

OPEI Warranty Web Service Specification

Prepared By

The OPEI e-Commerce Committee

April 30, 2008

With special appreciation to:

The Ariens Company and its business partner Skyline Technologies, Inc.
For providing the initial proposals and documentation upon
which the Work group developed this consensus specification.

TABLE OF CONTENTS

Topic	Page
Introduction	4
OPEI Web Services Standards Overview	4
Base Standards	5
Discovery & Transaction Standard Sets	6
The OPEI Standardized Web Service Specification	8
The Standardized Web Services	8
The Web Service Binding	8
The Web Service Operation	8
Message Schema	8
Versioning	9
Namespaces and Document Access	9
The WSDL Specification	10
Master WSDL Specification	10
Definitions Section	12
Types Section	12
Messages Section	13
Port Type and Operations Section	13
Binding Section	14

TABLE OF CONTENTS

Topic	Page
Discovery	14
Initial Endpoint	14
The Discovery Web Service	17
Discovery Submit Schema	19
Discovery Response Schema	20
Business Rules	23
Standard Business Rule Types	23
Business Rule Application	26
Handling Errors	28
Faults	29
Special Situations	30
Response Possibilities	32
WS-I Conformance	32

Introduction

This document contains the Outdoor Power Equipment Institute specification for the implementation of Web Service Standards. This standards program was developed as part of a master strategy to enable the use of “inter-operable” web services in the Outdoor Power Equipment industry. In this context, “inter-operable” means that any user complying with the OPEI web services standards may readily employ web services with any other user also in compliance. The intended users are dealers, distributors, and manufacturers, the latter two being referred to as “suppliers”. A critical feature of this standards program is that dealer participation requires representation by the dealer business system vendors (BSV’s), since web services capability must be programmed directly into dealer business systems. For this purpose, leading industry BSV’s have been invited to fully participate in the standards development.

The OPEI strategy is based on the use of Web Services technology in a tightly controlled manner with industry standards that provide approved specifications and guidance to users. The strategy calls for managed change of the standard specifications to allow more robust functionality over time as industry users indicate readiness and capability for advancement. Even though the tight standards may limit desired progress among more advanced users, by staying within these standard specifications, all users can be assured of inter-operability. It is projected that eventually, the full capability of Web Services and related technologies will be employed within the industry standard specifications.

OPEI Web Service Standards Overview

The nature of the “Tight” standards approach adopted by OPEI is to keep implementation as simple and cost effective as practical in the early stages of implementation, and then to add complexity and functionality over time in response to general readiness of the industry as a whole. In concert with this approach, the standards allow only one form of web services to be used by standardizing the binding for all web services within the WSDL specification. This ensures simplicity for all, even though it restricts the manner in which web services may be accomplished by sophisticated users. In the same manner, only certain business rule types have been standardized and allowed, and this too may restrict some practices desired by certain users. This strategy presumes that users will over time, petition the e-Commerce Committee for changes that will provide more freedom and flexibility. Such changes may then be granted once the committee has determined they will not disadvantage other industry users due to cost or technical expertise. The objective of course is to maintain orderly and effective growth of technical capability while minimizing costs for all participants.

The components of the OPEI Web Service Standards are as follows:

Base Standards

W3C XML & Web Services Specifications

The entire inventory of the World Wide Web Consortium (W3C), the official standards body for Extensible Markup Language (XML) and Web Services has been consulted during the development of the OPEI standards. The XML language, Web Services Description Language (WSDL), and the Simple Object Access Protocol (SOAP) standards have been adopted by the OPEI. In addition, other Web Services Standards work such as WS-Transfer, WS-MetadataExchange, and Universal Description Discovery and Integration (UDDI) were consulted and allowed to influence the project.

This Specification Document

This document provides background understanding and the specific requirements for implementation of the OPEI Web Services Standards. It is subject to review and revision as the project progresses and new standards are developed.

The Standard Web Services Description Language (WSDL) file which defines the web services.

The exact definition of all of the approved web services is contained within this standardized document. It specifies that all OPEI Web Services are structured exactly the same way, and the only variance between them is the messages or schemas they employ. The OPEI standard WSDL File is presented below, and the actual WSDL files for each standard web service may be obtained from www.opeiwebservices.org the official OPEI standards web site.

Business Rules Specification and Schema.

In order to accommodate the various business rules and techniques employed by suppliers, a unique approach to standardizing business rules has been employed. This approach was created by the Motorcycle Industry Council in its Power Sports Protocol standards, and is gratefully adopted with permission from PSP. It is based on the premise that only suppliers may institute business rules, and it specifies a set of standard business rule types which may be employed to web service transactions.

Master Schema (reusable data elements).

OPEI has not as yet adopted a general message standard due to significant confusion in the XML standards world. Instead, it has developed its own XML message structure employing reusable data elements and groups. All of these are incorporated within the OPEI Master Schema from which all other schema inherit the basic OPEI characteristics. It is the intention of the e-Commerce Committee to monitor XML standards evolution and at an appropriate time adopt a national or international XML standard language that has attained wide multi-industry acceptance.

Data Code Lists

Historically, much of the data exchanged by dealers and suppliers has been codified for convenience and brevity. The OPEI has adopted code lists of several kinds and many are mandatory within the OPEI web services messages (schemas). The employment of business rules may make the use of some code lists unnecessary, but they will be maintained and available as long as necessary.

