OGC 07-007r1

Open Geospatial Consortium Inc.

Date: 2007-05-08

Reference number of this OGC® project document: **OGC 07-007r1**

Version: 0.3.0

Category: OGC® Discussion Paper

Editor: Paul Watson

OWS4 — Topology Quality Assessment Interoperability Program Report

Copyright notice

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type: OGC® Discussion Paper
Document subtype: Interoperability Program Report
Document stage:
Document language: English

Contents

Figures

Tables

i. Submitting organizations

The following organizations submitted this Interoperability Progress Report to the Open Geospatial Consortium Inc.

a) 1Spatial

ii. Submission contact points

All questions regarding this submission should be directed to the editor or the submitters:

CONTACT	COMPANY	EMAIL
Paul Watson	1Spatial	paul.watson[at]1spatial.com

iii. Revision history

Date	Release	Author	Paragraph modified	Description
January 15, 2007	0.1	Paul Watson	All	First Issue
February 12, 2007	0.2	Paul Watson	Implementation	Second Draft
2007-05-08	0.2	Carl Reed	Various	Make ready for posting as DP

iv. Changes to the OGC® Abstract Specification

The OGC® Abstract Specification does not require changes to accommodate this OGC® document.

Foreword

This work is an interoperability program report compiled by the OGC Web Services, Phase 4 initiative of the OGC Interoperability Program.

This document presents the work completed with respect to the topology quality assessment activities within the OWS-4.

This is not a normative document.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

Introduction

As part of the Geo-processing Workflows thread of the OWS4 Web Services Initiative, an on-line topology quality assessment service was proposed to evaluate the logical consistency of features contained within other services in the OWS4 environment. This document describes the approach taken to realizing this service and the capabilities offered by it during the course of the interoperability experiments.

1 Scope

This document describes the purpose and function of the Topology Quality Assessment Service developed and deployed within the Geo-processing workflow thread of the OWS4 interoperability testbed.

2 References

OpenGIS 04-094:2004, *Web Feature Service (WFS) Implementation Specification.*

ISO 19119, *Geographic Information Services.*

ISO 19107, *Geographic Information – Spatial Schema*

ISO 19115, *Metadata*

ISO 19139, *Metadata – XML Schema Implementation*

MGCP Semantic Information Model TRD1 v1.0

OpenGIS 06-187:2006, OWS-4 Workflow Interoperability Progress Report, http://portal.opengeospatial.org/files/?artifact_id=19496

Web Ontology Language (OWL), W3C 2004, http://www.w3.org/2004/OWL/

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1 Datastore

A datastore is an external source of data (expected to be geographic data) such as an OGC WFS. Logical consistency rules are applied to data drawn from one or more datastores.

3.2 Rule

One or more conditions that objects from the datastore should satisfy. A rule is a structured tree of hierarchical conditions, against which objects can be tested. Rules are expressed in a form independent of the schema of any particular datastore. This means they can easily be re-used with different data sources.

In some rules-based systems, the term “rule” refers to a fact-pattern-action triplet:

• The data itself

• The conditions it should satisfy

• Procedures to execute if it does not

TQAS uses this "fact-pattern-action" paradigm. However, the term "rule" refers solely to the pattern part of such a rule. That is, the conditions that the data should satisfy. The action to be performed is selected independently of the pattern and may be simply a reporting action (in the web browser, to an XML file or to a message queue) or a reconciliation action that applies algorithms to correct a problem.

3.3 Session

A session is a sequence of tasks (q.v.) that describe a workflow. For example, open data from one or more data sources, check conformance against selected rules, apply rule-based transformations to resolve problems and commit the changes to the original data source(s).

3.4 Task

An abstraction referring to some process applied to data from a data store. Specific tasks performed by TQAS include:

• Open Data

• Discover Rules

• Check Rules

• Apply Action

• Apply Action Map

• Commit Data

A task may take seconds, minutes, hours or days; therefore it executes asynchronously, so that the user can monitor progress and abort sessions. Tasks continue processing even if the user who started them logs out or shuts down the client computer. All status values and reports associated with the task are retained to be retrieved later.

3.5 Action

An action is a procedure or process to be applied to one or more objects; usually when they are found to violate a particular rule. For example an action could modify a geometry or attribute in order to reconcile it with a particular rule.

3.6 Discovery

The process of analysing data from a data source, looking for patterns in it, and using statistical techniques to deduce a plausible set of rules that the data appears to satisfy.

4 Conventions

4.1 Symbols (and abbreviated terms)

This document uses the symbols and abbreviations defined in Subclause 5.1 of the OGC Web Services Common Implementation Specification 1.1.0 draft [OGC 06-121].

WSDL	Web Service Definition Language
SOAP	Simple Object Access Protocol
BPEL	Business Process Execution Language
REST	Representational State Transfer

Some frequently used abbreviated terms:

API	Application Program Interface
AJAX	Asynchronous Javascript and XML
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off The Shelf
DCE	Distributed Computing Environment
DCP	Distributed Computing Platform
DCOM	Distributed Component Object Model
IDL	Interface Definition Language
ISO	International Organization for Standardization
OGC	Open Geospatial Consortium
UML	Unified Modeling Language
XML	eXtended Markup Language
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional

4.2 UML Notation

The diagrams that appear in this document are presented using the Unified Modeling Language (UML) static structure diagram. The UML notations used in this standard are described in the diagram below.

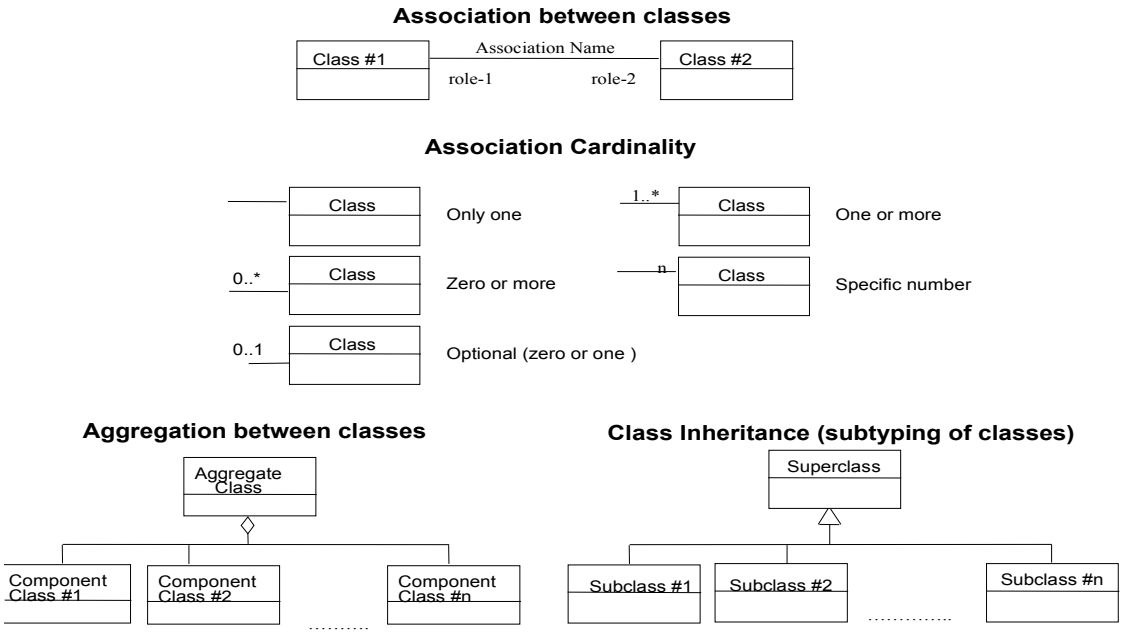


Figure 1 — UML notation

In this standard, the following three stereotypes of UML classes are used:

a) <<Interface>> A definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.

b) <<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations whose primary purpose is to hold the information.

c) <<CodeList>> is a flexible enumeration that uses string values for expressing a list of potential values.

In this standard, the following standard data types are used:

a) CharacterString – A sequence of characters

b) Integer – An integer number

c) Double – A double precision floating point number

d) Float – A single precision floating point number

5 Purpose

Interoperability of services which exchange spatial information is dependent on interoperability of the underlying spatial information at two quite separate levels:

- Syntactic
- Semantic

The syntax or encoding rules of the information must be well understood by the exchanging parties. Geographic Markup Language (GML) provides a good foundation for ensuring syntactic interoperability based as it is on XML and XML Schema with their own well-defined syntactic structure. GML provides additional encoding rules for constructing valid application schemata which further assist in the marshalling and unmarshalling of geographic features and their attributes and the representation of associations between them.

However, while these syntactic constraints are necessary for reliable data transport, they are not sufficient to ensure that the exchanging parties can correctly interpret the features meaning and exploit the features correctly for decision support purposes. To guarantee that the features are consistent with a particular domain interpretation (for example land management or contour data), it is necessary to describe the logical consistency constraints within that domain in a formal way and test the features against these constraints or rules. We know that, in general, land parcels should not overlap and contours should not intersect. However, simply building a GML application schema containing these terms (syntactic structure) does not guarantee that feature instances encoded in GML satisfy the logical domain constraints (semantic structure). The ability of the features to support reasoning and decision support tasks within the problem domain then depends on the degree of conformance with the domain consistency rules. These formal semantic rules must therefore be expressed in addition to the GML application schema.

The purpose of building the Topology Quality Assessment Service (TQAS) within the OWS 4 environment is to discover the extent to which it is possible to express, measure and record results of executing these logical consistency rules on features within a Web Services environment.

6 Requirements

The following high-level requirements for the TQAS service were identified:

1. The service must be able to access simultaneously, for read, feature definitions held in a number of remote Web Feature Services.

2. The service must be able to represent and store logical constraint information (rules) which apply to and between features in any of the accessible Web Feature Services.

3. The service must be able to execute the stored logical constraints on features or groups of features held in different remote services to measure the conformance of the features according to the rules.

4. The service must be able to report on the degree on conformance to an individual or a set of logical constraints at a summary level.

5. The service should be able to report summary conformance metadata in a standard ISO Metadata (19139-compliant) encoding.

6. The service should be able to report individual feature-level non-conformance to individual logical constraints.

7 Inputs

7.1 Government Furnished Information

This section reviews briefly the sources of Government Furnished Information which acted as the basis for the TQAS quality assessment service. This includes both the feature data and its schema which needed to be checked and the constraints which apply to the feature types described by the model.

7.2 Mission Specific Dataset (MSD3) - Aeronautical Profile

The OWS4 demonstration scenario involves, as part of the field hospital planning activity, making updates to aeronautical features to support re-designation of a hangar as a temporary field hospital. These aeronautical features are obtained from a Web Feature Service and are expressed in a GML application schema representing the Mission Specific Dataset v3 (MSD3) object model (ref).

Owing to a lack of a Web Feature Service serving GML compliant to the GML 3.2 application schema developed for the testbed, the features were obtained from a Web Feature Service which had previously been loaded via a ESRI Shape file encoding of the aeronautical features and served via a WFS implementation-derived GML 2.1 application schema.

7.3 MGCP Semantic Information Model

The source of logical constraints for the testbed was the Inter-governmental Geospatial Co-production agreement Semantic Information Model which provides a common logical model describing a common set of feature types found on military topographic maps and explaining the meaning and logical constraints which apply to the capture and representation of feature information within the model.

The model contains a number of domains such as landcover, buildings and constructions, hydrography.

For the purposes of the demonstration, rules were extracted from section 5 Air Transportation. The following is the relevant section from the MGCP UML model for Air transportation:

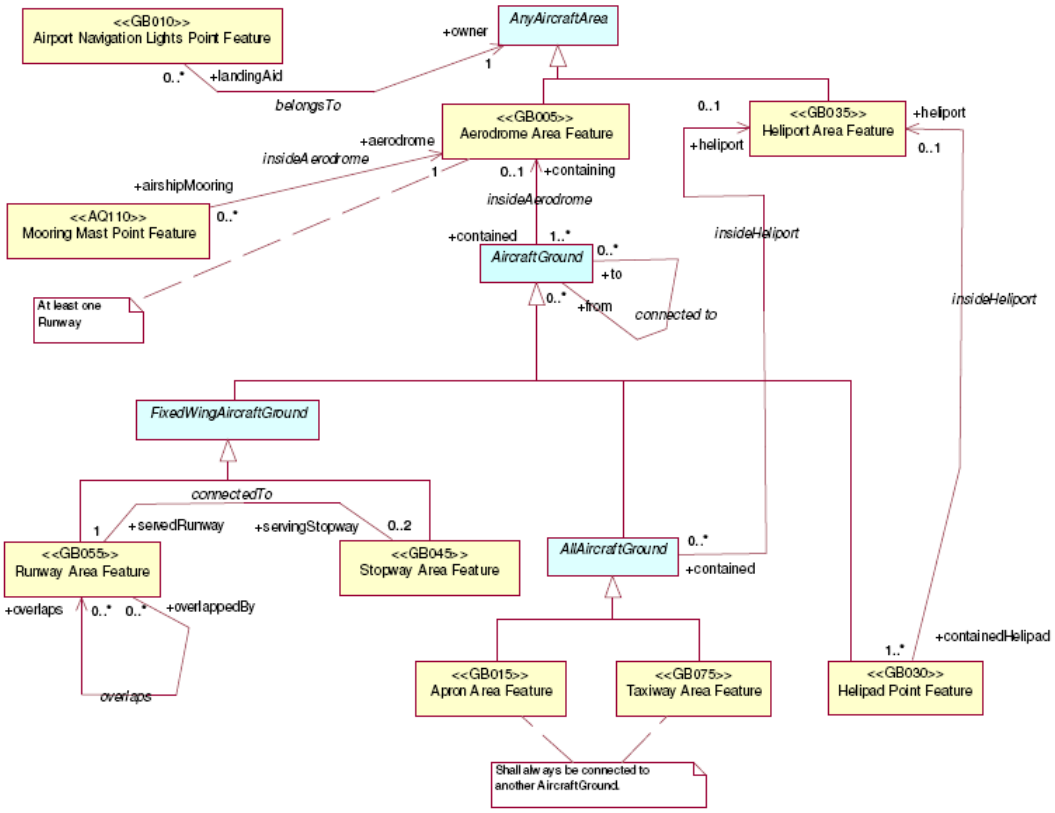


Figure 2: UML model for Mission Specific Dataset (MSD3) - Aeronautical Feature Types.

The rules which constrain spatial and other attribute relationships between these feature classes are expressed in tabular form:

Code	Feature	Association Name	Code	Target Feature	Target / sending features' roles	Association description
GB010	Airport Navigation Lights Point Feature	belongsTo		AnyAircraft Area GB005 Aerodrome GB035 Heliport	owner / landingAid	Aerodrome and Heliport may have landing aids by Airport Navigation Lights inside or outside its boundary. An Airport Navigation Lights Point Feature always belongs to (owned by) an Aerodrome or Heliport which means that it cannot exist by it's own without a nearby Aerodrome.

