</message_subject>
            <message_id>004365412121212</message_id>
            <message_order id="mo1">
                <fixed_order>
                    <message>
                        <source_service_endpoint service_id="1"/>
                        <target_service_endpoint service_id="2"/>
                        <latency_from_source_to_target>
                            <max_seconds>20</max_seconds>
                        </latency_from_source_to_target>
                    </message>
                    <message>
                        <source_service_endpoint service_id="1"/>
                        <target_service_endpoint service_id="3"/>
                    </message>
                    <message>
                        <source_service_endpoint service_id="1"/>
                        <target_service_endpoint service_id="4"/>
                    </message>
                </fixed_order>
            </message_order>
        </flow>
        <flow flow_id="fl2_wf1">
            <flow_name>port_Customer_Response</flow_name>
            <flow_type>point_to_point</flow_type>
            <sources>
                <source_service_endpoint service_id="2">
                    <message_subject>port_Response</message_subject>
                    <message_id>003453550</message_id>
                </source_service_endpoint>
                <source_service_endpoint service_id="3"/>
                <source_service_endpoint service_id="4"/>
                <source_service_endpoint service_id="5"/>
            </sources>
            <targets>
                <target_service_endpoint service_id="1"/>
            </targets>
            <message_subject>port_Customer_Response</message_subject>
            <message_id>004365412121212</message_id>
        </flow>
        <flow flow_id="fl3_wf1">
            <flow_name>Customer_sucessfully_ported</flow_name>
            <flow_type>publish-subscribe</flow_type>
            <sources>
                <source_service_endpoint service_id="1"/>
            </sources>
            <targets>
                <target_service_endpoint service_id="6"/>
            </targets>
            <message_subject>Customer_sucessfully_ported</message_subject>
            <message_id>004365412121212</message_id>
        </flow>
    </workflow>
    <dependencies>
        <order_dependency>
            <flow_order>
                <flow flow_id="fl1_wf1">
                    <next>
                        <flow flow_id="fl2_wf2"/>
                    </next>
                </flow>
```

```
                <flow flow_id="fl2_wf2">
                    <before>
                        <flow flow_id="fl1_wf1"/>
                    </before>
                    <next>
                        <flow flow_id="fl3_wf1"/>
                    </next>
                </flow>
                <flow flow_id="fl3_wf1">
                    <before>
                        <flow flow_id="fl2_wf2"/>
                    </before>
                </flow>
            </flow_order>
        </order_dependency>
        <time_dependency>
            <latency>
                <from_invocation_flow flow_id="fl1_wf1"/>
                <to_termination_flow flow_id="fl2_wf1"/>
                <max_seconds>10800</max_seconds>
            </latency>
            <latency>
                <from_termination_flow flow_id="fl2_wf1"/>
                <to_termination_flow flow_id="fl3_wf1"/>
                <max_seconds>7200</max_seconds>
            </latency>
        </time_dependency>
    </dependencies>
</tmss-test>
```

2.2 Explanation for Testing

tmss_test	Head element for the configuration XML file.		
	Attributes		
	test_scenario		Unique name for the test. A reference to the unique attribute “test_scenario”, which is used in the head element “tmss-test” in the configuration XML file.
	test_id		Unique id for the test. A reference to the unique attribute “test_id”, which is used in the head element “tmss-test” in the configuration XML file.

endpoints	Head element for endpoints.		
>> service_endpoint	Group element for service endpoints.		
	Attributes		
	service_id		Unique id for this service endpoint.
>>> name	Unique name for the service endpoint. This is the name that has to arise in the log file in the field “ServiceName”.		
>>> ta_agent_id	A reference to the unique agent_id of a test agent from the configuration file.		

workflow	Head element for workflows.		
	Attributes		
	wf_id		Unique id for this workflow.
> workflow_name	Unique name for the workflow.		

> *flow*	Group element for different sub flows in the workflow.	
