

The complete interface definition of fault tolerant web service via WSDL and OMG IDL

Chen-Liang Fang³, Deron Liang^{1,2} and Fengyi Lin⁴

¹Institute of Information Science, Academia Sinica, Taipei, Taiwan, 11529, R.O.C.

²Department of Computer and Information Science, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

³Department of Information Management, Jin-Wen Institute of Technology, Taipei, Taiwan, R.O.C.

⁴Department of Accounting, Chih Lee Institute of Technology, Taipei, Taiwan, R.O.C.

Abstract

This article present the interface definitions of the fault-tolerance SOAP (or FT-SOAP) in OMG IDL and its IDL mapping to the WSDL 1.1. This article is a supplement document for a previously published document under the same topic, TR-IIS-03-013. The purpose of this article is to illustrate the mapping between the human readable FT-SOAP (in OMG IDL) and machine readable form in WSDL.

1 The FT-SOAP service implementation detail

This is a supplemental document for our previous Technical Report TR-IIS-03-013. We present the complete interface definitions that are discussed in TR- IIS-03-013. The interface definitions in OMG IDL are listed in Section 2. The WSDL format of these interfaces is listed in Section 3.

2 The FT-SOAP service interface definitions in OMG IDL format

```
module FTSOAP {
typedef int ReplicationStyleValue;
typedef int MembershipStyleValue;
typedef int ConsistencyStyleValue;
typedef int FaultMonitoringStyleValue;
typedef int FaultMonitoringGranularityValue;
typedef int InitialNumberReplicasValue;
typedef int MinimumNumberReplicasValue;
typedef int Degree;
typedef int Threshold;
typedef long CheckpointInterval;
typedef string Location;
typedef string Service; //WSDL
typedef string typeId;
typedef Service ServiceGroup; //WSG
typedef Service FaultNotifier; //WSDL
typedef sequence<Location>Locations;
struct Property {
    string name;
    string value;
}
typedef sequence<Property> Criteria; // new design
struct FactoryInfo {
    GenericFactory the_factory;
    Location the_location;
}
```

```

Criteria the_criteria;
}
typedef sequence<FactoryInfo> FactoryInfos;
typedef FactoryInfos FactoriesValue;
struct Properties {
    Property[] props;
    FactoryInfos factoryInfos ;
}
// Exception Declaration
exception InterfaceNotFound {};
exception ServiceGroupNotFound {};
exception MemberNotFound {};
exception ServiceNotFound {};
exception ServiceNotDeployed {};
exception ServiceNotAdded {};
exception UnsupportedProperty {
    Property prop;
};
exception InvalidProperty {
    Property prop;
};
exception NoFactory {
    Location the_location;
    typeId type_id;
};
exception InvalidCriteria {
    Criteria invalid_criteria;
};
exception CannotMeetCriteria {
    Criteria unmet_criteria;
};
interface PropertyManager {
    void set_default_properties(in Properties props)
        raises (InvalidProperty,UnsupportedProperty);
    Properties get_default_properties()
        raises (InvalidProperty,UnsupportedProperty);
    void remove_default_properties(in Properties props)
        raises (InvalidProperty,UnsupportedProperty);
    void set_type_properties(in typeId the_typeId, in Properties props)
        raises (InvalidProperty,UnsupportedProperty);
    Properties get_type_properties(in typeId the_typeId)
        raises (ServiceNotFound);
    void remove_type_properties(in typeId the_typeId, in Properties props)
        raises (ServiceNotFound,InvalidProperty,UnsupportedProperty);
};
interface GenericFactory {
    Service create_service(in typeId the_typeId)
        raises(ServiceGroupNotFound,ServiceNotFound,ServiceNotAdded);
    void destroy_service(in Service service)
        raises(ServiceNotFound);
    string activate_service(in typeId the_typeId)
        raises(ServiceNotFound,ServiceNotAdded);
};
interface ServiceGroupManager {
    ServiceGroup create_member(in typeId the_typeId, in Location the_location)
        raises(ServiceGroupNotFound,MemberNotFound);
    ServiceGroup add_member(in ServiceGroup srvGrp, in Service ServiceMember)
        raises(ServiceGroupNotFound);
    ServiceGroup remove_member(in ServiceGroup srvGrp, in Location the_location)
        raises(ServiceGroupNotFound,MemberNotFound);
    ServiceGroup set_promary_member(in ServiceGroup srvGrp, in Location the_location)
        raises(ServiceGroupNotFound);
    Locations locations_of_member(in ServiceGroup srvGrp)
        raises(ServiceGroupNotFound);
};
interface NS_consumer {
    void push_event(in Event faultEvent);
};
interface FaultNotifier:NS_consumer {