						Note that this rule doesn't prescribe that all Airport Navigation Lights have to be positioned inside the bounding polygon of the Aerodrome to which they belong.
AQ110	Mooring Mast Point Feature	inside Aerodrome	GB005	Aerodrome Area Feature	aerodrome / airship Mooring	A Mooring Mast Point Feature is always inside an Aerodrome Area Feature. One or more Mooring Mast Point Feature(s) may exist inside an Aerodrome Area Feature.
	AircraftGround GB030 Helipad FixedWingAircraftGround GB045 Stopway GB055 Runway AllAircraftGround GB015 Apron GB075 Taxiway	inside Aerodrome	GB005	Aerodrome Area Feature	containing /contained	An Aerodrome Area Feature shall contain at least one Runway and may contain one or more Stopway(s), Taxiway(s), Apron(s) and Helipad(s). Runways and Stopways are always completely inside an Aerodrome. A Taxiway and an Apron are always completely inside either an aerodrome or a Heliport. A Helipad may be completely inside an Aerodrome or a Heliport.
	AircraftGround GB030 Helipad FixedWing AircraftGround GB045 Stopway GB055 Runway AllAircraftGround GB015 Apron GB075 Taxiway	connected to		AircraftGround GB030 Helipad FixedWing AircraftGround GB045 Stopway GB055 Runway AllAircraft Ground GB015 Apron GB075 Taxiway	to / from	An AircraftGround, i.e. an Apron, Runway, Stopway or a Taxiway, may be connected to one or several other Aircraft-Ground object(s). The following rules apply for the different subclasses: - An Apron shall always be connected to a Taxiway Runway or Stopway - A Runway may exist on its own or it can be connected to one or several Taxiway(s). - A Stopway is always connected to a Runway.
GB055	Runway Area Feature	overlaps	GB055	Runway Area Feature	overlaps / overlapped By	A Runway may overlap one or several other Runway(s). If two Runways overlap, the "full" geometry of both objects should be captured. Thus the two overlapping Runways will form two overlapping rectangular objects.
GB055	Runway Area Feature	connectedTo	GB045	Stopway Area Feature	serving Stopway /served	A Runway may be served by a connected Stopway.

					Runway	A Stopway is always serving a connected Runway.
	AllAircraftGround GB015 Apron GB075 Taxiway	insideHeliport	GB035	Heliport Area Feature	Heliport / contained	A Heliport Area Feature (GB035) may contain Taxiways (GB075) and/or Aprons (GB015).
GB030 Helipad	Helipad Point Feature	insideHeliport	GB035	Heliport Area Feature	Heliport / contained	A Heliport Area Feature shall contain at least one Helipad Point Feature. A Helipad Point Feature may be contained in one Heliport Area Feature (or in an Aerodrome Area Feature).

Table 1: MGCP Semantic Consistency Rules – Aeronautical Feature Types

While some of these rules are mandatory and are therefore logically testable, some are optional (e.g feature A may contain feature B) and should be viewed as guidance rules for data capture rather than as logical constraints.

While the MCGP model and the MSD3 models are distinct, there is obviously common heritage between them and many logical rules from the MGCP document empirically held true for MSD3 feature types.

8 Implementation

8.1 Overview

TQAS has been implemented as a number of stateful web services each of which manage a distinct set of entities within the system, such as datastores or rules. In each case, the service has been exposed using a standardised SOAP binding with request/response messages in RPC/literal form.

In addition, for the benefit of clearer demonstration, a thin, javascript, browser-based client was written to facilitate interaction with each of the service components.

8.2 Datastore

A datastore is an external repository for data, usually including a geographic component. A datastore acts as an abstraction over feature services including OGC Web Feature Service. The user selects some or all of the data in the store by specifying the feature types and attributes of interest and an extent for spatial selection. TQAS checks it for conformance to a defined set of rules and optionally applies automated corrections. Any corrected data may be returned to the same data store or a different data store. An optional schema mapping defines how data should be converted between the schema of a data store and the internal rules schema used by TQAS. It translates between GML feature and attribute types, and classes and properties in the TQAS workspace. The user selects which features and attributes to import and defines the names of the corresponding classes and properties for

the rules environment. This permits data from different stores to be compared more flexibly using a common set of terminology for any given domain such as transport or hydrography. It is possible to define several data stores that access the same WFS data through different schema mappings. It is possible to read data from several data stores into TQAS for processing against a set of rules that analyse relationships between datasets as well as within a dataset. Each data store has two schema mappings associated with it – one input and one output. Output mapping is not needed if TQAS is being used only for checking rules without changing data. It is also possible to input from one store and output to a different store. TQAS also provides integrated support for externally defined ontologies which describe the structure of the data in a specific data store. This is achieved by interfacing with the open source Jena ontology library (see http://jena.sourceforge.net/), allowing ontologies in various formats such as RDF and OWL to be read into TQAS and used for rules authoring and rules-based reasoning.

The datastore is accessed and configured using the TQAS DataStoreManager Web Service. It provides the following interface:

create
createShortcut
delete
deriveIdentityMapping
deriveReverseMapping
deriveTargetSchema
fetchSchema
force
get
getByName
getEnvironment
getEnvironmentValue
getPath
getStoreSpecs
getTarget
set
setEnvironment
testConnection

Table 2: Service interface for TQAS Datastore Manager

All operations are accessed using an RPC/literal SOAP binding as documented in the WSDL contained in Annex B. The key operations are illustrated in section 12

8.3 Rule

A rule is a logical expression which can be used to test the logical consistency of a feature. The rule is expressed using a rules language which is an XML encoding of first order logic, incorporating binary predicates (Boolean operators) for scalar comparisons such as greater or less than and also testing spatial relationships such as containment or within distance. For further details of the language, see section 9. Rules are managed within TQAS by a dedicated web service – the TQAS RuleManager Service. This service allows rule expressions, along with suitable metadata, to be stored and their definitions retrieved and used within conformance checking and data reconciliation tasks. The service interface methods are listed below:

create
createShortcut
delete

force
get
getByName
getEnvironment
getEnvironmentValue
getPath
getTarget
set
setEnvironment

Table 3: Service interface for TQAS Rule Manager

The detailed method signatures are documented in the TQAS WSDL service definition in Annex B. Example usage is contained in section 12.

8.4 Session

The SessionManager service allows the definition of an ordered sequence of tasks to process data. The service also manages the execution of these sequences against feature instance data and storage and retrieval of the resultant metadata.

Task types are:

• *Open Data*, which enables access to data from a defined datastore. A session may choose to open data from a number of data sources and then check rules based on relationships between features stored in different locations.

• *Discover Rules*, which analyses data based on a defined discovery specification to identify candidate rules.

• *Check Rules*, which checks a defined set of rules on the data and reports non-conformances. It is also possible to publish conformance checking results to a CS-W catalogue server. Published metadata consists of data quality items (qualitative and quantative metrics) and is encoded in a standard form (TC211 ISO XML - 19139).

• *Apply Action*, which applies one or more actions to the data.

• *Apply Action Map*, which checks a set of rules defined in an action map and applies the associated action to each non-conforming object.

• *Commit Data*, which will incrementally commit any data changes back to the data store it came from. Typically this occurs after a correcting action has been applied by an action or action map.

• *Copy to...* which will copy data to a different data store.

• *Pause*, which requests Radius Studio to suspend processing to allow results to be examined before processing the next task in the session sequence.

A session can be viewed as a specialised workflow template for data quality testing and reconciliation. The template can be stored, retrieved and modified as necessary to work against different sources of feature data or to incorporate new rule definitions or actions.

The web service interface methods are given below:

abort
closeSession
create
createShortcut
delete
force
get
getByName
getEnvironment
getEnvironmentValue
getErrorXML
getOGCMetadata
GetPath
GetProgress
GetResults
GetStatus
GetTarget
OpenSession
Pause
PublishOGCMetadata
Rewind
Run
Set
SetEnvironment
UpdateFeature

Table 4: Service interface for TQAS Session Manager

The detailed method signatures are documented in the TQAS WSDL service definition in Annex B. Example usage is contained in section 12.

8.5 TQAS client

To simplify interaction with the individual manager web services and to facilitate demonstration of the services, a thin AJAX (javascript) browser-based client application was developed.

8.5.1 Navigation & Entity Metadata

A row of tabs at the top of the user interface selects the component to work with. In this example, the **Rules** component is selected, allowing the user to maintain the rule base. The basic principles described here apply to all of the components; an example of each is shown later. The Rules are presented in a hierarchical folder structure. The path to the current folder appears below the main tabs, in this case:

Radius Studio >> Rules >> Philadelphia >> Fire_Stations 1.0

Philadelphia is the name of the folder and Fire_Stations 1.0 is the name of the selected rule.

The main part of the window is divided into two areas. The form on the left is for creating, deleting and modifying rules and rule folders and for editing options that apply to all the rules. The form on the right is for browsing and editing the currently selected rule. Each of

these forms provides tabs at the bottom that are used to view and edit different parts of this information.

Each persistent entity within the TQAS service in held in a standard repository using a standard common metadata schema which records name and decription along with basic authorship and change history metadata. This metadata is presented using a simple browser form:

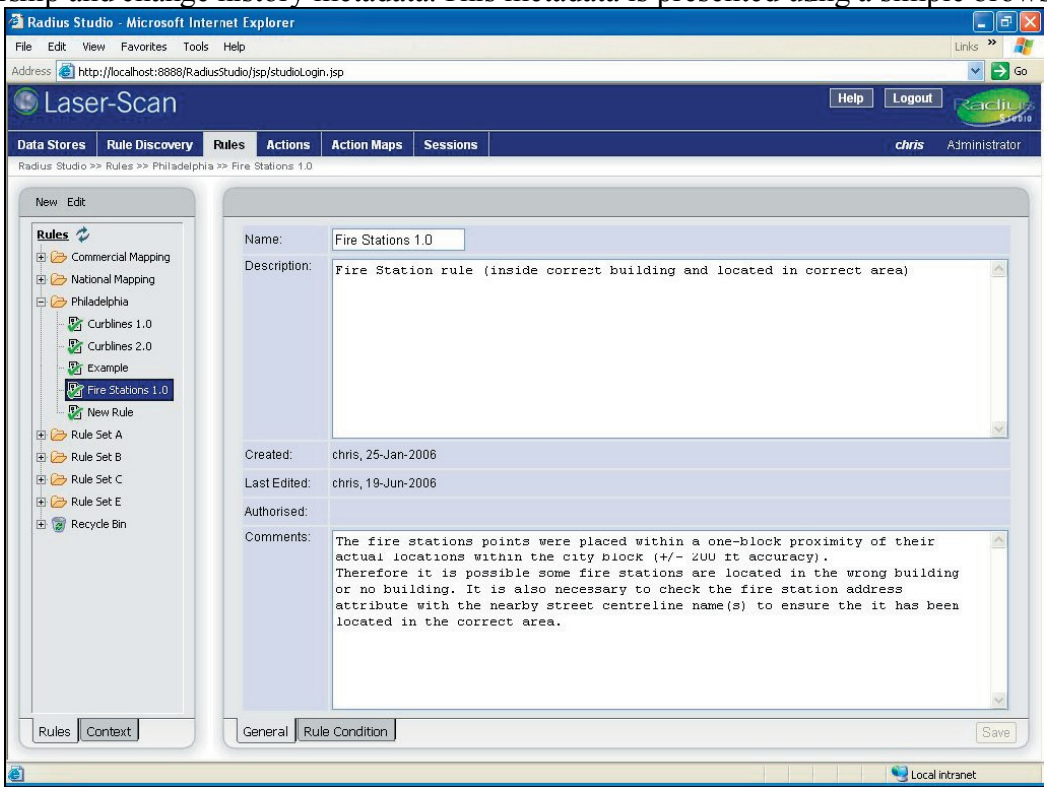


Figure 3: Entity Metadata Entry Form

8.5.2 Datastore Browser

The Data Stores interface is used to define connections to external feature servers that contain data to be processed by TQAS. Each datastore contains the following information in addition to the standard metadata described in the previous section:

• **Input Details** – the information required to connect to the external datasource and schema for an *Open Data* task e.g. URL of a WFS and GetCapabilities()

• **Input Mapping** – a mapping from the supllied application schema to the object-oriented rules authoring schema in TQAS.

• **Output Details** – the information required to connect to the output WFS for a *Copy To...* task, may be but is not necessarily the same as the *Input Details*.

• **Output Mapping** – a mapping from the object-oriented rules authoring schema in TQAS to the application schema of the WFS, by default the inverse of the *Input Mapping*.

The example below illustrates the Input Mapping interface. The names such as BUILDING and CURBLINE refer to feature types accessed through the input connection. The FIRE_STATION feature type has been expanded to show the feature type attributes. The ticks on the left hand side are used to select which feature types and which attributes from each selected feature type are to be analysed by TQAS. All other data is ignored.

The text boxes on the right hand side map feature types onto classes and attribute names onto properties in the TQAS schema. The type of each property may also be overridden – for example to decide whether to represent a decimal type as a *real* or an *integer*. By default, the interface is populated with an identity mapping.

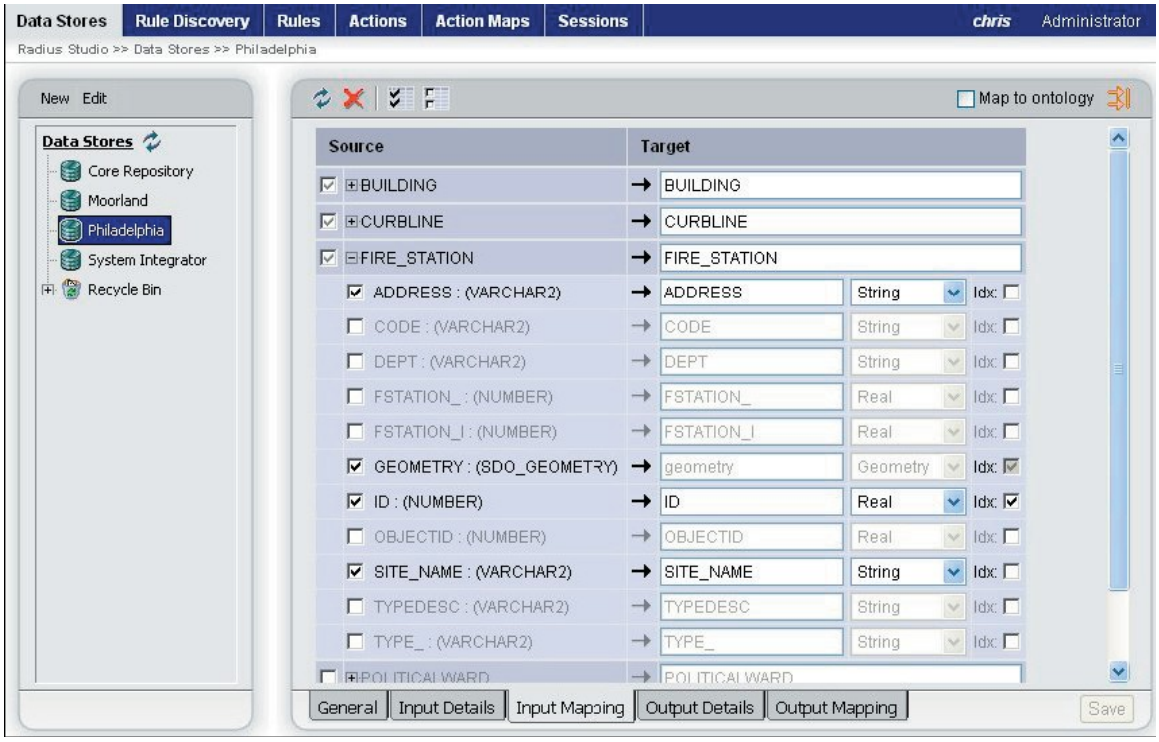


Figure 4: Datastore Browser Form

8.5.3 Rules Browser

The Rule Builder allows the definition of potentially complex rules with an easy to use, tree structrured interface. The rule is expressed as a series of clauses built up using pulldown menus from the bar immediately above the graphical illustration of the rule.

The description at the bottom provides English text representing the currently selected clause; in this case the complete rule. The element details are used to specify the parameters associated with the currently selected rule. This part of the form always includes a description of the information required. In this case, the top-level rule specifies the class to be checked. An optional name label is used when the rule needsto distinguish between two different features of the same class.

There are four types of rule clauses:

• Conditions, including comparisons, logical operators *(and, or, not), if ... then ... else*, existence and looping constructs.

• Values, including constants, dynamic values (attributes), built-in functions, aggregates, etc.

• Relationships, including *equals, less than, begins with, contains*, etc.

• Spatial relationships: TQAS implements all of the Open Geospatial Consortium spatial operators: *equal,*

disjoint, intersect, touch, overlaps, cross, within, contains and geometric relationships like *within distance* or *beyond*. Tooltips are provided to clarify each of these operators. For further details see section 9.