Discovery & Transaction Standard Sets

The following OPEI standards are specific to a given standardized web service, and may only be employed within that type of web service.

Submit and Response Schemas (messages).

Each OPEI standardized web service employs a “submit message” which may only be submitted from a customer to a supplier, and must conform to the OPEI schema standard for that particular web service. Also, a standard “response message” is required from the supplier back to the originating customer advising disposition of the original submit message. The schemas for all OPEI Web Service Standards are available at www.opeiwebservices.org/schemas .

Optional Business Rule Templates.

Standard templates can be created for each web service “submit message” schema defining what business rule types may be applied to specific data elements within the submit message for a given web service. Suppliers may optionally use these templates to specify requirements such as restricting the data allowed in a field, or to identify specific options for payment, transport, etc. These rules in effect, create the ability to customize each web service transaction in accordance with a supplier’s requirements for any given dealer customer. They further accomplish this customization without requiring any programming by the dealer business system vendors that is unique to a given supplier. At the time of this printing, a Business Rule standard has been provided for only the Purchase Order submission.

Summary

The use of all of these standards and specifications as published upon industry approval, are mandatory for compliance with the OPEI Warranty Web Service standard for the industry. Dealers and distributors will be dependent on the strict adherence to these specifications by manufacturers hosting an OPEI Warranty Web Service and by business system developers who provide client capabilities for use of the OPEI Standard Web Services.

The OPEI Standardized Web Service Specification

The Standardized Web Services

OPEI, and this specification do not support any web services other than the OPEI Standardized Web Services which may vary in number from time to time, but will always be identified as OPEI Standardized Web Services. Each OPEI Standardized Web Service is officially named and provided a Version Number of 1 or greater and a Variant number which may be zero or greater. All version and variants follow strict sequencing, meaning Version 3 immediately follows Version 2, and so on. Full information regarding all Standardized Web Services and supporting standards is available from www.opeiwebservices.org , the official OPEI Web Services web site and repository.

The Web Service Binding

The OPEI Web Services Standard named “opei-Transport” specifies that **SOAP over HTTP** will be the **ONLY** protocol used to carry messages and that these messages will use the **document** message style along with **literal** encoding (i.e., document/literal). This is also a best practice recommended by the Web Services Interoperability Group.

The Web Service Operation

OPEI specifies **ONLY** one operation that is called the “ProcessMessage Operation”, and is defined as requiring an input “SubmitMessage” and an output “ResponseMessage”.

Message Schema

Each Web Service employs unique “Submit” and “Response” messages in the form of OPEI schemas. Use of the appropriate schema is mandatory. All schema are based on the OPEI XML language and inherit the characteristics of the OPEI Master Schema for purposes of re-usability of schema components. The Master schema and all web service schemas are available at www.opeiwebservices.org .

Versioning

In accordance with the master Web Services strategy, the OPEI will change its schemas, operations, messages, or port types to keep pace with the changing business environment. As a result, all standard documents need to be versioned. Each OPEI Standard Web Service is identified by a version number and variant as appropriate, (i.e. Ver. 2.3 = version 2, variant 3.) Variants are used for minor changes that should not affect implementation, while version numbers are used for revisions of significance. All components of the standards program are coordinated by version number, meaning that Version 2 of a web service will require that the WSDL file, business rules and schemas for that web service all be identified and re-issued as Version 2 components. The full name of the document will then include the version as shown in the example below. Users should expect to use this full name to guard against confusion between versions of the documents.

Example: **WarrantyWebService Ver 1.1**

Although XML and Web Services standards are designed to accommodate multiple user variations of the same documents, these specifications do not. Any variance from these specifications is non-compliance with the OPEI Web Service Standards. Thus, there is no need in this specification to accommodate version control and identification at the user level.

Namespaces and Document Access

A namespace protocol has been adopted for all OPEI web service specifications and schemas. The first part of the namespace “<http://www.opeiwebservices.org>” uniquely identifies the markup as belonging to the OPEI. The next part of the namespace, “*documentName*” specifies the name of the web service specification or schema and includes the version number and variant.

Example: **<http://www.opeiwebservices.org/WarrantyWebService> Ver 1.1**

The OPEI web site “www.opeiwebservices.org” will provide access to all current and past documents that compose this specification. As changes are made, these files will be updated. The WSDL files include “import” references for these documents. It should be made clear that the intended response to this import instruction is to download the standard documents from the OPEI web site when creating the web service application. Once the latest documents have been built into the web service there is no longer a need to import these files until they change. Web service hosts need not and should not provide independent access to OPEI standard documents.

The WSDL Specification

The Specification for the OPEI Web Services Definition (WSDL) is strictly within the control of the OPEI e-Commerce Committee and must be used as specified for a user to be in compliance with the standard. This WSDL file will be maintained by the e-Commerce Committee as the means to standardizing the industry's web service specification.

The OPEI Web Services are defined in total by the following Web Services Description Language (WSDL) specification. The WSDL specification is tightly specified to limit choices and to ensure simplicity. Only one form of web service is allowed at present, but it is anticipated that as industry experience with web services mounts, and complexities become more manageable, a more open specification will be appropriate.