	Attributes	
	flow_id	Unique id for this flow.
>> *flow_name*	Unique name for this flow.	
>> *flow_type*	Possible Values	
	point-to-point	If the service endpoints involved communicate point to point
	publish-subscribe	If the service endpoints communicate via a bus structure, using publish-subscribe.
>> *sources*	Group element for the service endpoints functioning as a source, which means an endpoint initiating a flow.	
>>> *source_service_endpoint*	*Attributes*	
	service_id	A reference to the unique service_id, which is used in the element "endpoints -> service_endpoint".
>>>> *message_subject*	The subject of the message. This is the subject that has to arise in the log file in the field "MessageSubject". This has just to be used if the service endpoint has a different message_subject as indicated by the element "workflow->message_subject".	
>>>> *message_id*	A unique message_id for the message. This is the id that has to arise in the log file in the field "MessagID". This has just has to be used if the service endpoint has a different message_id as indicated by the element "Workflow->message_id".	
>> *targets*	Group element for the service endpoints functioning as a target.	
>>> *target_service_endpoint*	*Attributes*	
	service_id	A reference to the unique service_id, which is used in the element "endpoints -> service_endpoint".
>>>> *message_subject*	The subject of the message. This is the subject that has to arise in the log file in the field "MessageSubject". This has just to be used if the service endpoint has a different message_subject as indicated by the element "workflow->message_subject".	
>>>> *message_id*	A unique message_id for the message. This is the id that has to arise in the log file in the field "MessagID". This has just has to be used if the service endpoint has a different message_id as indicated by the element "Workflow->message_id".	
>>> *message_subject*	The subject of the message. This is the subject that has to arise in the log file in the field "MessageSubject".	
>>> *message_id*	A unique message_id for the message. This is the id that has to arise in the log file in the field "MessagID".	
>>> *message_order*	Group element for message order. If "message_order" is not used then no specified order of the message is needed.	
>>>> *fixed_order*	Group element for the messages.	
>>>>> *message*	Group element for the message.	
>>>>>> *source_service_endpoint*	*Attributes*	
	service_id	A reference to the unique service_id, which is used in the element "endpoints -> service_endpoint".
>>>>>> *target_service_endpoint*	*Attributes*	
	service_id	A reference to the unique service_id, which is used in the element "endpoints -> service_endpoint".
>>>>>> *latency_from_source_to_target*	Group element for the latency.	
>>>>>>> *max_seconds*	An indicator of the maximum latency between sending and receiving a message from source to target. In seconds.	

dependencies	Head element for workflows.
> *order_dependency*	Group element for the order of flows.
>> *flow_order*	Group element for flow_order.

>>> flow	*Attributes* flow_id — A reference to the unique flow_id, which is used in the element “workflow->flow”.
>>>> before	Indicating which flow has to be executed before this flow. If this element is not in a “flow” element than this is the first flow.
>>>> next	Indicating which flow has to be executed after this flow. If this element is not in a “flow” element than this is the last flow.
> time_dependency	Group element for time dependencies.
>> latency	Group element for latency.
>>> from_termination_flow	This is used indicating that the timespan should be counted from the time on this flow was terminated *Attributes* flow_id — A reference to the unique flow_id, which is used in the element “workflow->flow”.
>>> to_termination_flow	This is used indicating that the timepsan should be counted until the time this flow is terminated. *Attributes* flow_id — A reference to the unique flow_id, which is used in the element “workflow->flow”.
from_invocation_flow	*This is used indicating that the timespan should be counted from the time on this flow was initiated.* *Attributes* flow_id — A reference to the unique flow_id, which is used in the element “workflow->flow”.
fo_invocation_flow	*This is used indicating that the timespan should be counted until the time this flow gets initiated.* *Attributes* flow_id — A reference to the unique flow_id, which is used in the element “workflow->flow”.
>>> max_seconds	The maximum timespan between the “from” element until the “to” element. In seconds.