While editing a rule, it may temporarily be incomplete until a new clause is added or parameters are defined. These problems are highlighted clearly in red and a description of what is required displayed.

Multi-level undo/redo is available to recover from mistakes while editing. Drag and drop can be used to reorder clauses of a rule. Cut and paste can be used to transfer all or part of a rule into another rule.

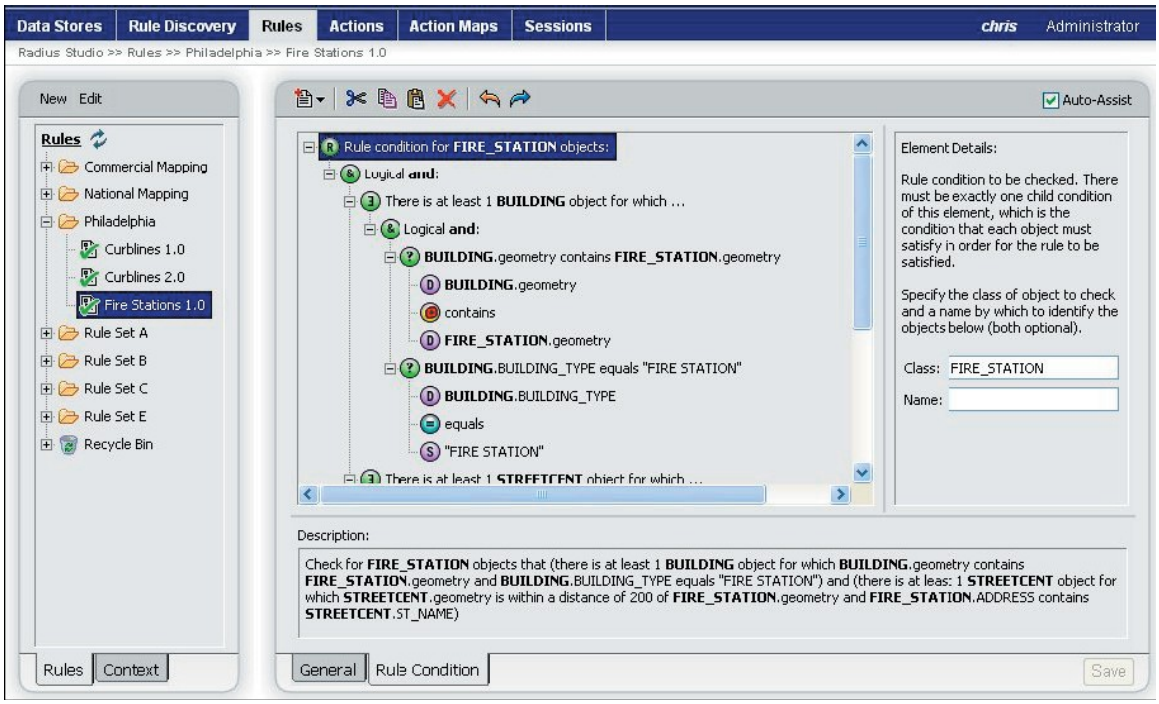


Figure 5: Rules Browser Form

8.5.4 Session Browser

The Sessions interface allows the construction of an ordered sequence of tasks to process data. Task types are:

• *Open Data*, which enables access to data from a defined data source. A session may choose to open data from a number of data sources and then check rules based on relationships between features stored in different locations.

• *Discover Rules*, which analyses data based on a discovery specification to identify candidate rules.

• *Check Rules*, which checks a defined set of rules on the data and reports non-conformances. It is also possible to publish conformance checking results to an OGC CS-W catalogue server. Metadata consists of data quality items (qualitative and quantative metrics) and is encoded in a standard form (TC211 ISO XML - 19139).

• *Apply Action*, which applies one or more actions to the data.

• *Apply Action Map*, which checks a set of rules defined in an action map and applies the associated action to each non-conforming object.

• *Commit Data*, which will incrementally commit any data changes back to the data store it came from. Typically this is after a correcting action has been applied by an action or action map.

• *Copy to...* which will copy data to a different data store.

• *Pause*, which requests TQAS to suspend processing to allow results to be examined before processing the next task.

The interface allows tasks to be added, deleted and reordered and for all required parameters to be browsed and updated.

Multi-level undo / redo may be used to correct mistakes.

Media player style controls are used to run tasks:

a) The play button starts a task or resumes a paused task.

b) The pause button suspends execution of a task.

c) The rewind button rewinds the most recent task performed.

d) The stop button discards all data and results and rewinds to the start of the first task.

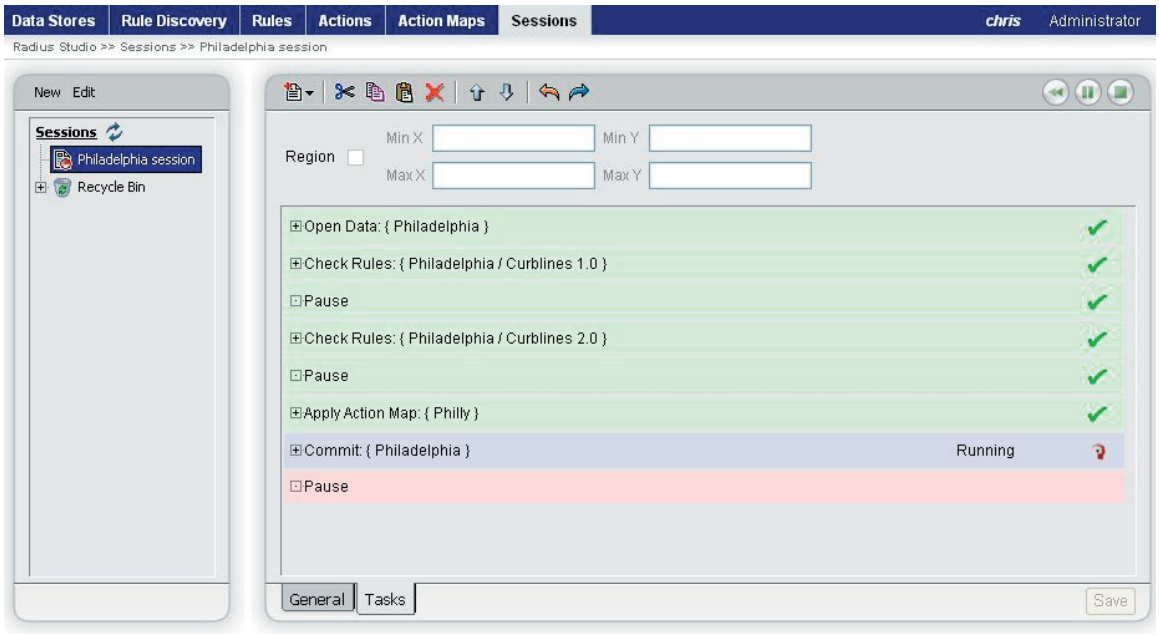


Figure 6: Session Browser Form

9 Rules Language

9.1 Requirements

The following requirements were put forward for the rules language used by TQAS.

a) Unambiguous. It is important that the domain constraints are expressed in a mathematically rigorous way. This allows the rules to be used as the basis of fair testing and means that the results of such testing are fully objective and are not open to interpretation.

b) Logical & Portable. Given the distributed and diverse nature of geographic data, it is advantageous to keep a logical separation between the terms and definitions (ontology) of the feature application schema and the terms and definitions of the domain model to which the rules apply. In other words, the rules are genuinely abstract knowledge and should be decoupled from any particular physical implementation of the instance data to allow reuse of the logical rules with any number of logically compliant feature sources. A feature source may be compliant with the rules either directly, because it is expressed using the same ontology as the rules, or alternatively, a feature source may be compliant with the logical model of the domain even though the syntactic structure of the data differs. In this case, various model transformations, whether simple styling rules or otherwise, may be used to covert feature instance data into a form which can be verified against the rules.

c) Compact. For reasons of manageability and ease of comprehension, it is important that the rules language has a concise grammar. The number of distinct concepts in the language should be kept as small as reasonably possible.

d) Intuitive. It is important that the language be naturalistic and easy to learn. The transformation between constraints expressed using natural, spoken language and the formal rules language should be kept as simple as possible. In some cases, this may conflict with the requirement for simplest terms, above. A balanced view is required in these cases.

e) Quantitative. The language should support quantitative reasoning about the feature data and the development of formal metrics which summarise the level of compliance to data quality measures.

f) Web compatible. The language should be compatible with feature data which is scattered across multiple physical and organisational barriers. Constraints which apply between feature data held under separate authority and control is often equally important to constraints which apply within data under the same authority. The principal implication of this requirement is that the rules language support a naming authority and disambiguation scheme or namespace support.

g) Declarative & Refinable. The nature of collection and management of feature data is such that the entire rules base which constrains the data is very rarely known completely at the start. Both within and between sources of feature data, new constraints are constantly be discovered and added to the rulesbase. It is important, therefore, that the rules language makes it simple to add and refine rules without disrupting the overall structure of the rulesbase unduely

9.2 Choice of rule language

The field of knowledge management or the authoring and exploitation of abstract knowledge representations has received much attention recently through initiatives such as the Semantic Web community (http://www.w3.org/2001/sw/). A key objective of this initiative is to make the exchange of mathematically rigorous models of knowledge such as conceptual graphs possible. This has led to the development of Web Ontology Language (OWL) (http://www.w3.org/TR/owl-features/) as the basis for conceptual representation. This language has its foundations in a field of mathematical logic called Discription Logic and this has been used to formally classify the different complexity classes of different sorts of logical expression. OWL divides the complexity of expressions into three kinds:

a) OWL Lite. This is a simple dialect suitable for expressing simple concepts and relationships.

b) OWL DL (Description Logic). This sub-language represents only concepts which are formally decideable (there exists a decision procedure whether a logic expression is true or false.)

c) OWL Full. This language permits a much richer range of expression (e.g. concepts which may represent both instances and classes) which make the language formally undecideable (there exists no decision procedure).

Development to date has concentrated on Owl Lite and OWL DL precisely because these sub-languages are mathematically tractable and therefore completely general tools support

is feasible. These languages support various reasoning tasks such as deriving complex logical classification schemes (as entailments) from a simpler set of declared relationships.

Some kinds of constraints, especially reasoning over relationships are not supported using the concepts defined in OWL but can be expressed using a rules language layered on top of OWL. An early candidate draft of Semantic Web Rules Language (SWRL) has been proposed within the Semantic Web initiative for this purpose. SWRL extends the conceptual model of OWL to include rules expressions. However, SWRL contains a fixed set of builtin operators which address only basic XML schema datatypes and therefore have no support for derived geometric types. This makes SWRL unsuitable for the current purpose. Instead, a dedicated XML grammer based on first-order logic (predicate logic) was developed and used for TQAS. It should be recognised that the approach here is similar to SWRL, however, and that in the future, it should be possible to convert TQAS to use SWRL-like rules when the operator infrastructure has matured sufficiently to represent spatial operators.

9.3 Rule Elements

TQAS rules have a very simple vocabulary.

- Predicate - an operator which returns either true or false
- Constant
- Variable - free & bound
- Builtin Function
- Logical Connective - NOT, AND, OR
- Quantifier – universal, existential

The way in which these language elements are structured is explained in the following sections.

9.4 Rule Predicates

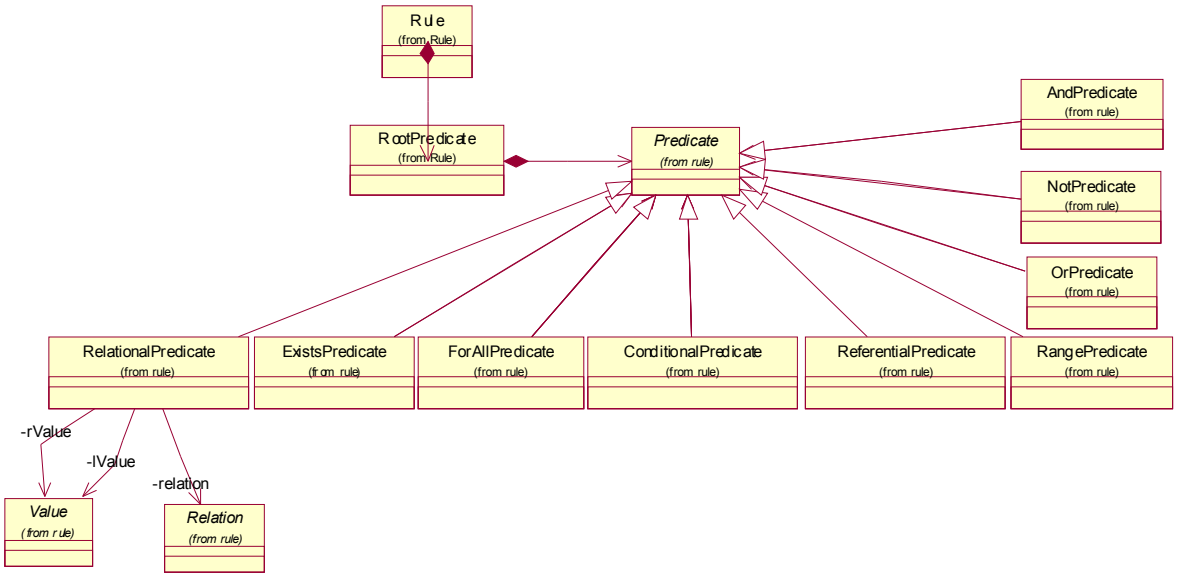


Figure 7: TQAS Rule Predicate Types

The root element of a rule identifies a target class or feature type to which the rule applies. Within the model, this class may be concrete or abstract. In either case, the semantics of application are that, the rule applies equally to all subclasses of the target class. The RootPredicate is a singleton Boolean condition which may be tested to verify whether the feature instance conforms to the rule expression or not. The RootPredicate is composed of any number of child predicates which may be combined using quantification (Existence and For All tests), conditionality (IF…THEN) and logical connectives (AND, OR, NOT) to yield the desired expression.

The simplest predicate type is the RelationalPredicate. It is used to check whether two Values (see below) have a defined relation. It consists of an LeftValue (Lvalue), a RightValue (Rvalue) and a comparison operator (Relation).

The ExistsPredicate is an existential quantifier. It contains a feature type, a numerical quantifier, a relation and a child predicate. It allows expressions of the form, "There exist greater than 3 features of type B for which the following condition holds –> {child predicate}". This may be used to test for the existence or absence of features of a particular type, as in "For Lake features: There exist exactly zero forest features for which the forest geometry is contained within the lake geometry."

The ForAllPredicate is a universal quantifier. It contains a feature type and two child predicates. It allows expressions of the form, "For all features of type B which satisfy {first child condition} verify that {second child condtion} also holds true.

The ConditionalPredicate permits conditional evaluation of parts of a rule. It contains two child predicates. It allows expressions of the form, "If {first child condition} holds then check that {second child condition} also holds."

The ReferentialPredicate tests whether a particular named association exists between two features. It contains two target feature types and an association name. It allows expressions of the form, "Check if there exists a relationship from {feature instance A} to {feature instance B} via the association {reference name}".

The RangePredicate tests whether a value lies in a range. It contains three Values and tests the first supplied Value to find whether in lies between the seond and third supplied Values. It allows expression of the form, "Check whether {First Value} lies between {Second Value} and {Third Value}."

The logical predicates AndPredicate, OrPredicate and NotPredicate allow for Boolean logic to be applied to any of the results returned by other predicate types. AndPredicate and OrPredicate take two child predicates and return the standard Booleanresult. The NotPredicate logically inverts the sense of the child predicate result.