```

```

        void subscribe(in Service rm, in string filter);
        void unsubscribe (in Service rm, in string filter);
};
interface FaultDetector {
    void register(in Service fault_notifier, in Service detectionTarget);
};
interface PullMonitorable {
    boolean is_alive();
};
interface ReplicationManager:PropertyManager,GenericFactory,ServiceGroupManager,NS_consumer {
    void register_fault_notifier(in FaultNotifier fault_notifier);
    FaultNotifier get_fault_notifier()
        raises (InterfaceNotFound);
};
};

```

3 The FT-SOAP service interface definitions in WSDL

```

<?xml version="1.0" ?>
<definitions name="ReplicationManager" targetNamespace="http://host.domain/ReplicationManager"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://host.domain/ReplicationManager"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <types>
    <schema targetNamespace="http://host.domain/ReplicationManager.xsd"
xmlns:xsd="http://www.w3.org/wsdl/">
      <xsd:complexType name="Locations">
        <xsd:complexContent>
          <xsd:sequence>
            <xsd:element name="item" maxOccurs="unbounded" type="Location"/>
          </xsd:sequence>
        </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="MemberNotFound">
        <xsd:sequence>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:simpleType name="ServiceGroup">
        <xsd:restriction base="Service">
        </xsd:simpleType>
      <xsd:simpleType name="Location">
        <xsd:restriction base="string">
        </xsd:simpleType>
      <xsd:complexType name="ServiceNotAdded">
        <xsd:sequence>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="ServiceGroupNotFound">
        <xsd:sequence>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:simpleType name="Service">
        <xsd:restriction base="string">
        </xsd:simpleType>
      <xsd:complexType name="ServiceNotFound">
        <xsd:sequence>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:simpleType name="typeId">
        <xsd:restriction base="string">
        </xsd:simpleType>
      <xsd:complexType name="UnsupportedProperty">
        <xsd:sequence>
          <xsd:element name="prop" type="Property"/>
        </xsd:sequence>
      </xsd:complexType>
    </schema>
  </types>

```

```

<xsd:complexType name="Property">
  <xsd:sequence>
    <xsd:complexType name="name" type="string"/>
    <xsd:complexType name="value" type="string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="InvalidProperty">
  <xsd:sequence>
    <xsd:element name="prop" type="Property"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Properties">
  <xsd:complexContent>
    <xsd:sequence>
      <xsd:element name="item" maxOccurs="unbounded" type="Property"/>
    </xsd:sequence>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="InterfaceNotFound">
  <xsd:sequence>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="FaultNotifier">
  <xsd:restriction base="Service">
  </xsd:simpleType>
</schema>
</types>
<message name="register_fault_notifier">
  <part name="fault_notifier" type="xsd:FaultNotifier" />
</message>
<message name="register_fault_notifierResponse">
  <part name="_return" type="void"/>
</message>
<message name="get_fault_notifier">
</message>
<message name="get_fault_notifierResponse">
  <part name="_return" type="FaultNotifier"/>
</message>
<message name="_exception.InterfaceNotFound"/>
  <part name="exception" type="InterfaceNotFound"/>
</message>
<message name="set_default_proerties">
  <part name="props" type="Properties" />
</message>
<message name="set_default_proertiesResponse">
  <part name="_return" type="void"/>
</message>
<message name="_exception.InvalidProperty"/>
  <part name="exception" type="InvalidProperty"/>
</message>
<message name="_exception.UnsupportedProperty"/>
  <part name="exception" type="UnsupportedProperty"/>
</message>
<message name="get_default_properties">
</message>
<message name="get_default_propertiesResponse">
  <part name="_return" type="Properties"/>
</message>
<message name="_exception.InvalidProperty"/>
  <part name="exception" type="InvalidProperty"/>
</message>
<message name="_exception.UnsupportedProperty"/>
  <part name="exception" type="UnsupportedProperty"/>
</message>
<message name="remove_default_properties">
  <part name="props" type="Properties" />
</message>
<message name="remove_default_propertiesResponse">
  <part name="_return" type="void"/>