Both the Abstract and Concrete Binding portions of the WSDL are combined, and a single WSDL definition applies to all OPEI Standardized web Services, with the exception of the web service name and the messages used within the web service. For this reason, the Master WSDL is shown and explained here, while the specific WSDL files for each Standardized Web Service are available at www.opeiwebservices.org. The only difference between the WSDL files for each web service is the schemas used for the submit and response messages. In order to accomplish this simplicity, the committee decided to leave the web address (URL or the namespace for a user's web service) out of the WSDL. Web service URL's are to be obtained through the "Discovery" technique specified below.

Master WSDL Specification

The following Master WSDL specification defines the structure of all web services standardized by OPEI. The header column on the left is for section identification and location purposes only and is not part of the WSDL file itself.

	<?xml version="1.0" encoding="utf-8" ?>
Definitions	<definitions
Start	wsdl:service name="***Insert Standard Name of Web Service***"
	targetNamespace="http://www.opeiwebservices.org/bindings"
	xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
	xmlns:xsd="http://www.w3.org/2001/XMLSchema"
	xmlns:tns="http://www.opeiwebservices.org/bindings"
	xmlns:opeiws="http://www.opeiwebservices.org/"
	xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" >

Types

```
<wsdl:types>
  <xsd:schema targetNamespace="http://www.opeiwebservices.org/schemas"
    elementFormDefault="qualified">

    <xsd:schemaLocation="http://www.opeiwebservices.org/schemas" />

    <xsd:import namespace="http://www.opeiwebservices.org/Schemas/**Insert Submit Schema Name**.xsd"
    <xsd:import namespace="http://www.opeiwebservices.org/Schemas/**Insert Response Schema Name**.xsd"

    </xsd:schema>
  </wsdl:types>
```

Messages

```
<wsdl:message name="SubmitMessage">
  <wsdl:part element="**Insert Submit Schema Root Element Name**" />
</wsdl:message>

<wsdl:message name="ResponseMessage">
  <wsdl:part element="**Insert Response Schema Root Element Name**" />
</wsdl:message>
```

Port Type and Operations

```
<wsdl:portType name="opeiTransportPortTypes">
  <operation name="ProcessMessage">
    <input message="tns:SubmitMessage" />
    <output message="tns:ResponseMessage" />
  </operation>
</wsdl:portType>
```

Binding

```
<wsdl:binding name="opeiTransport" type="tns:opeiTransportPortTypes">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <operation name="ProcessMessage">
    <soap:operation soapAction="http://www.opeiwebservices.org/bindings/ProcessMessage" style="document" />
```

```
<input>
  <soap:body use="literal" />
</input>
<output>
  <soap:body use="literal" />
</output>
</operation>
</wsdl:binding>
```

End </definitions>

Definitions Section

This section cites definitions important to information within the WSDL file.

wsdl:Service Name: This line cites the name of the web service defined by the WSDL file. The name inserted here **MUST** be the Standard Name which includes Version Number assigned by OPEI to the web service being used.

TargetNamespace: This is the namespace for retrieval of the specific WSDL for the named web service.

Standard Definitions: The remaining definitions cite the specific standards employed as foundation for the OPEI Standardized Web Services, including the locations for OPEI's standards, the SOAP, and W3C (XML) standards.

Types Section

Before any communication can take place between the client and web service, the structure must be clearly understood by both. In other words, both the client and web service must understand what a "Submit Message" looks like. The responsibility for defining the format of this and other messages falls to the *types* section of the WSDL document. The *types section* relies upon the XML Schema Definitions (XSD) to define message formats which are then referenced in the individual *message sections* with one section for each message. This section defines the "submit" and "response" message types (schemas) used within operations in the web service.

xsd:schemaLocation: Defines the location for the OPEI schemas to be used in this web service.

xsd:import namespace: Defines the namespace for the schema to be used within the web service. This namespace **MUST** be exactly what is specified in the OPEI WSDL for a given web service type and version.

Messages Section

The message schemas are compliant with the W3C XML Standard structure and employ individual fields which have specific “types” such as string, integer, or boolean. For example, *WarrantyClaimSubmitRequest* defines an input message as consisting of a warranty claim schema which, in turn consists of structures for Dealer, Reference, Owner, Equipment, and Claim Detail etc. Within the Reference structure the ClaimId field is of type “string” while in the Owner structure the DateProductFailed field is of type “date.”

Each schema is composed of *parts*, the *Root Element* specifies the underlying message type. Using SOAP over HTTP in compliance with the WS-I specification dictates that the OPEI WSDL document have a single *part* for each message. The root element of the "Submit" schema is the referenced part for the submit message, and the root element of the response schema is the referenced part for the "Response" message. The schemas **MUST** be bound to the message name with the exact title of the Root Element (part element reference) within the schema since this supports invoking the correct schema. In this manner the proper schemas are bound to the corresponding web service.

Port Type and Operations Section

A collection of operations constitutes a WSDL *port type* in version 1.1 of the WSDL specification. The term “port type” has led to much confusion, prompting version 1.2 of the WSDL specification to adopt the more descriptive term “interface”. This section identifies the Port Types available and the operations used by the web service. OPEI allows only one Port Type; "opeiTransportPortTypes", and only one operation.

operation name="ProcessMessage": This operation is specifically defined as requiring an input to the web service of a "SubmitMessage", and an output from the web service of a "ResponseMessage".