9.5 Rule Values

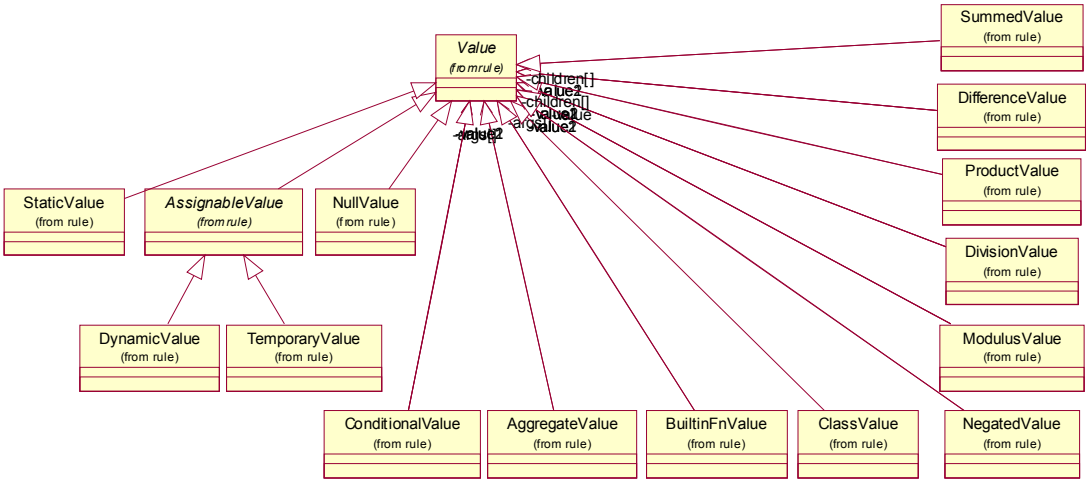


Figure 8: TQAS Rule Value Types

TQAS rules language supports a number of different Value types.

A StaticValue is a typed constant. Its value is assigned explicitly within the rule expression and this value can then be used within other comparisons such as RelationalPredicates. The only datatypes currently supported are simple scalar datatypes such as integers, reals and strings.

An AssignableValue represents a variable in a rule expression is one of two types – a DynamicValue is a typed attribute fetched from a feature instance, a TemporaryValue is used to hold a derived result within a rule for comparison in a later and possibly unrelated clause.

A ConditionalValue is a value which may take one of two values depending upon the truth of a child predicate. It contains two values and a predicate. If the predicate evaluates to true the the first value is returned elsethe second is returned.

An AggregateValue is used to return someAggregated result (sum, average, concatenation, geometric union etc.) from a number of features. It contains a feature type, a feature attribute name, an aggregation function and a child predicate which holds true for the features to be aggregated. It allows expressions of the form, " For features of type {Type} which satisfy {Child Predicate}, compute and return the {Aggregation Function} from the attributes {Attribute Name}."

A BuiltinFnValue is used to derive one Value from another using a specified algorithm. It contains a Value of any type and an algorithm name. A variety of algorithms are supported varying by the datatype of the Value supplied, including simple matematical and string manipulation functions as well as geometric algorithms such as convex hull, buffer or Douglas Peucker simplification. This functionality can be used, for example, to test whether a feature lies within a specified buffer of the geometry of another feature. (The set of supported algorithms can be augmented by implementing an algorithm according to a particular rules system interface. The algorithm then becomes available as another Builtin function within the rules language.)

A ClassValue returns the class name or feature type of a feature.

The final set of Value types are simple arithmetic convenience types, SummedValue, DifferenceValue, ProductValue, DivisionValue, ModulusValue, NegatedValue, having the conventional meanings.

9.6 Rule Relations

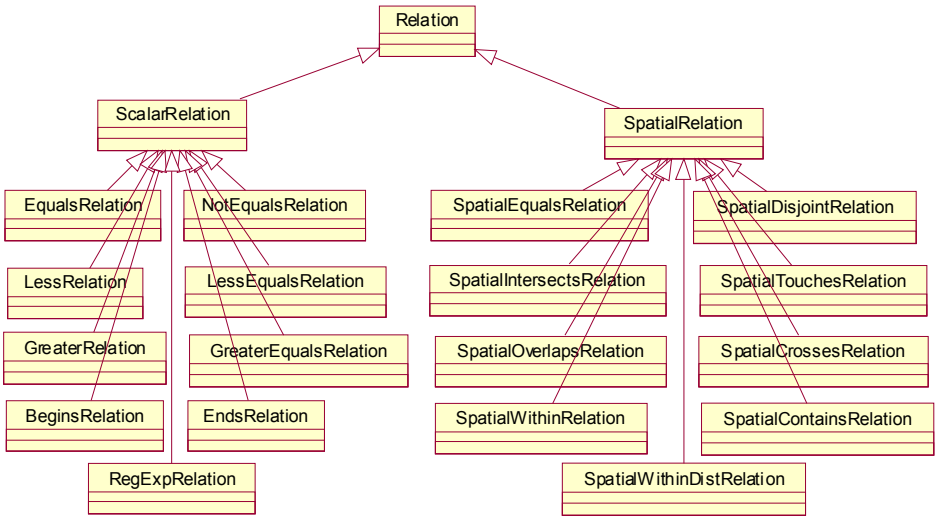


Figure 9: TQAS Rule Relation Types

Relation types are currently gathered into two groups, ScalarRelation and SpatialRelation. ScalarRelation specifies a relationship test between two scalar values of an appropriate type. Numerical relationships supported are EqualsRelation, NotEqualsRelation, LessRelation, LessEqualsRelation, GreaterRelation, GreaterEqualsRelation, with the conventional meanings. Character String relationships are BeginsRelation and

EndsRelation which test whether a character string value begins or ends with the supplied fragment or RegExpRelation which tests whether a character strng value matches a supplied fragment according to a PERL-compatible regular expression.

SpatialRelation types correspond to the OGC Simple Feature specification spatial interaction types and take those meanings. In addition to the topological interaction types, SpatialWithinDistanceRelation can be used to test whether to geometries approach within a user specified distance.

10 Rule Examples

In this section, two very simple example rules are presented to illustrate the range of expression and abstract (implementation independent) nature of the rules.

10.1 Coniferous Forest-Water Area Consistency Rule

This example represents the simplest spatial consistency test possible. It states the constraint that, in most cases, the presence of forest within water areas is inconsistent. Therefore, forest features should be tested to ensure that their geometry does not intersect the geometry of any water body features. The illegal forest features can be depicted graphically:

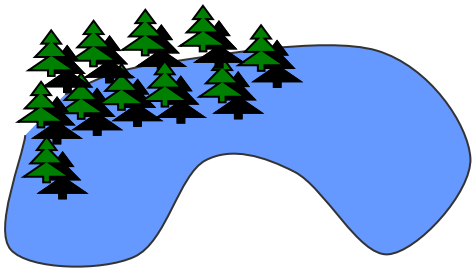


Figure 10: Illegal Coniferous Forest-Water Body Relationship

The constraint might be expressed in prose as follows:

Check for **Coniferous Forest** objects that there are no **Water Area** objects
for which **Coniferous Forest**.geometry overlaps **Water Area**.geometry

The rule can be visualised using a predicate tree structure as follows:

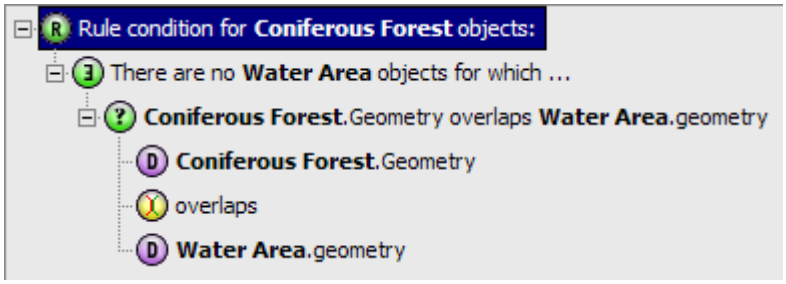


Figure 11: Predicate Tree Structure for Coniferous Forest – Water Area Consistency Rule

This tree shows that the main rule structure is an ExistentialPredicate testing for the existence (or non-existence in this case) of Water Area features which meet a particular RelationalPredicate. The RelationalPredicate tests candidate Water Area features to see whether their geometries overlap the Coniferous Forest feature currently under test.

This predicate tree corresponds very closely with the XML serialisation of this rule:

```
<?xml version="1.0"?>
<Rule>
    <RootPredicate classLabel="Coniferous Forest">
        <ExistsPredicate qualifier="exactly" n="0" classLabel="Water Area">
            <RelationalPredicate>
                <DynamicValue classRef="Coniferous Forest" propName="geometry"/>
                <SpatialOverlapsRelation/>
                <DynamicValue classRef="Water Area" propName="geometry"/>
            </RelationalPredicate>
        </ExistsPredicate>
    </RootPredicate>
</Rule>
```

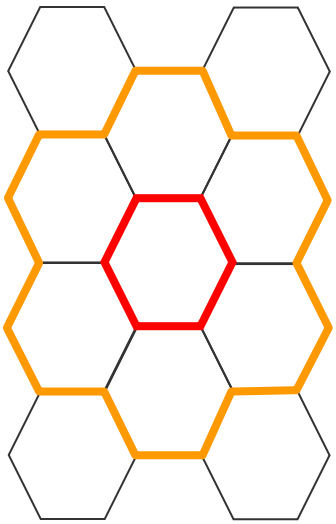
The target feature types appear as the classLabel and classRef attributes of the appropriate Predicates and Values and the feature property names for each DynamicValue is given in the Value propName attribute.

The correspondence between the prose expression, the Rule tree and the XML is extremely close- so close, that the prose can actually be generated from the XML automatically via an XSLT stylesheet (Annex C).

10.2 Island – Water Consistency Rule

In this second example, some slightly more advanced features of the rules language such as BuiltinFunctionValues and AggregateValues are used to show that some complex and powerful expressions may be constructed from realtively simple building blocks.

This rule tests that the shoreline of Island features matches the corresponding limits of all of the Water Areas which border the Island.



We can portray the correct relationship between Island and Water Area:

The Island is the brown hexagon at the centre of the picture. It is surrounded by a number of Water Area features (blue hexagons). The derived shoreline of the Island is drawn in red. The derived set of Water Area features which abut the Island are outlined in orange.

The rule can be expressed in something approaching prose as:

(

Check for **Island** objects that outer_ring(**Island**.geometry) equals intersection(**Island**.geometry,union(**Water Area**.geometry) over all **Water Area** objects for which **Water Area**.geometry touches **Island**.geometry))

Figure 12: Island Water Area Consistency Rule

The corresponding predicate tree looks like this:

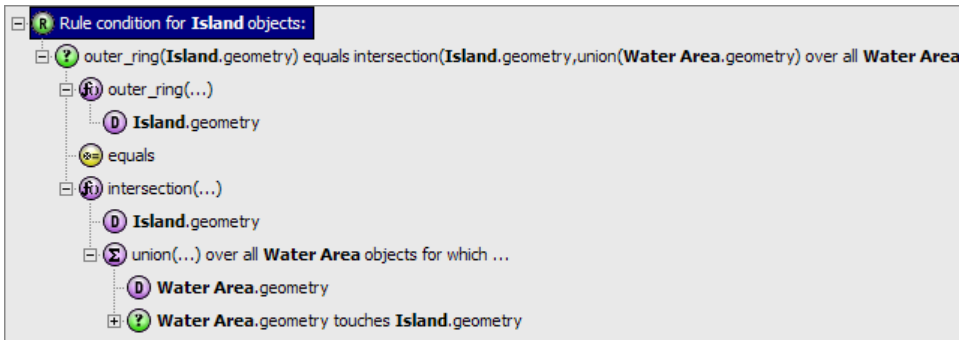


Figure 13: Predicate Tree for Island Water Area Consistency Rule

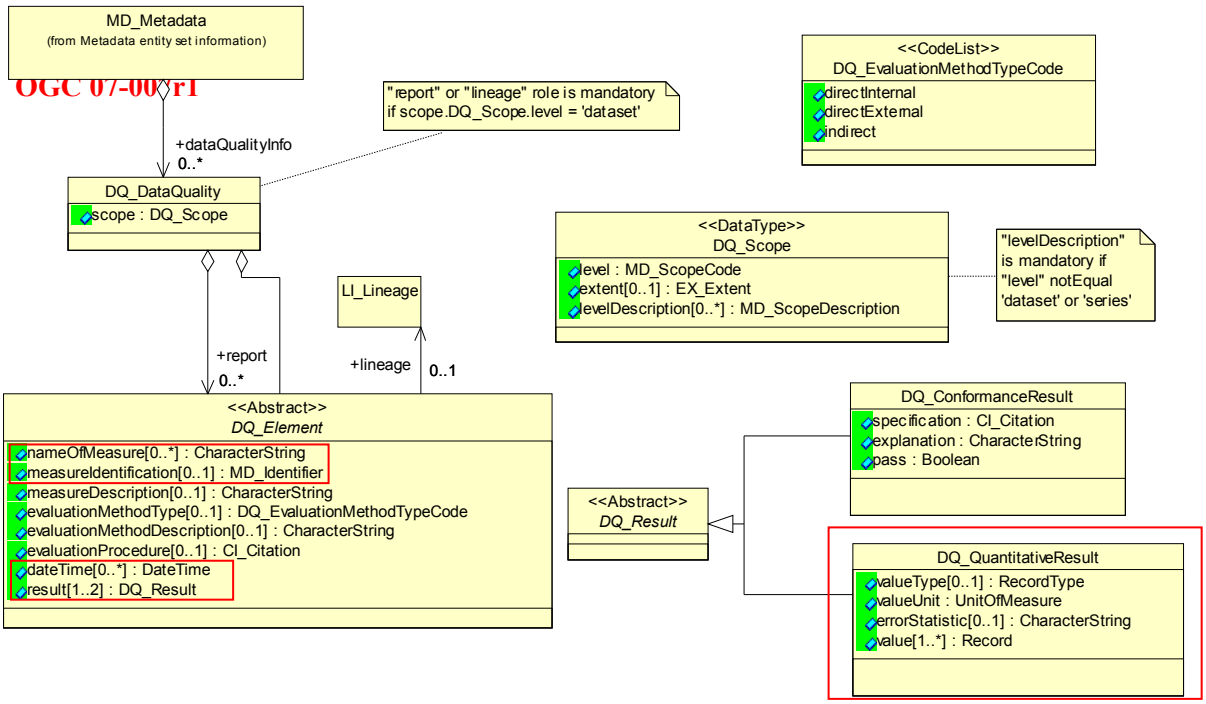
This tree is a simple RelationalPredicate which compares a BuiltinFunctionValue (outer_ring) with another BuiltinFunctionValue (geometric intersection) which in turn nests an AggregateValue (geometric union over Water Areas touching the Island) and tests them for (geometric) equality. The resulting tree is very compact for such a sophisticated expression.

Once again the XML encoding closely mirrors the predicate tree structure.

```
<?xml version="1.0"?>
<Rule>
    <RootPredicate classLabel="Island">
        <RelationalPredicate>
            <BuiltinFnValue fnName="outer_ring">
                <DynamicValue classRef="Island" propName="geometry"/>
            </BuiltinFnValue>
            <SpatialEqualsRelation/>
            <BuiltinFnValue fnName="intersection">
                <DynamicValue classRef="Island" propName="geometry"/>
                <AggregateValue fnName="union" classLabel="Water Area">
                    <DynamicValue classRef="Water Area" propName="geometry"/>
                    <RelationalPredicate>
                        <DynamicValue classRef="Water Area" propName="geometry"/>
                        <SpatialTouchesRelation/>
                        <DynamicValue classRef="Island" propName="geometry"/>
                    </RelationalPredicate>
                </AggregateValue>
            </BuiltinFnValue>
        </RelationalPredicate>
    </RootPredicate>
</Rule>
```

The XSLT stylesheet also caters for rendering this rule into pseudo-prose, although as rules incorporate many BuiltinFnValues and AggregateValues the result becomes less clear and less like spoken English. It remains a good sanity check, however, as reading the styled rule through helps to confirm that the meaning has been captured correctly.