```

```

</message>
<message name="_exception.InvalidProperty"/>
  <part name="exception" type="InvalidProperty"/>
</message>
<message name="_exception.UnsupportedProperty"/>
  <part name="exception" type="UnsupportedProperty"/>
</message>
<message name="set_type_properties">
  <part name="the_typeId" type="typeId" />
  <part name="props" type="Properties" />
</message>
<message name="set_type_propertiesResponse">
  <part name="_return" type="void"/>
</message>
<message name="_exception.InvalidProperty"/>
  <part name="exception" type="InvalidProperty"/>
</message>
<message name="_exception.UnsupportedProperty"/>
  <part name="exception" type="UnsupportedProperty"/>
</message>
<message name="get_type_properties">
  <part name="the_typeId" type="typeId" />
</message>
<message name="get_type_propertiesResponse">
  <part name="_return" type="Properties"/>
</message>
<message name="_exception.ServiceNotFound"/>
  <part name="exception" type="ServiceNotFound"/>
</message>
<message name="remove_type_properties">
  <part name="the_typeId" type="typeId" />
  <part name="props" type="Properties" />
</message>
<message name="remove_type_propertiesResponse">
  <part name="_return" type="void"/>
</message>
<message name="_exception.ServiceNotFound"/>
  <part name="exception" type="ServiceNotFound"/>
</message>
<message name="_exception.InvalidProperty"/>
  <part name="exception" type="InvalidProperty"/>
</message>
<message name="_exception.UnsupportedProperty"/>
  <part name="exception" type="UnsupportedProperty"/>
</message>
<message name="create_service">
  <part name="the_typeId" type="typeId" />
</message>
<message name="create_serviceResponse">
  <part name="_return" type="Service"/>
</message>
<message name="_exception.ServiceGroupNotFound"/>
  <part name="exception" type="ServiceGroupNotFound"/>
</message>
<message name="_exception.ServiceNotFound"/>
  <part name="exception" type="ServiceNotFound"/>
</message>
<message name="_exception.ServiceNotAdded"/>
  <part name="exception" type="ServiceNotAdded"/>
</message>
<message name="destroy_service">
  <part name="service" type="Service" />
</message>
<message name="destroy_serviceResponse">
  <part name="_return" type="void"/>
</message>
<message name="_exception.ServiceNotFound"/>
  <part name="exception" type="ServiceNotFound"/>
</message>

```

```

<message name="activate_service">
  <part name="the_typeId" type="typeId" />
</message>
<message name="activate_serviceResponse">
  <part name="_return" type="string"/>
</message>
<message name="_exception.ServiceNotFound"/>
  <part name="exception" type="ServiceNotFound"/>
</message>
<message name="_exception.ServiceNotAdded"/>
  <part name="exception" type="ServiceNotAdded"/>
</message>
<message name="create_member">
  <part name="the_typeId" type="typeId" />
  <part name="the_location" type="Location" />
</message>
<message name="create_memberResponse">
  <part name="_return" type="ServiceGroup"/>
</message>
<message name="_exception.ServiceGroupNotFound"/>
  <part name="exception" type="ServiceGroupNotFound"/>
</message>
<message name="_exception.MemberNotFound"/>
  <part name="exception" type="MemberNotFound"/>
</message>
<message name="add_member">
  <part name="srvGrp" type="ServiceGroup" />
  <part name="ServiceMember" type="Service" />
</message>
<message name="add_memberResponse">
  <part name="_return" type="ServiceGroup"/>
</message>
<message name="_exception.ServiceGroupNotFound"/>
  <part name="exception" type="ServiceGroupNotFound"/>
</message>
<message name="remove_member">
  <part name="srvGrp" type="ServiceGroup" />
  <part name="the_location" type="Location" />
</message>
<message name="remove_memberResponse">
  <part name="_return" type="ServiceGroup"/>
</message>
<message name="_exception.ServiceGroupNotFound"/>
  <part name="exception" type="ServiceGroupNotFound"/>
</message>
<message name="_exception.MemberNotFound"/>
  <part name="exception" type="MemberNotFound"/>
</message>
<message name="set_promary_member">
  <part name="srvGrp" type="ServiceGroup" />
  <part name="the_location" type="Location" />
</message>
<message name="set_promary_memberResponse">
  <part name="_return" type="ServiceGroup"/>
</message>
<message name="_exception.ServiceGroupNotFound"/>
  <part name="exception" type="ServiceGroupNotFound"/>
</message>
<message name="locations_of_member">
  <part name="srvGrp" type="ServiceGroup" />
</message>
<message name="locations_of_memberResponse">
  <part name="_return" type="Locations"/>
</message>
<message name="_exception.ServiceGroupNotFound"/>
  <part name="exception" type="ServiceGroupNotFound"/>
</message>
<message name="push_event">
  <part name="faultEvent" type="Event" />