Binding Section

A port type/interface needs to be *bound* to specific network protocols and a URL in order to be realized. This is accomplished in the *binding section* of a WSDL document. In this section, the OPEI WSDL specifies that SOAP over HTTP will be used to carry messages and that these messages will use the *document* message style along with *literal* encoding (i.e., document/literal). OPEI has allowed only one binding; "SOAP over HTTP" **MUST** be used with a document/literal style of message for all Standardized Web Services.

soap:operation soapAction: The SOAP action is identical for all OPEI web services; that of the ProcessMessage operation defined above.

Discovery

The only standardized OPEI web service that is mandatory is the Discovery Web Service. Each Web Service host is responsible for establishing a Discovery Web Service through which the addresses and specifications for other web services will be discoverable. The namespace or address (also called an "endpoint") for Discovery Web Services is made available to dealer business systems in an automated fashion. This information falls into the class of "Metadata". This Metadata for a host's web service includes which web services are supported including their version numbers, the business rules for each web service, and the endpoint for each.

Any of a supplier's metadata is subject to change; the web services supported may change with new versions of the standards; business rules may change based on company policy, and even endpoints may change from time to time. All of this information has to be instantly accessible to the dealers. For this purpose, OPEI is requiring that each web service host also provide an OPEI Standard Discovery Web Service. Dealer business systems can then be programmed to automatically use this Web Service periodically to check for and obtain any changed metadata.

Initial Endpoint

In order to locate a starting point for obtaining a suppliers metadata, the address or endpoint of each supplier's Discovery Web Service will be made available via download from the OPEI Web Services web site at www.opeiwebservices.org/endpoints/endpoint file. The Endpoint file is a simple XML file which lists the web service hosts and their Discovery Web Service URLs. It will be updated as required by OPEI based on information received from suppliers hosting OPEI Standardized Web Services. It is to be downloaded by a dealer's business system as desired, and used to update business system tables of Discovery Web Service addresses.

The Endpoint File Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This schema is for providing users with the Discovery URL's of suppliers that host OPEI standardized web services. It is to
be obtained by downloading from the OPEI web Services web site.-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.opeiwebservices.org/Schemas/EndpointFile"
  targetNamespace="http://www.opeiwebservices.org/Schemas/EndpointFile"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xsd:element name="EndpointFile">
    <xsd:annotation>
      <xsd:documentation>This is the Root element for this schema.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="HostInfo" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>This element group is to be repeated for each host on record as providing OPEI standard web
services.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:sequence>
          <xsd:element name="Name" type="xsd:string">
            <xsd:annotation>
              <xsd:documentation>This element is for the company name of the host as provided by the host.</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="DiscoveryEndpoint" type="xsd:anyURI">
            <xsd:annotation>
              <xsd:documentation>This element is for the URL of the host's Discovery Web Service.</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
```

```
</xsd:element>
</xsd:sequence>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Endpoint File Example

In an instance of the Endpoint File schema, the XML HostInfo element group is repeated for each supplier hosting web services. An instance of the file might look as follows:

```
<HostInfo>
  <Name>Ariens Company</Name >
  <DiscoveryEndpoint>www.ariens.com/metadataexchange</DiscoveryEndpoint>
  <Name>MTD Corporation</Name>
  <DiscoveryEndpoint>www.mtdproducts.com/discovery/</DiscoveryEndpoint>
  <Name>Briggs & Stratton</Name>
  <DiscoveryEndpoint>www.briggsandstratton.com/mdx/</DiscoveryEndpoint>
</HostInfo>
```

This specification does not include requirements for the host's name and Discovery Web service URL. Users may provide any URL for the location of that web service, and their company name will be recorded as submitted. It is recommended however that efforts be made to limit changes to the Discovery Web Service endpoint since it is the host's responsibility to advise OPEI of changes in a timely manner in order to ensure proper updating of the Endpoint File.

The Discovery Web Service

The Discovery Web Service is an OPEI Standardized Web Service using the standard WSDL and schemas. It is the only mandatory standardized web service. In order to employ any of the business function standardized web services, a host must provide the Discovery Web Service. It is also the simplest of the web services since it will receive a very short "Submit" schema ("DiscoveryRequest") and then send a simple "Response" schema ("DiscoveryResponse") which will include the appropriate metadata. No back end processing should be required to support this web service.

Discovery Web Service WSDL File

```
<?xml version="1.0" encoding="utf-8" ?>

<definitions
  wsdl:service name="discoverywebservice ver 1.0"
  targetNamespace="http://www.opeiwebservices.org/bindings"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.opeiwebservices.org/bindings"
  xmlns:opeiws="http://www.opeiwebservices.org/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" >

  <wsdl:types>
    <xsd:schema targetNamespace="http://www.opeiwebservices.org/schemas"
      elementFormDefault="qualified">

      <xsd:schemaLocation="http://www.opeiwebservices.org/schemas" />

      <xsd:import namespace="http://www.opeiwebservices.org/Schemas/DiscoverySubmit ver 1.0.xsd"
        <xsd:import namespace="http://www.opeiwebservices.org/Schemas/DiscoveryResponse ver 1.0.xsd"

    </xsd:schema>
  </wsdl:types>
```

```
<wsdl:message name="SubmitMessage">
  <wsdl:part element="DiscoverySubmit" />
</wsdl:message>

<wsdl:message name="ResponseMessage">
  <wsdl:part element="DiscoveryResponse" />
</wsdl:message>

  <wsdl:portType name="opeiTransportPortTypes">
    <operation name="ProcessMessage">
      <input message="tns:SubmitMessage" />
      <output message="tns:ResponseMessage" />
    </operation>
  </wsdl:portType>

  <wsdl:binding name="opeiTransport" type="tns:opeiTransportPortTypes">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
    <operation name="ProcessMessage">
      <soap:operation soapAction="http://www.opeiwebservices.org/bindings/ProcessMessage" style="document" />
      <input>
        <soap:body use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
      </output>
    </operation>
  </wsdl:binding>
</definitions>
```