A further advantage of the strict hierarchical structure is that it is simple to parse the rule to determine its validity and feedback any syntactic inconsistencies (e.g. values out of scope) in the rule to the user.



11 Metadata Reporting

The final results of the TQAS conformance tests are obtained in the form of metadata which is compliant to the conceptual model of ISO 19115 Metadata and encoded in the form recommended in ISO 19139 Metadata – XML Schema Implementation. The key elements and attributes are highlighted in red. The nameofMeasure and measureIdentification are taken from the corresponding TQAS rule or Rule folder (container) . The dateTime is taken from the completion time of the conformance check and the results (DQ_Result) are compiled from the appropriate summary statistics within the conformance checking session. The metadata can be published automatically to a compliant OGC CS-W for long-term archiving and to facilitate discovery of data with appropriate quality characteristics.

Figure 14: ISO 19115 Metadata – Data Quality Elements

12 Soap Messaging

12.1 Introduction

This section describes the use of the TQAS SOAP Web Service interfaces for the purposes of the OWS4 GPW Feature Update Workflow. The scenario will use just one of the TQAS service endpoints – the SessionManager. All other objects, Datastores and Rules, are assumed to exist before the session commences. Note that Datastores and Rules may be created dynamically through the relevant web service endpoint just like the Session, but this usage will not be illustrated directly.

12.2 Create Session

The first task is to create a work session definition in which the topology rules will be checked. This session definition consists of a number of tasks:

- A connection must be established to the relevant data source(s)
- A set of rules will be indicated for checking

In the current demo scenario, it is assumed here that both the location of the data (WFS endpoint) and the rule definitions can be pre-arranged. The system supports supplying all of these parameters dynamically. But, it simplifies the experiment if they are fixed ahead of time.

The SOAP message to create the session object within the system is as follows:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://laserscan.com/RadiusStudio">
    <soap:Body>
        <ns:create>
            <DTO_1>
                <id xmlns:ns1="http://www.w3.org/2001/XMLSchema-instance" ns1:nil="true"/>
                <metadataXml><![CDATA[<Metadata>
<Name value="Test Session"/>
<Description value="Created by TQAS Web Service"/>
<Comments value="Should be deleted after use."/>
</Metadata>]]></metadataXml>
                <name>Test Session</name>
                <parentId>0ac8dd7dc0a85abd01bd967519e60748</parentId>
                <referencesXml xmlns:ns2="http://www.w3.org/2001/XMLSchema-instance" ns2:nil="true"/>
                <sequence>1</sequence>
                <xml><![CDATA[<Session>
<Sequence>
<Task label="1" type="Open Data">
<DataStoreRef ref_id="0ac3a94ac0a85abd0181f53b7fc2e033"/>
</Task>
<Task label="2" type="Open Data">
<DataStoreRef ref_id="0ac98ebdc0a85abd019732afc38ace6e"/>
</Task>
<Task label="3" type="Check Rules">
<RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be"/>
</Task>
</Sequence>
</Session>]]></xml>
            </DTO_1>
        </ns:create>
    </soap:Body>
</soap:Envelope>
```

There are two main sections to the request: the metadataXML which includes author and name metadata and definition of the sequence of tasks itself xml. This example states that the system should open and read data from two data sources (each known by a unique identifier) and that is followed by conformance checking a single rule, again known by its unique identifier. The check rules task can also refer to a folder (logical collection of rules) but the syntax is identical.

The Datastore and Rule objects which are referenced within the body of the session definition are defined in a similar way to the Session object, by the create() method, and may be queried and retrieved by invoking the get() and getByName() methods on the appropriate DatastoreManager or RuleManager service endpoint.

The response to this request is given below:

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

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns0="http://laserscan.com/RadiusStudio"
xmlns:ns1="http://dto.studio.radius.laserscan.com/jaws">
    <env:Body>
        <ns0:createResponse>
            <result>
                <id>1a7331fbc0a85abd006c7fcd01bc0b08</id>
                <metadataXml>
                    <Metadata>
                        <Name value="Test Session"/>
                        <Description value="Created by TQAS Web Service"/>
                        <Comments value="Should be deleted after use."/>
                        <Created date="2006-11-24 14:51:34" user="anonymous"/>
                        <LastEdited date="2006-11-24 14:51:34" user="anonymous"/>
                    </Metadata>
                </metadataXml>
                <name>Test Session</name>
                <parentId>0ac8dd7dc0a85abd01bd967519e60748</parentId>
                <referencesXml>
                    <References>
                        <DataStoreRef name="moorland" ref_id="0ac3a94ac0a85abd0181f53b7fc2e033">
                            <Path name="Data Stores"/>
                        </DataStoreRef>
                        <RuleRef name="moorland_WS" ref_id="1a714a8ec0a85abd01dafa55d327e7be">
                            <Path name="Rules"/>
                        </RuleRef>
                        <DataStoreRef name="defra" ref_id="0ac98ebdc0a85abd019732afc38ace6e">
                            <Path name="Data Stores"/>
                        </DataStoreRef>
                    </References>
                </referencesXml>
                <sequence>1</sequence>
                <xml>
                    <Session>
                        <Sequence>
                            <Task label="1" type="Open Data">
                                <DataStoreRef ref_id="0ac3a94ac0a85abd0181f53b7fc2e033"/>
                            </Task>
                            <Task label="2" type="Open Data">
                                <DataStoreRef ref_id="0ac98ebdc0a85abd019732afc38ace6e"/>
                            </Task>
                            <Task label="3" type="Check Rules">
                                <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be"/>
                            </Task>
                        </Sequence>
                    </Session>
                </xml>
            </result>
        </ns0:createResponse>
    </env:Body>
</env:Envelope>
```

This is in effect a copy of the input embellished with further metadata and internal references and crucially an identifier in the id element which can be used to refer back to this session definition.

12.3 Run Session

Once the session has been defined, it can be run using the run() method. The request is given below:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://laserscan.com/RadiusStudio">
    <soap:Body>
        <ns:run>
```

```
            <String_1>1a7331fbc0a85abd006c7fcd01bc0b08</String_1>
        </ns:run>
    </soap:Body>
</soap:Envelope>
```

The request is very simple and requires only the passing of the session identifier.

The response is equally simple in the normal (successful) case:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns0="http://laserscan.com/RadiusStudio"
xmlns:ns1="http://dto.studio.radius.laserscan.com/jaws">
    <env:Body>
        <ns0:runResponse/>
    </env:Body>
</env:Envelope>
```

Once the session has commenced executing, it runs asynchronously in the background through to completion.

12.4 Monitoring progress

The progress of any session can be monitored using the getSessionProgress() method by passing in the persistent identifier of the Session. The request is given below:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://laserscan.com/RadiusStudio">
    <soap:Body>
        <ns:getProgress>
            <String_1>1a7154bfc0a85abd006c7fcd67afd715</String_1>
        </ns:getProgress>
    </soap:Body>
</soap:Envelope>
```

The only parameter required is the session identifier which was obtained when the session was created. The session passes through a number of states while executing and the the sequence number of the current working task and % complete are reported while the session continues to execute. Once the session has run to completion, it is said to be in the "finished" state as given in the status attribute of the Progress element. The matching response is given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns0="http://laserscan.com/RadiusStudio"
xmlns:ns1="http://dto.studio.radius.laserscan.com/jaws">
    <env:Body>
        <ns0:getProgressResponse>
            <result>
                <Progress status="Finished"/>
            </result>
        </ns0:getProgressResponse>
    </env:Body>
</env:Envelope>
```

Once this message has been received, the Session results can be checked.

12.5 Checking results

To obtain the Session results, use the getSessionResults() method. An example request is given below:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://laserscan.com/RadiusStudio">
    <soap:Body>
        <ns:getResults>
            <String_1>1a7154bfc0a85abd006c7fcd67afd715</String_1>
            <String_2>3</String_2>
            <int_3>1</int_3>
            <int_4>10</int_4>
        </ns:getResults>
    </soap:Body>
</soap:Envelope>
```

This method takes four parameters:

1. the identifier of the session
2. the task within the session for which the results are required (3 was the Check Rules task)
3. the index of the first result required starting at 1
4. the index of the final result required (0 for all results)

The response to this message is given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns0="http://laserscan.com/RadiusStudio"
xmlns:ns1="http://dto.studio.radius.laserscan.com/jaws">
    <env:Body>
        <ns0:getResultsResponse>
            <result>
                <Results count="10" finished="1164379794439" first="1" last="10" started="1164379791002">
                    <Summary count="228" error="0" label="3" total="1355" type="Check Rules">
                        <Object class="MOORLAND" count="228" error="0" total="1355"/>
                        <RuleRef count="228" error="0" ref_id="1a714a8ec0a85abd01dafa55d327e7be" total="1355"/>
                    </Summary>
                    <Object class="MOORLAND">
                        <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                        <Attribute name="ID">62.0</Attribute>
                        <Attribute name="geometry">
                            <MBR x0="304193.000139739" x1="361489.001589227" y0="482708.995242653"
y1="527026.999345939"/>
                        </Attribute>
                    </Object>
                    <Object class="MOORLAND">
                        <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                        <Attribute name="ID">203.0</Attribute>
                        <Attribute name="geometry">
                            <MBR x0="247704.99818803798" x1="272127.99874286697" y0="56898.0023714717"
y1="94185.99879352421"/>
                        </Attribute>
                    </Object>
                    <Object class="MOORLAND">
```

```
                        <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                        <Attribute name="ID">299.0</Attribute>
                        <Attribute name="geometry">
                            <MBR x0="268000.9993692" x1="292948.995944581" y0="139344.00244944"
y1="148122.999487678"/>
                        </Attribute>
                    </Object>
                    <Object class="MOORLAND">
                        <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                        <Attribute name="ID">115.0</Attribute>
                        <Attribute name="geometry">
                            <MBR x0="394215.998941906" x1="430414.002116927" y0="371973.00309068"
y1="416624.00187523"/>
                        </Attribute>
                    </Object>
                    <Object class="MOORLAND">
                        <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                        <Attribute name="ID">259.0</Attribute>
                        <Attribute name="geometry">
                            <MBR x0="352585.003449988" x1="377663.997556143" y0="444445.997911707"
y1="468042.003625331"/>
                        </Attribute>
                    </Object>
                    <Object class="MOORLAND">
                        <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                        <Attribute name="ID">97.0</Attribute>
                        <Attribute name="geometry">
                            <MBR x0="363375.001600675" x1="421715.000931845" y0="449858.998456419"
y1="492496.003740612"/>
                        </Attribute>
                    </Object>
                    <Object class="MOORLAND">
                        <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                        <Attribute name="ID">1.0</Attribute>
                        <Attribute name="geometry">
                            <MBR x0="354922.000058606" x1="413573.003094509" y0="489909.00426151"
y1="560768.001084658"/>
                        </Attribute>
                    </Object>
                    <Object class="MOORLAND">
                        <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                        <Attribute name="ID">223.0</Attribute>
                        <Attribute name="geometry">
                            <MBR x0="456788.996152258" x1="497738.999067593" y0="490162.001841706"
y1="515321.999867187"/>
                        </Attribute>
                    </Object>
                    <Object class="MOORLAND">
                        <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                        <Attribute name="ID">144.0</Attribute>
                        <Attribute name="geometry">
                            <MBR x0="377055.998505486" x1="403718.003117747" y0="593894.998580264"
y1="630102.004432648"/>
                        </Attribute>
                    </Object>
```

```
                        <Object class="MOORLAND">
                            <RuleRef ref_id="1a714a8ec0a85abd01dafa55d327e7be">&amp;lt;font
color=red&amp;gt;&amp;lt;u&amp;gt;there is exactly 1 LFA object for which MOORLAND.geometry is contained within
LFA.geometry&amp;lt;/u&amp;gt;&amp;lt;/font&amp;gt;</RuleRef>
                            <Attribute name="ID">315.0</Attribute>
                            <Attribute name="geometry">
                                <MBR x0="210050.999673174" x1="222062.998329656" y0="69590.99903897011"
y1="86005.0005799032"/>
                            </Attribute>
                        </Object>
                    </Results>
                </result>
            </ns0:getResultsResponse>
        </env:Body>
</env:Envelope>
```


The Results element indicates the number of individual results and the start and end timestamps as milliseconds since UTC origin.

The response has two major sections:

1. the summary which given overall conformance levels for the rules checked

2. the detailed per-feature metadata (i.e. which features failed which rule checks). The Rule identifier and text is included with any unique key attributes and the bounding box (envelope) of the checked feature geometry attribute.

This information can be used to make a range of decisions about data quality in the overall context of a workflow. In the example above 288 out of 1355 (16.8%) failed the single rule check giving an overall conformance level of 83.2%.

13 Demonstration Results

13.1 Intent

The intent of developing TQAS within the Geo-processing Workflow thread was that the service should be invoked as a validation process within the context of a Feature Update Workflow orchestrated by an enterprise XML workflow engine via BPEL. This was the main motivation for developing TQAS as a set of standard SOAP web service endpoints. For further details of the workflow elements of OWS-4, see the OWS-4 Workflow IPR given in the references.

Feature data for ingest to the TQAS validation service was not available in time to allow adequate system integration time. As a result, the final integration of TQAS with the enterprise workflow service via SOAP was not possible during the live demonstration event.

The connectivity and interoperability with other OWS4 services was therefore demonstrated using only conventional OGC service interfaces, principally WFS, and manual workflow processes. Nevertheless, the feasibility of performing a distributed

feature update and validation cycle with a WFS client and TQAS service at remote locations sharing feature data from a WFS server at a third remote location was shown and proved to be tractable in real time update.

13.2 Application Schema

MSD3 feature data for ingest to the GPW workflow and the TQAS service was provided in ESRI shape format. This data was to have been converted to a GML 3.2 application schema developed as part of the thread. However, because of the complex mapping process between the MSD3 logical model and the ESRI Shape format, the corresponding transformation from ESRI Shape files to the GML 3.2 application schema developed from the MSD3 UML model proved too difficult and as a result, no GML 3.2 feature data was available for the demonstration.

The production of the feature data for the demonstration was achieved by importing the ESRI Shape files directly into the WFS client tool (GeoMedia, Intergraph) and re-exporting the data as a GML 2.1 compliant application schema using tools from that platform. A consequence of this was that the feature types and attribute names visible in the GML were those names derived from the ESRI Shape file (with complex name mangling applied) rather than the names from the original MSD3 logical model. This made interpretation of the WFS and TQAS results complex on occasions. This problem was mitigated by the fact that most of the MGCP rules under consideration were of a geometric and topological character and made little reference to the mapped attribute names. The aeronautical feature classes involved fortunately had names which were not significantly affected by Shape import and GML re-export process.

13.3 Demonstration Rule

For clarity, the demonstration focused on one simple rule from the MGCP Semantic Information Model – "Runways and Stopways are always completely inside an Aerodrome". The TQAS rules language encodes this rule as follows:

```
<?xml version="1.0"?>
<Rule>
    <RootPredicate classLabel="RUNWAY_A" objLabel="">
        <ExistsPredicate qualifier="at least" n="1" classLabel="AERODROME_A" objLabel="">
            <RelationalPredicate>
                <DynamicValue classRef="AERODROME_A" objRef="" propName="geometry"/>
                <SpatialContainsRelation/>
                <DynamicValue classRef="RUNWAY_A" objRef="" propName="geometry"/>
            </RelationalPredicate>
        </ExistsPredicate>
    </RootPredicate>
</Rule>
```

Note that RUNWAY_A and AERODROME_A were the feature type names advertised from the WFS.