```

```

</message>
<message name="push_eventResponse">
  <part name="_return" type="void"/>
</message>
<portType name="ReplicationManagerPortType">
  <operation name="register_fault_notifier">
    <input message="tns:register_fault_notifier" />
  </operation>
  <operation name="get_fault_notifier">
    <output message="tns:get_fault_notifierResponse" />
    <fault message name="_exception.InterfaceNotFound"/>
  </operation>
  <operation name="set_default_proerties">
    <input message="tns:set_default_proerties" />
    <fault message name="_exception.InvalidProperty"/>
    <fault message name="_exception.UnsupportedProperty"/>
  </operation>
  <operation name="get_default_properties">
    <output message="tns:get_default_propertiesResponse" />
    <fault message name="_exception.InvalidProperty"/>
    <fault message name="_exception.UnsupportedProperty"/>
  </operation>
  <operation name="remove_default_properties">
    <input message="tns:remove_default_properties" />
    <fault message name="_exception.InvalidProperty"/>
    <fault message name="_exception.UnsupportedProperty"/>
  </operation>
  <operation name="set_type_properties">
    <input message="tns:set_type_properties" />
    <fault message name="_exception.InvalidProperty"/>
    <fault message name="_exception.UnsupportedProperty"/>
  </operation>
  <operation name="get_type_properties">
    <input message="tns:get_type_properties" />
    <output message="tns:get_type_propertiesResponse" />
    <fault message name="_exception.ServiceNotFound"/>
  </operation>
  <operation name="remove_type_properties">
    <input message="tns:remove_type_properties" />
    <fault message name="_exception.ServiceNotFound"/>
    <fault message name="_exception.InvalidProperty"/>
    <fault message name="_exception.UnsupportedProperty"/>
  </operation>
  <operation name="create_service">
    <input message="tns:create_service" />
    <output message="tns:create_serviceResponse" />
    <fault message name="_exception.ServiceGroupNotFound"/>
    <fault message name="_exception.ServiceNotFound"/>
    <fault message name="_exception.ServiceNotAdded"/>
  </operation>
  <operation name="destroy_service">
    <input message="tns:destroy_service" />
    <fault message name="_exception.ServiceNotFound"/>
  </operation>
  <operation name="activate_service">
    <input message="tns:activate_service" />
    <output message="tns:activate_serviceResponse" />
    <fault message name="_exception.ServiceNotFound"/>
    <fault message name="_exception.ServiceNotAdded"/>
  </operation>
  <operation name="create_member">
    <input message="tns:create_member" />
    <output message="tns:create_memberResponse" />
    <fault message name="_exception.ServiceGroupNotFound"/>
    <fault message name="_exception.MemberNotFound"/>
  </operation>
  <operation name="add_member">
    <input message="tns:add_member" />
    <output message="tns:add_memberResponse" />
  </operation>