Discovery Submit Schema

As stated above, the schemas used in the discovery web service are very simple. Below is the DiscoverySubmit schema which does nothing more than identify the submitting party and trigger a response from the supplier's service. This approach allows the supplier to control access to its discovery web service based on supplier assigned ID of the dealer.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.opeiwebservices.org/Schemas/DiscoverySubmit"
  targetNamespace="http://www.opeiwebservices.org/Schemas/DiscoverySubmit"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xsd:element name="DiscoverySubmit">
    <xsd:annotation>
      <xsd:documentation>This is the Root element for this schema.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="SubmitterParty" type="tns:SubmitterType" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Only the mandatory data elements are to be used in this schema.</xsd:documentation>
          </xsd:annotation>
          <xsd:sequence>
            <xsd:element name="SubmitterID" type="xsd:string">
              <xsd:annotation>
                <xsd:documentation>This element is for the Id code assigned to the submitter by the web service host or a
commonly recognized party.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="SubmitterName" type="xsd:string">
              <xsd:annotation>
                <xsd:documentation>This element is for the name of an individual responsible for submitting the claim.
```

```
        </xsd:documentation>
      </xsd:annotation>
    <xsd:sequence>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Discovery Response Schema

The Discovery Response schema is designed to transport all critical information about the web services hosted by the supplier queried. This information is contained in an element group called "Discovery", and it provides mandatory fields for the type of web service, the URL address of that service, and the expiration date for the service. It also includes optional fields for the namespace for the applicable business rules document if one exists, and a start date if the service is planned but not yet available.

Where business rules are available and the supplier has customized them for its own use in a given web service, the namespace for the applicable business rules document is to be provided within the "BusinessRulesDoc" data element. If there are no business rules applicable, this field will not be used by the supplier. This design anticipates that dealers will not want to receive all business rule documents automatically with every discovery inquiry. However, receiving the namespaces for all business rule documents will allow dealers to download those documents they intend to use.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This schema is for use by an OPEI Discovery Web Service only. Upon receipt of a DiscoverySubmit schema the web service
immediately issues this schema containing the desired discovery information.-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.opeiwebservices.org/Schemas/DiscoveryResponse"
  targetNamespace="http://www.opeiwebservices.org/Schemas/DiscoveryResponse"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
```

```

<xsd:element name="DiscoveryResponse">
  <xsd:annotation>
    <xsd:documentation>This is the Root Element for this schema.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Discovery" maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation>This element group is for the requested discovery data. This group of element must be repeated for each
web service hosted, other than the Discovery Web Service which uses this schema.</xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Service" type="xsd:string">
              <xsd:annotation>
                <xsd:documentation>This element is for the name of the web service for which information is being provided. The OPEI
standard name for the web service must be used.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="Endpoint" type="xsd:anyURI">
              <xsd:annotation>
                <xsd:documentation>This element is for the URL or endpoint address required to consume the web service named
above.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="Expiration" type="xsd:date">
              <xsd:annotation>
                <xsd:documentation>This element is for the date on which validity of the above URL expires. If there is no planned
expiration, the host must insert a date which guarantees the URL's validity to at least that date.</xsd:documentation>
                <xsd:documentation>The purpose of providing a date is to reassure user's that an endpoint will be vaild for a guaranteed
time during which they will have to resubmit a request to confirm further validity or obtain the new endpoint.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```

    </xsd:annotation>
  </xsd:element>
  <xsd:element name="BusinessRulesDoc" type="xsd:string" minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>This element is for the namespace of the business rules document for this web service if one exists. It
is the responsibility of the web service host to provide the namespace here whereby the dealer can download the applicable business
rules document.</xsd:documentation>
      <xsd:documentation>Care must be taken to ensure that this namespace is for the business rules document applicable. If
there is no business rules document applicable, this field will not be used.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="StartDate" type="xsd:date" minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>This optional element is for specifying an effective date for a web service that is
pending.</xsd:documentation>
      <xsd:documentation>If a host has set up another web site based on new Version of an OPEI standard web service, or is
simply changing the URL for a web service, this field can be used to notify users when the URL will be activated and available for
use.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

Business Rules

OPEI has borrowed business rule definitions and technique from the Power Sports industry's PSP standards. This cross industry support is greatly appreciated. At the time of this publication, business rules have been specified for the Purchase Order only. In the future, rules for other transactions will be developed when requested.

OPEI defines five business rule types that are detailed below. The data structure for each business rule type is shown here and specified within the Business Rules Base Schema. Each rule in the Base Schema is defined as an XML Datatype for purposes of controlled re-use. This tool is for use by programmers to accommodate the requirements of each rule type, and is available at:

[opeiwebservices.org/business rules/BusinessRulesBase.xsd](http://opeiwebservices.org/business%20rules/BusinessRulesBase.xsd).