13.4 System Configuration

The physical configuration used to show TQAS functionality during the demonstration event was as follows:

Component	Contributor	Location
Transactional Web Feature Server (WFS-T)	UniBW	Munich, Germany
Transactional WFS Client	Intergraph	Demonstration Location
TQAS Server	1Spatial	Cambridge, UK
TQAS Client	1Spatial	Demonstration Location

Table 5: Feature Update Workflow Components

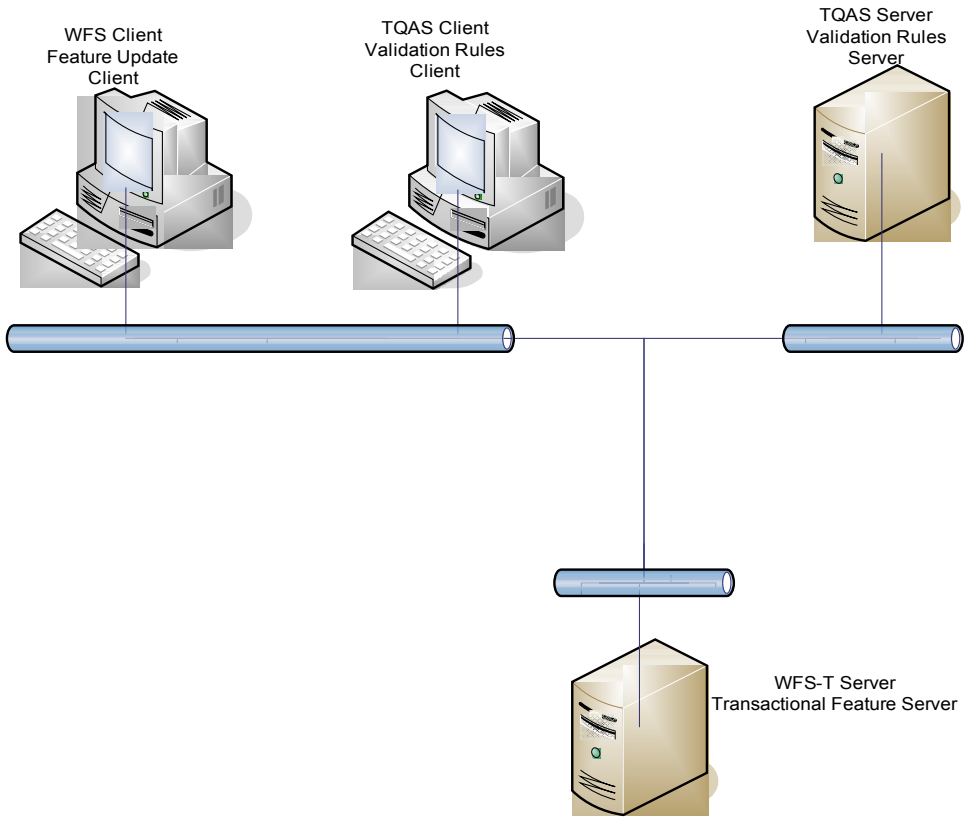


Figure 15: Deployment diagram for TQAS service for use within the Feature Update Workflow

13.5 Demonstration Sequence

During the demonstration, shared access to the MSD3 feature data was provided by a single WFS-T. These features were updated using a transactional WFS-T update client. Modifications included changes to the Runwayand Aerodrome feature types which have a specific set of constraints mandated by the MGCPSemantic Information Model, see section 7.3. Following the feature update, the browser-based TQAS Client application was used to instruct the TQAS Server to access the schema of the remote WFS-T and select the relevant feature types and attributes for validation. A rule conformance validation session was created using the TQAS client, nominating the WFS-T as the feature source and indicating a predefined set of aeronautical validation rules from the MGCP catalogue. The TQAS client was used to execute the conformance checking session on the TQAS server,

to monitor progress of the session and display the conformance result via the browser. The browser clearly showed an "orphaned" runway, a runway not contained within any aerodrome, with a distinctive identifier which had been deliberately created in error by the transactional WFS client. The aeronautical rules from MGCP expressed abstractly in TQAS's rules language were capable of applying this abstract knowledge to concrete feature data obtain via the WFS and return pertinent data quality information to the user updating the data in a fully location transparent way.

As previously noted, it was not possible due to time constraints prior to the demonstration event to integrate the TQAS service with the enterprise workflow engine, however, the exercising of the SOAP interfaces and the location transparent data access and reporting capability give good confidence that this can be achieved with modest effort.

13.6 Conclusions and Opportunities for Further Work

The OWS4 Demonstration has provided good evidence that distributed data validation and data quality reporting is feasible within an OGC Open Web Services environment. In addition, the utility of encoding quality rules and constraints in an open XML-based form has been shown and the approach followed, that of using a first-order logic formalism, appears to yield rules which are both compact and to a large degree generic and implementation independent. The choice to implement the service using standard SOAP RPC/literal bindings and the success of the feature update scenario also shows that the methodology is suitable for integration into much richer and potentially dynamic workflow environments such as are enabled by enterprise workflow technologies like BPEL.

Future work in this area should concentrate on the scope of quality rules in a distributed environment and could examine the relationship between GML application schemas and the underlying conceptual models against which the rules are specified. Eliminating or automating these model transformation steps will be key to improving the usability of an on-line quality assessment service. The value of standard rules domains (e.g. Link-Node Transportation Networks) and the provision of a set of standardized validation rulesets for enhancing semantic interoperability between datasets with distinct application schemas should be explored.

The invocation of the TQAS service from an enterprise workflow engine and specified using BPEL appears quite feasible. However, further work is now required to understand the implications of large distributed transaction systems for current client technology especially in exceptional (fault) conditions. For example, reliable messaging between the components of the service chain may become essential. Introduction of potentially long, validation phases during transactional update may strain the current synchronous mode of communication between a WFS client and server. The practical limits of this and the impact of introducing any asynchronous alternative communication route should be evaluated.

Annex A

Rule Language Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:annotation>
        <xs:documentation>XML Schema for rule definitions</xs:documentation>
    </xs:annotation>
    <xs:element name="Rule">
        <xs:annotation>
            <xs:documentation>Rule definition</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="RootPredicate">
                    <xs:annotation>
                        <xs:documentation>Root predicate of a rule</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="Predicate"/>
                        </xs:sequence>
                        <xs:attribute name="classLabel" type="xs:string" use="optional"/>
                        <xs:attribute name="objLabel" type="xs:string" use="optional"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Value" abstract="true">
        <xs:annotation>
            <xs:documentation>Arbitrary value</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Relation" abstract="true">
        <xs:annotation>
            <xs:documentation>Arbitrary relation</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Predicate" abstract="true">
        <xs:annotation>
            <xs:documentation>Arbitrary predicate</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="NullValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>A null value</xs:documentation>
        </xs:annotation>
        <xs:complexType/>
    </xs:element>
    <xs:element name="StaticValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>A fixed literal value</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:attribute name="datatype" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="boolean"/>
                        <xs:enumeration value="integer"/>
```

```
                        <xs:enumeration value="real"/>
                        <xs:enumeration value="string"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="value" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="DynamicValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>An attribute value from an object</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:attribute name="classRef" type="xs:string" use="optional"/>
            <xs:attribute name="objRef" type="xs:string" use="optional"/>
            <xs:attribute name="propName" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="ClassValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>A value computed by returning the class name of an object</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:attribute name="classRef" type="xs:string" use="optional"/>
            <xs:attribute name="objRef" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="BuiltinFnValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>A value computed using a built-in function</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="fnName" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="AggregateValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>A value computed using an aggregate function on a set of
objects</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Predicate"/>
            </xs:sequence>
            <xs:attribute name="fnName" type="xs:string" use="required"/>
            <xs:attribute name="classLabel" type="xs:string" use="optional"/>
            <xs:attribute name="objLabel" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="NegatedValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>A value computed by reversing the sign of another value</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="SummedValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>The sum of two or more other values</xs:documentation>
        </xs:annotation>
        <xs:complexType>
```

```
            <xs:sequence>
                <xs:element ref="Value" minOccurs="2" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="DifferenceValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>The first child value minus the second</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value" minOccurs="2" maxOccurs="2"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ProductValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>The product of two or more other values</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value" minOccurs="2" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="DivisionValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>The first child value divided by the second</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value" minOccurs="2" maxOccurs="2"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ModulusValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>The remainder after dividing the first child value by the
second</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value" minOccurs="2" maxOccurs="2"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ConditionalValue" substitutionGroup="Value">
        <xs:annotation>
            <xs:documentation>The first child value, if the predicate is true, or the second child value
otherwise</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Predicate"/>
                <xs:element ref="Value" maxOccurs="2"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="EqualsRelation" substitutionGroup="Relation">
        <xs:complexType/>
    </xs:element>
    <xs:element name="NotEqualsRelation" substitutionGroup="Relation">
        <xs:complexType/>
    </xs:element>
    <xs:element name="LessRelation" substitutionGroup="Relation">
        <xs:complexType/>
    </xs:element>
    <xs:element name="LessEqualsRelation" substitutionGroup="Relation">
```

```
    <xs:complexType/>
</xs:element>
<xs:element name="GreaterRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="GreaterEqualsRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="BeginsRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="EndsRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="ContainsRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="RegExpRelation" substitutionGroup="Relation">
    <xs:complexType>
        <xs:attribute name="reType" use="optional" default="pcre">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="pcre"/>
                    <xs:enumeration value="wildcard"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="one_char" type="xs:string" use="optional" default="?"/>
        <xs:attribute name="any_chars" type="xs:string" use="optional" default="%"/>
    </xs:complexType>
</xs:element>
<xs:element name="SpatialEqualsRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="SpatialDisjointRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="SpatialIntersectsRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="SpatialTouchesRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="SpatialOverlapsRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="SpatialCrossesRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="SpatialWithinRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="SpatialContainsRelation" substitutionGroup="Relation">
    <xs:complexType/>
</xs:element>
<xs:element name="SpatialWithinDistRelation" substitutionGroup="Relation">
    <xs:complexType>
        <xs:attribute name="distance" type="xs:double" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="SpatialBeyondRelation" substitutionGroup="Relation">
    <xs:complexType>
        <xs:attribute name="distance" type="xs:double" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="RelationalPredicate" substitutionGroup="Predicate">
    <xs:annotation>
        <xs:documentation>Compare two values using any relation</xs:documentation>
```

```
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value"/>
                <xs:element ref="Relation"/>
                <xs:element ref="Value"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ReferentialPredicate" substitutionGroup="Predicate">
        <xs:annotation>
            <xs:documentation>Test if two objects are related via a reference</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:attribute name="classRef1" type="xs:string" use="optional"/>
            <xs:attribute name="objRef1" type="xs:string" use="optional"/>
            <xs:attribute name="refName" type="xs:string" use="required"/>
            <xs:attribute name="classRef2" type="xs:string" use="optional"/>
            <xs:attribute name="objRef2" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="RangePredicate" substitutionGroup="Predicate">
        <xs:annotation>
            <xs:documentation>Test if the first value lies in the range between the second and third
values</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value" minOccurs="3" maxOccurs="3"/>
            </xs:sequence>
            <xs:attribute name="minInclusive" type="xs:boolean" use="required"/>
            <xs:attribute name="maxInclusive" type="xs:boolean" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="AndPredicate" substitutionGroup="Predicate">
        <xs:annotation>
            <xs:documentation>Test if two or more other predicates are all true</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Predicate" minOccurs="2" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="OrPredicate" substitutionGroup="Predicate">
        <xs:annotation>
            <xs:documentation>Test whether any of the child predicates are true</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Predicate" minOccurs="2" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="NotPredicate" substitutionGroup="Predicate">
        <xs:annotation>
            <xs:documentation>Invert the child predicate's result</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Predicate"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ConditionalPredicate" substitutionGroup="Predicate">
        <xs:annotation>
            <xs:documentation>Conditionally test the second or third child predicate based on the result of the first
child predicate</xs:documentation>
```

```
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Predicate" minOccurs="2" maxOccurs="3"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ExistsPredicate" substitutionGroup="Predicate">
        <xs:annotation>
            <xs:documentation>Test for the existence or absence of objects matching the child
predicate</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Predicate"/>
            </xs:sequence>
            <xs:attribute name="qualifier" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="at least"/>
                        <xs:enumeration value="at most"/>
                        <xs:enumeration value="exactly"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="n" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:integer">
                        <xs:minInclusive value="0"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="classLabel" type="xs:string" use="optional"/>
            <xs:attribute name="objLabel" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="ForAllPredicate" substitutionGroup="Predicate">
        <xs:annotation>
            <xs:documentation>Check that all objects returned by the first child predicate also match the
second</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Predicate" minOccurs="2" maxOccurs="2"/>
            </xs:sequence>
            <xs:attribute name="classLabel" type="xs:string" use="optional"/>
            <xs:attribute name="objLabel" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="NearestPredicate" abstract="true" substitutionGroup="Predicate">
        <xs:annotation>
            <xs:documentation>Test whether the object is one of the nearest objects to a geometry value, possibly
satisfying an additional constraint (not currently implemented)</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value"/>
                <xs:element ref="Predicate" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="classRef" type="xs:string" use="optional"/>
            <xs:attribute name="objRef" type="xs:string" use="optional"/>
            <xs:attribute name="n" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:integer">
                        <xs:minInclusive value="1"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
```

```
            <xs:attribute name="distance" type="xs:double" use="optional"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Annex B

TQAS WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="studio" targetNamespace="http://laserscan.com/RadiusStudio"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:ns1="http://dto.studio.radius.laserscan.com/jaws"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://laserscan.com/RadiusStudio"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <types>
        <schema targetNamespace="http://laserscan.com/RadiusStudio"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ns2="http://dto.studio.radius.laserscan.com/jaws"
xmlns:soap11-enc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://laserscan.com/RadiusStudio"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <import namespace="http://dto.studio.radius.laserscan.com/jaws"/>
            <complexType name="AttributeValue.Array">
                <sequence>
                    <element maxOccurs="unbounded" minOccurs="0" name="value" nillable="true"
type="ns2:AttributeValue"/>
                </sequence>
            </complexType>
            <complexType name="ContainerException">
                <sequence>
                    <element name="message" nillable="true" type="string"/>
                </sequence>
            </complexType>
            <complexType name="DTO">
                <sequence>
                    <element name="id" nillable="true" type="string"/>
                    <element name="metadataXml" nillable="true" type="string"/>
                    <element name="name" nillable="true" type="string"/>
                    <element name="parentId" nillable="true" type="string"/>
                    <element name="referencesXml" nillable="true" type="string"/>
                    <element name="sequence" type="long"/>
                    <element name="xml" nillable="true" type="string"/>
                </sequence>
            </complexType>
            <complexType name="EntityException">
                <sequence>
                    <element name="message" nillable="true" type="string"/>
                </sequence>
            </complexType>
            <complexType name="EntityNotFoundException">
                <complexContent>
                    <extension base="tns:EntityException">
                        <sequence/>
                    </extension>
                </complexContent>
            </complexType>
            <complexType name="EntityReferencedException">
                <complexContent>
                    <extension base="tns:EntityException">
                        <sequence/>
                    </extension>
                </complexContent>
            </complexType>
            <complexType name="EnvironmentVariable.Array">
                <sequence>
                    <element maxOccurs="unbounded" minOccurs="0" name="value" nillable="true"
type="ns2:EnvironmentVariable"/>
                </sequence>
            </complexType>
```