```

```

<fault message name="_exception.ServiceGroupNotFound"/>
</operation>
<operation name="remove_member">
<input message="tns:remove_member" />
<output message="tns:remove_memberResponse" />
<fault message name="_exception.ServiceGroupNotFound"/>
<fault message name="_exception.MemberNotFound"/>
</operation>
<operation name="set_promary_member">
<input message="tns:set_promary_member" />
<output message="tns:set_promary_memberResponse" />
<fault message name="_exception.ServiceGroupNotFound"/>
</operation>
<operation name="locations_of_member">
<input message="tns:locations_of_member" />
<output message="tns:locations_of_memberResponse" />
<fault message name="_exception.ServiceGroupNotFound"/>
</operation>
<operation name="push_event">
<input message="tns:push_event" />
</operation>
</portType>
<binding name="ReplicationManagerSoapBinding" type="ReplicationManagerPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="register_fault_notifier">
<soap:operation soapAction="http://host.domain/register_fault_notifier"/>
<input>
<soap:body use="encoded" namespace="http://host.domainReplicationManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
</operation>
<operation name="get_fault_notifier">
<soap:operation soapAction="http://host.domain/get_fault_notifier"/>
<output>
<soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="set_default_proerties">
<soap:operation soapAction="http://host.domain/set_default_proerties"/>
<input>
<soap:body use="encoded" namespace="http://host.domainPropertyManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
</operation>
<operation name="get_default_properties">
<soap:operation soapAction="http://host.domain/get_default_properties"/>
<output>
<soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="remove_default_properties">
<soap:operation soapAction="http://host.domain/remove_default_properties"/>
<input>
<soap:body use="encoded" namespace="http://host.domainPropertyManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
</operation>
<operation name="set_type_properties">
<soap:operation soapAction="http://host.domain/set_type_properties"/>
<input>
<soap:body use="encoded" namespace="http://host.domainPropertyManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
</operation>
<operation name="get_type_properties">
<soap:operation soapAction="http://host.domain/get_type_properties"/>
<input>

```



```

    <soap:body use="encoded" namespace="http://host.domainPropertyManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="remove_type_properties">
  <soap:operation soapAction="http://host.domain/remove_type_properties"/>
  <input>
    <soap:body use="encoded" namespace="http://host.domainPropertyManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
</operation>
<operation name="create_service">
  <soap:operation soapAction="http://host.domain/create_service"/>
  <input>
    <soap:body use="encoded" namespace="http://host.domainGenericFactory"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="destroy_service">
  <soap:operation soapAction="http://host.domain/destroy_service"/>
  <input>
    <soap:body use="encoded" namespace="http://host.domainGenericFactory"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
</operation>
<operation name="activate_service">
  <soap:operation soapAction="http://host.domain/activate_service"/>
  <input>
    <soap:body use="encoded" namespace="http://host.domainGenericFactory"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="create_member">
  <soap:operation soapAction="http://host.domain/create_member"/>
  <input>
    <soap:body use="encoded" namespace="http://host.domainServiceGroupManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="add_member">
  <soap:operation soapAction="http://host.domain/add_member"/>
  <input>
    <soap:body use="encoded" namespace="http://host.domainServiceGroupManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="remove_member">
  <soap:operation soapAction="http://host.domain/remove_member"/>
  <input>

```

```

    <soap:body use="encoded" namespace="http://host.domainServiceGroupManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="set_promary_member">
  <soap:operation soapAction="http://host.domain/set_promary_member"/>
  <input>
    <soap:body use="encoded" namespace="http://host.domainServiceGroupManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="locations_of_member">
  <soap:operation soapAction="http://host.domain/locations_of_member"/>
  <input>
    <soap:body use="encoded" namespace="http://host.domainServiceGroupManager"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="http://host.domain"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="push_event">
  <soap:operation soapAction="http://host.domain/push_event"/>
  <input>
    <soap:body use="encoded" namespace="http://host.domainNS_consumer"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
</operation>
</binding>
<service name="ReplicationManagerService">
  <port name="ReplicationManagerPort" binding="ReplicationManagerBinding"
location="http://host.domain/ReplicationManager"/>
  </port>
</service>
</definitions>

```