Standard Business Rule Types

OptionalityRule Type

This rule allows the supplier to designate whether it wishes to ignore, accept or require an optional field in the schema. By selecting either "ignored", "accepted", or "required", the supplier indicates its expectations. If "ignored" is indicated, the dealer's business system can suppress the field and not use it in the PO. The example below shows how a business rule field named RequestedShipDate

```
<RequestedShipDate>  
  <Ignored/>  
</RequestedShipDate>
```

Table Type

Table fields are used to indicate a restricted list of values that may be used in that field. A KeyedText element contains two child elements, a key and text. The text may be used in a drop down box in the dealer's business system. The following example shows how a Table Type rule might be applied to an OrderType element.

```

<OrderType>
  <KeyedText>
    <Key>STK</Key>
    <Text>Regular Stock Order</Text>
  </KeyedText>
  <KeyedText>
    <Key>WCL</Key>
    <Text>Will Call</Text>
  </KeyedText>
</OrderType>

```

IgnorableTable Type

This type of ruled indicates the selected element will be either ignored or a table of restricted values must be used. If Ignored is indicated, the dealer business system should not use the field in the PO. If a table is indicated, then only those values may be used in this element. The following example shows two elements subject to the IgnorableTable Rule type.

```

<AllowPartialShip>
  <KeyedText>
    <Key>No Partial Ship</Key>
    <Text>If the entire requested quantity is not available, cancel the order line </Text>
  </KeyedText>
  <KeyedText>
    <Key>No Back Order </Key>
    <Text>Ship only the available quantity and cancel the remaining quantity</Text>
  </KeyedText>
</AllowPartialShip>
<LineLevelAllowPartialShip>
  <Ignored />
</LineLevelAllowPartialShip>

```

IgnoredAcceptedTable Type

This rule type is used for multiple options; the supplier may choose to ignore the data element, or may indicate it will accept any value within the element, or it may provide a restricted list of acceptable values for the element. In addition, the supplier may provide a list of proposed values for the field but indicate willingness to accept any value used by the dealer. This rule type contains child elements of **Ignored**, **AcceptedOnly**, a set of **KeyedText** elements, or the sequence of **AcceptedOrKeyedText** followed by a keyed text list of values.

The following example shows this rule applied to three elements. The first **BillTo**, is "**ignored**" indicating the supplier will ignore any data in that field. The second, **ShipTo**, contains **KeyedText** which indicates the supplier will only accept one of a restricted list of values in this field. The third element **CreditCard**, contains the sequence **AcceptedOrKeyedText** which indicates a choice by the supplier to accept an arbitrary value in the field or a selection from a proposed list of values.

```
<BillTo>
  <Ignored />
</ BillTo>
<ShipTo>
  <KeyedText>
    <Key>1</Key>
    <Text>Dealership </Text>
  </KeyedText>
  <KeyedText>
    <Key>2 </Key>
    <Text>Off-site Wharehouse</Text>
  </KeyedText>
</ ShipTo>
<CreditCard>
  <AcceptedOrKeyedText>
    <KeyedText>
      <Key>1</Key>
      <Text>Dealer Amex credit card </Text>
    </KeyedText>
    <KeyedText>
```

```
<Key>2 </Key>
<Text>Dealer VISA credit card</Text>
</KeyedText>
</ CreditCard>
```

LimitedField Type

The limited field rule supports assigning an upper limit to a value associated with the field. It may contain the child element **Unlimited** or the child element **Limit** with a positive numeric value. Where Unlimited is used there is no upper limit. If Limit is used, the numeric value assigned represents the highest value which can be present in the field. This rule can limit more than just the value of a field however. It may also be used to indicate a limit on the number of occurrences of a given field. The following example shows both usages. The first, **OrderLines** shows that occurrences of this field in the same schema instance is to be limited to 99 order lines. The second indicates there is no limit on the quantity that may appear in the **Quantity** field of an order.

Business Rule Application

For a given transaction such as the Purchase Order, the data elements or fields to which a business rule may be applied is strictly defined. Thus a Standardized Business Rule Schema is created with all the rules being optional. By customizing the Purchase Order Business Rule Template, a host creates a document that expresses its unique business rules for the Purchase Order. Dealers must obtain this document from the host to use it as an overlay within their business system when generating a Purchase Order for that host. In this manner, purchase orders are generated that comply with the rules of the selected supplier.

When applying the rule types to a schema such as the Purchase Order, the data elements subject to a business rule are shown by name with the element Datatype changed to match the rule Datatype. This method binds a certain rule type to a particular data element in the schema. For instance, if the element "Quantity" is subject to the LimitedField type rule, it will be designated in the Standard Business Rule Schema as follows;

```
<xsd:element name="Quantity" type = LimitedFieldType minOccurs=(1)>
```

And thus would adopt all the characteristics of the rule Datatype.

At the time of release of these specifications, the above standardized business rules have been applied only to the Purchase Order (Submit) schema. Since business rules are to be utilized by web service hosts only, there will be no business rules applicable to "response" schemas. In the future as users request additional business rule applications the OPEI e-Commerce Committee will address those requests.

Each Standardized Business Rule application to a web service, will be represented by a Standard Business Rules Schema appropriately named for the web service transaction to which it applies. This document specifies which rules may be applied to which data elements within a Standardized Web service Submit Schema. Only those rule types specified may be applied, and then only may be applied to the data elements specified. No business rule of any type may be applied to a data element not subjected to business rule by the Standardized Business Rules Schema. Web service hosts should carefully follow all implementation annotations found in the Standardized Business Rules Schema applicable to each web service.