```
            <complexType name="FeatureUpdateResult">
                <sequence>
                    <element name="result" nillable="true" type="string"/>
                    <element maxOccurs="unbounded" minOccurs="0" name="violatedRules" nillable="true"
type="ns2:ViolatedRule"/>
                </sequence>
            </complexType>
            <complexType name="IllegalParentException">
                <complexContent>
                    <extension base="tns:EntityException">
                        <sequence/>
                    </extension>
                </complexContent>
            </complexType>
            <complexType name="NameNotUniqueException">
                <complexContent>
                    <extension base="tns:EntityException">
                        <sequence/>
                    </extension>
                </complexContent>
            </complexType>
            <complexType name="NoResultsException">
                <complexContent>
                    <extension base="tns:EntityException">
                        <sequence/>
                    </extension>
                </complexContent>
            </complexType>
            <complexType name="ReferencesNotFoundException">
                <complexContent>
                    <extension base="tns:EntityException">
                        <sequence/>
                    </extension>
                </complexContent>
            </complexType>
            <complexType name="SessionOperationException">
                <complexContent>
                    <extension base="tns:EntityException">
                        <sequence/>
                    </extension>
                </complexContent>
            </complexType>
            <complexType name="StaleDataException">
                <complexContent>
                    <extension base="tns:EntityException">
                        <sequence/>
                    </extension>
                </complexContent>
            </complexType>
            <complexType name="String.Array">
                <sequence>
                    <element maxOccurs="unbounded" minOccurs="0" name="value" nillable="true"
type="string"/>
                </sequence>
            </complexType>
            <element name="ContainerException" type="tns:ContainerException"/>
            <element name="EntityNotFoundException" type="tns:EntityNotFoundException"/>
            <element name="EntityReferencedException" type="tns:EntityReferencedException"/>
            <element name="IllegalParentException" type="tns:IllegalParentException"/>
            <element name="NameNotUniqueException" type="tns:NameNotUniqueException"/>
            <element name="NoResultsException" type="tns:NoResultsException"/>
            <element name="ReferencesNotFoundException" type="tns:ReferencesNotFoundException"/>
            <element name="SessionOperationException" type="tns:SessionOperationException"/>
            <element name="StaleDataException" type="tns:StaleDataException"/>
        </schema>
        <schema targetNamespace="http://dto.studio.radius.laserscan.com/jaws"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ns1="http://laserscan.com/RadiusStudio" xmlns:soap11-
```

```
enc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://laserscan.com/RadiusStudio"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <import namespace="http://laserscan.com/RadiusStudio"/>
            <complexType name="AttributeValue">
                <sequence>
                    <element name="name" nillable="true" type="string"/>
                    <element name="type" nillable="true" type="string"/>
                    <element name="value" nillable="true" type="string"/>
                </sequence>
            </complexType>
            <complexType name="EnvironmentVariable">
                <sequence>
                    <element name="description" nillable="true" type="string"/>
                    <element name="name" nillable="true" type="string"/>
                    <element name="value" nillable="true" type="string"/>
                </sequence>
            </complexType>
            <complexType name="ViolatedRule">
                <sequence>
                    <element name="reasonForFailure" nillable="true" type="string"/>
                    <element name="ruleId" nillable="true" type="string"/>
                </sequence>
            </complexType>
            <element name="EntityException" type="tns:EntityException"/>
        </schema>
    </types>
    <message name="EntityNotFoundException">
        <part element="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </message>
    <message name="ContainerException">
        <part element="tns:ContainerException" name="ContainerException"/>
    </message>
    <message name="EntityReferencedException">
        <part element="tns:EntityReferencedException" name="EntityReferencedException"/>
    </message>
    <message name="IllegalParentException">
        <part element="tns:IllegalParentException" name="IllegalParentException"/>
    </message>
    <message name="NameNotUniqueException">
        <part element="tns:NameNotUniqueException" name="NameNotUniqueException"/>
    </message>
    <message name="NoResultsException">
        <part element="tns:NoResultsException" name="NoResultsException"/>
    </message>
    <message name="ReferencesNotFoundException">
        <part element="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
    </message>
    <message name="StaleDataException">
        <part element="tns:StaleDataException" name="StaleDataException"/>
    </message>
    <message name="ActionManagerEndpoint_create">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="ActionManagerEndpoint_createResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="ActionManagerEndpoint_createShortcut">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="ActionManagerEndpoint_createShortcutResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="ActionManagerEndpoint_delete">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="ActionManagerEndpoint_deleteResponse"/>
    <message name="ActionManagerEndpoint_force">
```

```
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="ActionManagerEndpoint_forceResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="ActionManagerEndpoint_get">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="ActionManagerEndpoint_getResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="ActionManagerEndpoint_getByName">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="ActionManagerEndpoint_getByNameResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="ActionManagerEndpoint_getEnvironment"/>
    <message name="ActionManagerEndpoint_getEnvironmentResponse">
        <part name="result" type="tns:EnvironmentVariable.Array"/>
    </message>
    <message name="ActionManagerEndpoint_getEnvironmentValue">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="ActionManagerEndpoint_getEnvironmentValueResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="ActionManagerEndpoint_getPath">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="ActionManagerEndpoint_getPathResponse">
        <part name="result" type="tns:String.Array"/>
    </message>
    <message name="ActionManagerEndpoint_getTarget">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="ActionManagerEndpoint_getTargetResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="ActionManagerEndpoint_set">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="ActionManagerEndpoint_setResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="ActionManagerEndpoint_setEnvironment">
        <part name="arrayOfEnvironmentVariable_1" type="tns:EnvironmentVariable.Array"/>
    </message>
    <message name="ActionManagerEndpoint_setEnvironmentResponse"/>
    <message name="ActionMapManagerEndpoint_create">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="ActionMapManagerEndpoint_createResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="ActionMapManagerEndpoint_createShortcut">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="ActionMapManagerEndpoint_createShortcutResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="ActionMapManagerEndpoint_delete">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="ActionMapManagerEndpoint_deleteResponse"/>
    <message name="ActionMapManagerEndpoint_force">
```

```
    <part name="DTO_1" type="tns:DTO"/>
</message>
<message name="ActionMapManagerEndpoint_forceResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="ActionMapManagerEndpoint_get">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="ActionMapManagerEndpoint_getResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="ActionMapManagerEndpoint_getByName">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
</message>
<message name="ActionMapManagerEndpoint_getByNameResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="ActionMapManagerEndpoint_getEnvironment"/>
<message name="ActionMapManagerEndpoint_getEnvironmentResponse">
    <part name="result" type="tns:EnvironmentVariable.Array"/>
</message>
<message name="ActionMapManagerEndpoint_getEnvironmentValue">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="ActionMapManagerEndpoint_getEnvironmentValueResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="ActionMapManagerEndpoint_getPath">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="ActionMapManagerEndpoint_getPathResponse">
    <part name="result" type="tns:String.Array"/>
</message>
<message name="ActionMapManagerEndpoint_getTarget">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="ActionMapManagerEndpoint_getTargetResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="ActionMapManagerEndpoint_set">
    <part name="DTO_1" type="tns:DTO"/>
</message>
<message name="ActionMapManagerEndpoint_setResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="ActionMapManagerEndpoint_setEnvironment">
    <part name="arrayOfEnvironmentVariable_1" type="tns:EnvironmentVariable.Array"/>
</message>
<message name="ActionMapManagerEndpoint_setEnvironmentResponse"/>
<message name="DataStoreManagerEndpoint_create">
    <part name="DTO_1" type="tns:DTO"/>
</message>
<message name="DataStoreManagerEndpoint_createResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="DataStoreManagerEndpoint_createShortcut">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
</message>
<message name="DataStoreManagerEndpoint_createShortcutResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="DataStoreManagerEndpoint_delete">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="DataStoreManagerEndpoint_deleteResponse"/>
<message name="DataStoreManagerEndpoint_deriveIdentityMapping">
```

```
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_deriveIdentityMappingResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_deriveReverseMapping">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_deriveReverseMappingResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_deriveTargetSchema">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
        <part name="String_3" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_deriveTargetSchemaResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_fetchSchema">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_fetchSchemaResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_force">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="DataStoreManagerEndpoint_forceResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="DataStoreManagerEndpoint_get">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_getResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="DataStoreManagerEndpoint_getByName">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_getByNameResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="DataStoreManagerEndpoint_getEnvironment"/>
    <message name="DataStoreManagerEndpoint_getEnvironmentResponse">
        <part name="result" type="tns:EnvironmentVariable.Array"/>
    </message>
    <message name="DataStoreManagerEndpoint_getEnvironmentValue">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_getEnvironmentValueResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_getPath">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_getPathResponse">
        <part name="result" type="tns:String.Array"/>
    </message>
    <message name="DataStoreManagerEndpoint_getStoreSpecs"/>
    <message name="DataStoreManagerEndpoint_getStoreSpecsResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="DataStoreManagerEndpoint_getTarget">
        <part name="String_1" type="xsd:string"/>
    </message>
```

```
<message name="DataStoreManagerEndpoint_getTargetResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="DataStoreManagerEndpoint_set">
    <part name="DTO_1" type="tns:DTO"/>
</message>
<message name="DataStoreManagerEndpoint_setResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="DataStoreManagerEndpoint_setEnvironment">
    <part name="arrayOfEnvironmentVariable_1" type="tns:EnvironmentVariable.Array"/>
</message>
<message name="DataStoreManagerEndpoint_setEnvironmentResponse"/>
<message name="DataStoreManagerEndpoint_testConnection">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="DataStoreManagerEndpoint_testConnectionResponse">
    <part name="result" type="xsd:long"/>
</message>
<message name="DiscoveryManagerEndpoint_create">
    <part name="DTO_1" type="tns:DTO"/>
</message>
<message name="DiscoveryManagerEndpoint_createResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="DiscoveryManagerEndpoint_createShortcut">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
</message>
<message name="DiscoveryManagerEndpoint_createShortcutResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="DiscoveryManagerEndpoint_delete">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="DiscoveryManagerEndpoint_deleteResponse"/>
<message name="DiscoveryManagerEndpoint_force">
    <part name="DTO_1" type="tns:DTO"/>
</message>
<message name="DiscoveryManagerEndpoint_forceResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="DiscoveryManagerEndpoint_get">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="DiscoveryManagerEndpoint_getResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="DiscoveryManagerEndpoint_getByName">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
</message>
<message name="DiscoveryManagerEndpoint_getByNameResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="DiscoveryManagerEndpoint_getDiscoverySpecs"/>
<message name="DiscoveryManagerEndpoint_getDiscoverySpecsResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="DiscoveryManagerEndpoint_getEnvironment"/>
<message name="DiscoveryManagerEndpoint_getEnvironmentResponse">
    <part name="result" type="tns:EnvironmentVariable.Array"/>
</message>
<message name="DiscoveryManagerEndpoint_getEnvironmentValue">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="DiscoveryManagerEndpoint_getEnvironmentValueResponse">
    <part name="result" type="xsd:string"/>
```

```
    </message>
    <message name="DiscoveryManagerEndpoint_getPath">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="DiscoveryManagerEndpoint_getPathResponse">
        <part name="result" type="tns:String.Array"/>
    </message>
    <message name="DiscoveryManagerEndpoint_getTarget">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="DiscoveryManagerEndpoint_getTargetResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="DiscoveryManagerEndpoint_set">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="DiscoveryManagerEndpoint_setResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="DiscoveryManagerEndpoint_setEnvironment">
        <part name="arrayOfEnvironmentVariable_1" type="tns:EnvironmentVariable.Array"/>
    </message>
    <message name="DiscoveryManagerEndpoint_setEnvironmentResponse"/>
    <message name="FolderManagerEndpoint_create">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="FolderManagerEndpoint_createResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="FolderManagerEndpoint_createShortcut">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="FolderManagerEndpoint_createShortcutResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="FolderManagerEndpoint_delete">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="FolderManagerEndpoint_deleteResponse"/>
    <message name="FolderManagerEndpoint_force">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="FolderManagerEndpoint_forceResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="FolderManagerEndpoint_get">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="FolderManagerEndpoint_getResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="FolderManagerEndpoint_getByName">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="FolderManagerEndpoint_getByNameResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="FolderManagerEndpoint_getContents">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="FolderManagerEndpoint_getContentsResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="FolderManagerEndpoint_getEnvironment"/>
    <message name="FolderManagerEndpoint_getEnvironmentResponse">
        <part name="result" type="tns:EnvironmentVariable.Array"/>
```

```
</message>
<message name="FolderManagerEndpoint_getEnvironmentValue">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="FolderManagerEndpoint_getEnvironmentValueResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="FolderManagerEndpoint_getPath">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="FolderManagerEndpoint_getPathResponse">
    <part name="result" type="tns:String.Array"/>
</message>
<message name="FolderManagerEndpoint_getTarget">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="FolderManagerEndpoint_getTargetResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="FolderManagerEndpoint_recursiveCopy">
    <part name="String_1" type="xsd:string"/>
    <part name="DTO_2" type="tns:DTO"/>
</message>
<message name="FolderManagerEndpoint_recursiveCopyResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="FolderManagerEndpoint_set">
    <part name="DTO_1" type="tns:DTO"/>
</message>
<message name="FolderManagerEndpoint_setResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="FolderManagerEndpoint_setEnvironment">
    <part name="arrayOfEnvironmentVariable_1" type="tns:EnvironmentVariable.Array"/>
</message>
<message name="FolderManagerEndpoint_setEnvironmentResponse"/>
<message name="RuleManagerEndpoint_create">
    <part name="DTO_1" type="tns:DTO"/>
</message>
<message name="RuleManagerEndpoint_createResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="RuleManagerEndpoint_createShortcut">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
</message>
<message name="RuleManagerEndpoint_createShortcutResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="RuleManagerEndpoint_delete">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="RuleManagerEndpoint_deleteResponse"/>
<message name="RuleManagerEndpoint_force">
    <part name="DTO_1" type="tns:DTO"/>
</message>
<message name="RuleManagerEndpoint_forceResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="RuleManagerEndpoint_get">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="RuleManagerEndpoint_getResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="RuleManagerEndpoint_getByName">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
```

```
    </message>
    <message name="RuleManagerEndpoint_getByNameResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="RuleManagerEndpoint_getEnvironment"/>
    <message name="RuleManagerEndpoint_getEnvironmentResponse">
        <part name="result" type="tns:EnvironmentVariable.Array"/>
    </message>
    <message name="RuleManagerEndpoint_getEnvironmentValue">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="RuleManagerEndpoint_getEnvironmentValueResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="RuleManagerEndpoint_getPath">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="RuleManagerEndpoint_getPathResponse">
        <part name="result" type="tns:String.Array"/>
    </message>
    <message name="RuleManagerEndpoint_getTarget">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="RuleManagerEndpoint_getTargetResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="RuleManagerEndpoint_set">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="RuleManagerEndpoint_setResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="RuleManagerEndpoint_setEnvironment">
        <part name="arrayOfEnvironmentVariable_1" type="tns:EnvironmentVariable.Array"/>
    </message>
    <message name="RuleManagerEndpoint_setEnvironmentResponse"/>
    <message name="SessionManagerEndpoint_abort">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_abortResponse"/>
    <message name="SessionManagerEndpoint_closeSession">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_closeSessionResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_create">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="SessionManagerEndpoint_createResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="SessionManagerEndpoint_createShortcut">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_createShortcutResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="SessionManagerEndpoint_delete">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_deleteResponse"/>
    <message name="SessionManagerEndpoint_force">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="SessionManagerEndpoint_forceResponse">
        <part name="result" type="tns:DTO"/>
```

```
</message>
<message name="SessionManagerEndpoint_get">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="SessionManagerEndpoint_getByName">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getByNameResponse">
    <part name="result" type="tns:DTO"/>
</message>
<message name="SessionManagerEndpoint_getEnvironment"/>
<message name="SessionManagerEndpoint_getEnvironmentResponse">
    <part name="result" type="tns:EnvironmentVariable.Array"/>
</message>
<message name="SessionManagerEndpoint_getEnvironmentValue">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getEnvironmentValueResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getErrorXML">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getErrorXMLResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getOGCMetadata">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getOGCMetadataResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getPath">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getPathResponse">
    <part name="result" type="tns:String.Array"/>
</message>
<message name="SessionManagerEndpoint_getProgress">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getProgressResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getResults">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
    <part name="int_3" type="xsd:int"/>
    <part name="int_4" type="xsd:int"/>
</message>
<message name="SessionManagerEndpoint_getResultsResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getStatus">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getStatusResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getTarget">
    <part name="String_1" type="xsd:string"/>
</message>
<message name="SessionManagerEndpoint_getTargetResponse">
```