A web service host applying business rules to a web service is required to generate an instance of the Standardized Business Rules Schema which contains their own customization and make it available to consumers of its web service. This access is to be accomplished by publishing the namespace for the host's business rules document within the Discovery Response schema as described above. Hosts are free to apply any namespace, but it is highly recommended that the document name include clear reference to the schema and version number to which it applies. See example below:

Business Rules Namespace = tmwmanufacturing.com/webservices//businessrules/POBusRulesDoc ver 1.1

It is also the strict responsibility of the web service host to maintain the expiration dates within its business rules documents to ensure currency. This does not require a change of anything other than the date specified within the "expiration" field, if it is the intention of the host to continue use of the rules unchanged. Dealers are entitled to expect that the business rules document they have just downloaded is fully functional up to the specified expiration date, at which time they are responsible for downloading the next current rules document.

Once a given supplier's business rules document has been obtained, a properly programmed dealer's business system will use it as an overlay against the OPEI Standardized Submit Schema for the business transaction. This overlay approach will invoke the designed structure of each business rule, and ensure that any instance of the schema complies fully with the requirements of this specific supplier. This of course includes any customization by the supplier such as the use of specific tables or selected values for each data field. This design anticipates that a dealer's business system may display the choices for field values provided by the supplier in question, within a drop down box, thus assuring the usage of only approved values. It will also suppress fields that are to be ignored by the supplier, and ensure usage of fields required by the supplier.

Although the number of business rule types is limited, and their application to a given web service is strictly controlled by OPEI specifications, it is anticipated that over time more rule types and applications will be desired. Any request for additional rules or their application to other web services should be directed to the OPEI e-Commerce Committee.

Handling Errors

The following request processing model will be used to describe the response scenarios available for the warranty claim web service.

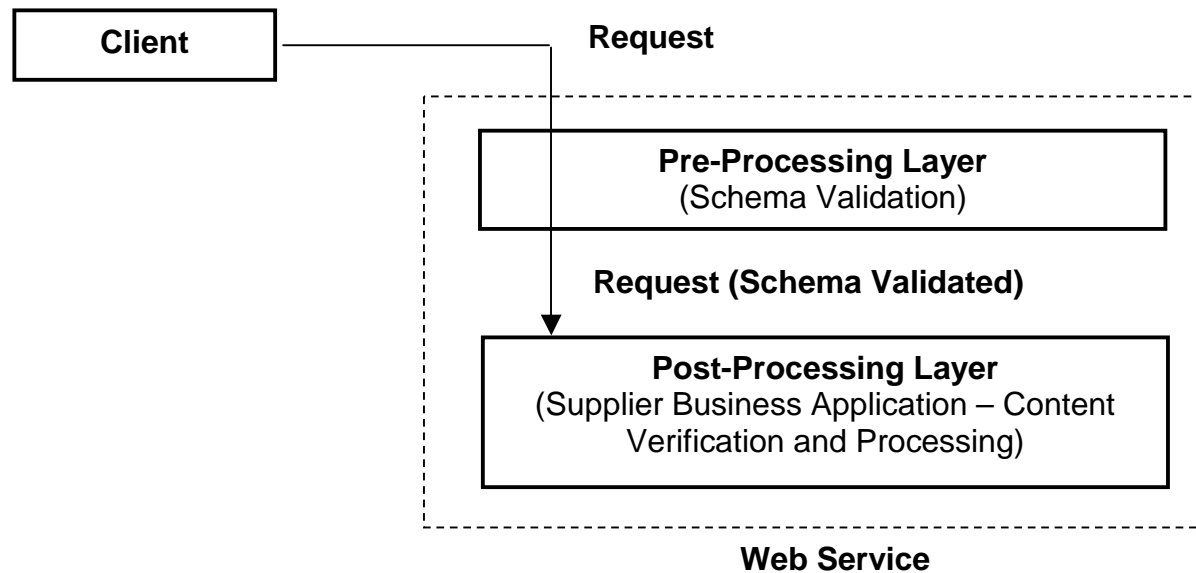


Figure 1: Web Service Request Processing Model

As shown in **Figure 1**, a request received by a web service enters a *Pre-Processing Layer* designed to assess version alignment and to validate the request against the SOAP and the appropriate schemas. A successfully validated request is passed onto the *Post-Processing Layer* which is the embodiment of the web service host's business application.

Faults

In the case of an error being detected during the web service operation, a SOAP fault message is to be employed. Where the web services tool being used does not control this function and automatically generate fault messages, the OPEI **FaultMessage** schema defined below may be used. This message will advise the client (claim submitter) of an error in its attempt to employ the web service. If a fault occurs or there is no response message synchronously received by the client attempting to submit a warranty claim, TOTAL FAILURE is to be assumed. This specification requires the assumption that the entire process has failed and the claim must be resubmitted as if new.

Certain types of errors will be detected in either the “Pre-processing layer” or within the supplier’s application. This specification requires that a SOAP Fault will be used only when an error is detected within the “Pre-processing layer” and prior to forwarding the message to the supplier’s application for processing. Once the message is received by the application, the appropriate Response schema must be used to advise of errors or any other disposition.

The FaultMessage schema to be used in the SOAP fault process is defined in the specification as follows.

```
<xs:element name="FaultMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="FaultType" type="xs:string" />
    <xs:element name="MessageContent" type="xs:string" />
  </xs:sequence>
</xs:complexType>
```

Within this schema, the element FaultType can contain one of three choices of data. For each of these three choices there is a corresponding choice to be used within the MessageContent element. These choices are displayed in Table 1 below.

FaultType Data Choices	MessageContent Data Choices
InvalidXmlSchema	Name of invalid schema
InvalidNamespace	Name of invalid namespace
UnDefinedError	Undefined error message

Table 1: OPEI Fault Types

The FaultMessage schema is used within the SOAP envelope as defined by the W3C SOAP standard. For an example of this usage in compliance with this OPEI Web Service Specification.

Special Situations

Web service implementations ranging from “home-grown” to Microsoft’s ASP.NET, to Apache with JAVA based web service support, provide varying levels of “automated” schema validation in the *Pre-Processing Layer*. These, packages such as Microsoft’s ASP.NET often automatically detect SOAP and warranty claim schema errors, invalid operations, etc., but may miss errors such as an invalid namespace. From a practical perspective, the developer has very little control over the error responses associated with auto-detected violations. Often, they are expressed to the web service client as SOAP faults with a “CLIENT” fault code and a package specific message. Client applications must be flexible enough to handle such messages.

At the other extreme, web service implementers may decide to develop their application “from scratch,” implementing all underlying SOAP manipulation and schema validation. With these solutions, the implementers will need to generate their own SOAP faults in response to errors.

To handle the situation in which the package implemented in the *Pre-Processing Layer* does not automatically detect and respond to all “Pre-processing” errors, or when the implementer has decided to create web service functionality from scratch, the OPEI has developed a custom SOAP fault. This fault is described in the OPEI WSDL document for the Warranty Claim Service and will be presented to the client with a fault code of “CLIENT.” The following is an example of an OPEI SOAP fault nested inside an HTTP response. It indicates that the message received was found to have an error of “InvalidNamespace” and the location of the error is namespace <http://www.tractors.com/WarrantyWebService/WarrantyClaim.xsd>.

```
HTTP/1.1 500 Internal Server Error.  
Server: Microsoft-IIS/5.0  
Date: Tue, 24 Feb 2004 23:36:53 GMT  
Cache-Control: private  
Content-Type: text/xml; charset=utf-8
```

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>
```

```

<soap:Fault>
  <faultcode xmlns:q0="http://schemas.xmlsoap.org/wsdl/soap/">q0:Client</faultcode>
  <faultstring>Error with message version</faultstring>
  <faultactor>http://UserCompany.com/WarrantyWebService Ver 1.1/WarrantyClaimSubmit</faultactor>
  <detail>
    <FaultMessage xmlns="http://www.opei.org/WarrantyWebService Ver 1.1/WarrantyWebService.xsd">
      <FaultType>InvalidNamespace</FaultType>
      .....<MessageContent>http://www.tractors.com/WarrantyWebService/WarrantyClaim.xsd</MessageContent>
    </FaultMessage>
  </detail>
</soap:Fault>
</soap:Body>
</soap:Envelope>

```

The “faultcode” element is defined in the SOAP specification and is used to identify the fault. Possible values are *VersionMismatch*, *MustUnderstand*, *Client*, and *Server*. Here, *VersionMismatch* is set aside to indicate a SOAP envelope with an invalid namespace (i.e., other than http://schemas.xmlsoap.org/soap/envelope/). The WS-I specification recommends using a faultcode of *Client* for errors associated with malformed requests (such as a schema or namespace error).

The “faultstring” element is set aside to carry an error message while the “faultactor” element provides the URL of the web service that caused the fault to occur.

The final “detail” element is designed to carry an OPEI defined XML message called *FaultMessage* that further describes the error. The sub-element <FaultType> communicates the specific type of fault encountered while the <MessageContent> element will contain text further describing the fault. The following constitutes specified error types along with associated message element content.

Fault Type Data Choices	Message Content Data Choices
InvalidXmlSchema	Name of invalid schema
InvalidNamespace	Name of invalid namespace
UnDefinedError	Undefined error message

Table 2: OPEI Fault Types

Response Possibilities

Given the above context, the following types of responses could occur when interacting with an OPEI Standard Web Service.

No Response – As a result of network or web service failure, a client’s request may not receive a response. In other words, the client request “times out.” Total failure is presumed.

SOAP Fault – If the network and web service are working properly the *Pre-Processing Layer* will receive the client request. If there are schema or namespace errors discovered at the “Pre-processing” level, it will respond with a SOAP fault.

Response Schema Message – A successfully validated web service message will pass to the *Post-Processing Layer* which invokes the supplier’s business application. This layer will always respond with a Response message schema that will be delivered to the client as a disposition of the request, submission, or inquiry.

WS-I Conformance

The XML, XSD, SOAP, and WSDL recommendations all allow enough “wobble-room” to make interoperability difficult across web service providers and consumers. As a result, the *Web Services Interoperability (WS-I)* group was formed, and has put together a basic profile outlining best practices to increase the likelihood of two parties interoperating. More information concerning this organization can be found at <http://www.ws-i.org/>.

It is recommended that any WSDL document conform to the “Basic Profile – Version 1.0” (See <http://www.ws-i.org/Documents.aspx> for the details). The WS-I has created a conformance tool that can simplify this task.