```
        <part name="result" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_openSession">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_openSessionResponse">
        <part name="result" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_pause">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_pauseResponse"/>
    <message name="SessionOperationException">
        <part element="tns:SessionOperationException" name="SessionOperationException"/>
    </message>
    <message name="SessionManagerEndpoint_publishOGCMetadata">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_publishOGCMetadataResponse"/>
    <message name="SessionManagerEndpoint_rewind">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_rewindResponse"/>
    <message name="SessionManagerEndpoint_run">
        <part name="String_1" type="xsd:string"/>
    </message>
    <message name="SessionManagerEndpoint_runResponse"/>
    <message name="SessionManagerEndpoint_set">
        <part name="DTO_1" type="tns:DTO"/>
    </message>
    <message name="SessionManagerEndpoint_setResponse">
        <part name="result" type="tns:DTO"/>
    </message>
    <message name="SessionManagerEndpoint_setEnvironment">
        <part name="arrayOfEnvironmentVariable_1" type="tns:EnvironmentVariable.Array"/>
    </message>
    <message name="SessionManagerEndpoint_setEnvironmentResponse"/>
    <message name="SessionManagerEndpoint_updateFeature">
        <part name="String_1" type="xsd:string"/>
        <part name="String_2" type="xsd:string"/>
        <part name="arrayOfString_3" type="tns:String.Array"/>
        <part name="arrayOfAttributeValue_4" type="tns:AttributeValue.Array"/>
        <part name="arrayOfAttributeValue_5" type="tns:AttributeValue.Array"/>
    </message>
    <message name="SessionManagerEndpoint_updateFeatureResponse">
        <part name="result" type="tns:FeatureUpdateResult"/>
    </message>
    <portType name="ActionManagerEndpoint">
        <operation name="create" parameterOrder="DTO_1">
            <input message="tns:ActionManagerEndpoint_create"/>
            <output message="tns:ActionManagerEndpoint_createResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="createShortcut" parameterOrder="String_1 String_2">
            <input message="tns:ActionManagerEndpoint_createShortcut"/>
            <output message="tns:ActionManagerEndpoint_createShortcutResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="delete" parameterOrder="String_1">
            <input message="tns:ActionManagerEndpoint_delete"/>
            <output message="tns:ActionManagerEndpoint_deleteResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
```

```
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:EntityReferencedException" name="EntityReferencedException"/>
        </operation>
        <operation name="force" parameterOrder="DTO_1">
            <input message="tns:ActionManagerEndpoint_force"/>
            <output message="tns:ActionManagerEndpoint_forceResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="get" parameterOrder="String_1">
            <input message="tns:ActionManagerEndpoint_get"/>
            <output message="tns:ActionManagerEndpoint_getResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getByName" parameterOrder="String_1 String_2">
            <input message="tns:ActionManagerEndpoint_getByName"/>
            <output message="tns:ActionManagerEndpoint_getByNameResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getEnvironment">
            <input message="tns:ActionManagerEndpoint_getEnvironment"/>
            <output message="tns:ActionManagerEndpoint_getEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getEnvironmentValue" parameterOrder="String_1">
            <input message="tns:ActionManagerEndpoint_getEnvironmentValue"/>
            <output message="tns:ActionManagerEndpoint_getEnvironmentValueResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getPath" parameterOrder="String_1">
            <input message="tns:ActionManagerEndpoint_getPath"/>
            <output message="tns:ActionManagerEndpoint_getPathResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getTarget" parameterOrder="String_1">
            <input message="tns:ActionManagerEndpoint_getTarget"/>
            <output message="tns:ActionManagerEndpoint_getTargetResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="set" parameterOrder="DTO_1">
            <input message="tns:ActionManagerEndpoint_set"/>
            <output message="tns:ActionManagerEndpoint_setResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:StaleDataException" name="StaleDataException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="setEnvironment" parameterOrder="arrayOfEnvironmentVariable_1">
            <input message="tns:ActionManagerEndpoint_setEnvironment"/>
            <output message="tns:ActionManagerEndpoint_setEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
    </portType>
    <portType name="ActionMapManagerEndpoint">
        <operation name="create" parameterOrder="DTO_1">
            <input message="tns:ActionMapManagerEndpoint_create"/>
            <output message="tns:ActionMapManagerEndpoint_createResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
```

```
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="createShortcut" parameterOrder="String_1 String_2">
            <input message="tns:ActionMapManagerEndpoint_createShortcut"/>
            <output message="tns:ActionMapManagerEndpoint_createShortcutResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="delete" parameterOrder="String_1">
            <input message="tns:ActionMapManagerEndpoint_delete"/>
            <output message="tns:ActionMapManagerEndpoint_deleteResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:EntityReferencedException" name="EntityReferencedException"/>
        </operation>
        <operation name="force" parameterOrder="DTO_1">
            <input message="tns:ActionMapManagerEndpoint_force"/>
            <output message="tns:ActionMapManagerEndpoint_forceResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="get" parameterOrder="String_1">
            <input message="tns:ActionMapManagerEndpoint_get"/>
            <output message="tns:ActionMapManagerEndpoint_getResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getByName" parameterOrder="String_1 String_2">
            <input message="tns:ActionMapManagerEndpoint_getByName"/>
            <output message="tns:ActionMapManagerEndpoint_getByNameResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getEnvironment">
            <input message="tns:ActionMapManagerEndpoint_getEnvironment"/>
            <output message="tns:ActionMapManagerEndpoint_getEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getEnvironmentValue" parameterOrder="String_1">
            <input message="tns:ActionMapManagerEndpoint_getEnvironmentValue"/>
            <output message="tns:ActionMapManagerEndpoint_getEnvironmentValueResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getPath" parameterOrder="String_1">
            <input message="tns:ActionMapManagerEndpoint_getPath"/>
            <output message="tns:ActionMapManagerEndpoint_getPathResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getTarget" parameterOrder="String_1">
            <input message="tns:ActionMapManagerEndpoint_getTarget"/>
            <output message="tns:ActionMapManagerEndpoint_getTargetResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="set" parameterOrder="DTO_1">
            <input message="tns:ActionMapManagerEndpoint_set"/>
            <output message="tns:ActionMapManagerEndpoint_setResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:StaleDataException" name="StaleDataException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="setEnvironment" parameterOrder="arrayOfEnvironmentVariable_1">
```

```
            <input message="tns:ActionMapManagerEndpoint_setEnvironment"/>
            <output message="tns:ActionMapManagerEndpoint_setEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
    </portType>
    <portType name="DataStoreManagerEndpoint">
        <operation name="create" parameterOrder="DTO_1">
            <input message="tns:DataStoreManagerEndpoint_create"/>
            <output message="tns:DataStoreManagerEndpoint_createResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="createShortcut" parameterOrder="String_1 String_2">
            <input message="tns:DataStoreManagerEndpoint_createShortcut"/>
            <output message="tns:DataStoreManagerEndpoint_createShortcutResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="delete" parameterOrder="String_1">
            <input message="tns:DataStoreManagerEndpoint_delete"/>
            <output message="tns:DataStoreManagerEndpoint_deleteResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:EntityReferencedException" name="EntityReferencedException"/>
        </operation>
        <operation name="deriveIdentityMapping" parameterOrder="String_1 String_2">
            <input message="tns:DataStoreManagerEndpoint_deriveIdentityMapping"/>
            <output message="tns:DataStoreManagerEndpoint_deriveIdentityMappingResponse"/>
        </operation>
        <operation name="deriveReverseMapping" parameterOrder="String_1">
            <input message="tns:DataStoreManagerEndpoint_deriveReverseMapping"/>
            <output message="tns:DataStoreManagerEndpoint_deriveReverseMappingResponse"/>
        </operation>
        <operation name="deriveTargetSchema" parameterOrder="String_1 String_2 String_3">
            <input message="tns:DataStoreManagerEndpoint_deriveTargetSchema"/>
            <output message="tns:DataStoreManagerEndpoint_deriveTargetSchemaResponse"/>
        </operation>
        <operation name="fetchSchema" parameterOrder="String_1">
            <input message="tns:DataStoreManagerEndpoint_fetchSchema"/>
            <output message="tns:DataStoreManagerEndpoint_fetchSchemaResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="force" parameterOrder="DTO_1">
            <input message="tns:DataStoreManagerEndpoint_force"/>
            <output message="tns:DataStoreManagerEndpoint_forceResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="get" parameterOrder="String_1">
            <input message="tns:DataStoreManagerEndpoint_get"/>
            <output message="tns:DataStoreManagerEndpoint_getResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getByName" parameterOrder="String_1 String_2">
            <input message="tns:DataStoreManagerEndpoint_getByName"/>
            <output message="tns:DataStoreManagerEndpoint_getByNameResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getEnvironment">
            <input message="tns:DataStoreManagerEndpoint_getEnvironment"/>
            <output message="tns:DataStoreManagerEndpoint_getEnvironmentResponse"/>
```

```
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getEnvironmentValue" parameterOrder="String_1">
            <input message="tns:DataStoreManagerEndpoint_getEnvironmentValue"/>
            <output message="tns:DataStoreManagerEndpoint_getEnvironmentValueResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getPath" parameterOrder="String_1">
            <input message="tns:DataStoreManagerEndpoint_getPath"/>
            <output message="tns:DataStoreManagerEndpoint_getPathResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getStoreSpecs">
            <input message="tns:DataStoreManagerEndpoint_getStoreSpecs"/>
            <output message="tns:DataStoreManagerEndpoint_getStoreSpecsResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getTarget" parameterOrder="String_1">
            <input message="tns:DataStoreManagerEndpoint_getTarget"/>
            <output message="tns:DataStoreManagerEndpoint_getTargetResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="set" parameterOrder="DTO_1">
            <input message="tns:DataStoreManagerEndpoint_set"/>
            <output message="tns:DataStoreManagerEndpoint_setResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:StaleDataException" name="StaleDataException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="setEnvironment" parameterOrder="arrayOfEnvironmentVariable_1">
            <input message="tns:DataStoreManagerEndpoint_setEnvironment"/>
            <output message="tns:DataStoreManagerEndpoint_setEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="testConnection" parameterOrder="String_1">
            <input message="tns:DataStoreManagerEndpoint_testConnection"/>
            <output message="tns:DataStoreManagerEndpoint_testConnectionResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
    </portType>
    <portType name="DiscoveryManagerEndpoint">
        <operation name="create" parameterOrder="DTO_1">
            <input message="tns:DiscoveryManagerEndpoint_create"/>
            <output message="tns:DiscoveryManagerEndpoint_createResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="createShortcut" parameterOrder="String_1 String_2">
            <input message="tns:DiscoveryManagerEndpoint_createShortcut"/>
            <output message="tns:DiscoveryManagerEndpoint_createShortcutResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="delete" parameterOrder="String_1">
            <input message="tns:DiscoveryManagerEndpoint_delete"/>
            <output message="tns:DiscoveryManagerEndpoint_deleteResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:EntityReferencedException" name="EntityReferencedException"/>
        </operation>
        <operation name="force" parameterOrder="DTO_1">
            <input message="tns:DiscoveryManagerEndpoint_force"/>
```

```
            <output message="tns:DiscoveryManagerEndpoint_forceResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="get" parameterOrder="String_1">
            <input message="tns:DiscoveryManagerEndpoint_get"/>
            <output message="tns:DiscoveryManagerEndpoint_getResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getByName" parameterOrder="String_1 String_2">
            <input message="tns:DiscoveryManagerEndpoint_getByName"/>
            <output message="tns:DiscoveryManagerEndpoint_getByNameResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getDiscoverySpecs">
            <input message="tns:DiscoveryManagerEndpoint_getDiscoverySpecs"/>
            <output message="tns:DiscoveryManagerEndpoint_getDiscoverySpecsResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getEnvironment">
            <input message="tns:DiscoveryManagerEndpoint_getEnvironment"/>
            <output message="tns:DiscoveryManagerEndpoint_getEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getEnvironmentValue" parameterOrder="String_1">
            <input message="tns:DiscoveryManagerEndpoint_getEnvironmentValue"/>
            <output message="tns:DiscoveryManagerEndpoint_getEnvironmentValueResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getPath" parameterOrder="String_1">
            <input message="tns:DiscoveryManagerEndpoint_getPath"/>
            <output message="tns:DiscoveryManagerEndpoint_getPathResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getTarget" parameterOrder="String_1">
            <input message="tns:DiscoveryManagerEndpoint_getTarget"/>
            <output message="tns:DiscoveryManagerEndpoint_getTargetResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="set" parameterOrder="DTO_1">
            <input message="tns:DiscoveryManagerEndpoint_set"/>
            <output message="tns:DiscoveryManagerEndpoint_setResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:StaleDataException" name="StaleDataException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="setEnvironment" parameterOrder="arrayOfEnvironmentVariable_1">
            <input message="tns:DiscoveryManagerEndpoint_setEnvironment"/>
            <output message="tns:DiscoveryManagerEndpoint_setEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
    </portType>
    <portType name="FolderManagerEndpoint">
        <operation name="create" parameterOrder="DTO_1">
            <input message="tns:FolderManagerEndpoint_create"/>
            <output message="tns:FolderManagerEndpoint_createResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
```

```
        <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
    </operation>
    <operation name="createShortcut" parameterOrder="String_1 String_2">
        <input message="tns:FolderManagerEndpoint_createShortcut"/>
        <output message="tns:FolderManagerEndpoint_createShortcutResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="delete" parameterOrder="String_1">
        <input message="tns:FolderManagerEndpoint_delete"/>
        <output message="tns:FolderManagerEndpoint_deleteResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        <fault message="tns:EntityReferencedException" name="EntityReferencedException"/>
    </operation>
    <operation name="force" parameterOrder="DTO_1">
        <input message="tns:FolderManagerEndpoint_force"/>
        <output message="tns:FolderManagerEndpoint_forceResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        <fault message="tns:IllegalParentException" name="IllegalParentException"/>
        <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
    </operation>
    <operation name="get" parameterOrder="String_1">
        <input message="tns:FolderManagerEndpoint_get"/>
        <output message="tns:FolderManagerEndpoint_getResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getByName" parameterOrder="String_1 String_2">
        <input message="tns:FolderManagerEndpoint_getByName"/>
        <output message="tns:FolderManagerEndpoint_getByNameResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getContents" parameterOrder="String_1">
        <input message="tns:FolderManagerEndpoint_getContents"/>
        <output message="tns:FolderManagerEndpoint_getContentsResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getEnvironment">
        <input message="tns:FolderManagerEndpoint_getEnvironment"/>
        <output message="tns:FolderManagerEndpoint_getEnvironmentResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
    </operation>
    <operation name="getEnvironmentValue" parameterOrder="String_1">
        <input message="tns:FolderManagerEndpoint_getEnvironmentValue"/>
        <output message="tns:FolderManagerEndpoint_getEnvironmentValueResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
    </operation>
    <operation name="getPath" parameterOrder="String_1">
        <input message="tns:FolderManagerEndpoint_getPath"/>
        <output message="tns:FolderManagerEndpoint_getPathResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getTarget" parameterOrder="String_1">
        <input message="tns:FolderManagerEndpoint_getTarget"/>
        <output message="tns:FolderManagerEndpoint_getTargetResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
    </operation>
    <operation name="recursiveCopy" parameterOrder="String_1 DTO_2">
        <input message="tns:FolderManagerEndpoint_recursiveCopy"/>
        <output message="tns:FolderManagerEndpoint_recursiveCopyResponse"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
```

```
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
        </operation>
        <operation name="set" parameterOrder="DTO_1">
            <input message="tns:FolderManagerEndpoint_set"/>
            <output message="tns:FolderManagerEndpoint_setResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:StaleDataException" name="StaleDataException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="setEnvironment" parameterOrder="arrayOfEnvironmentVariable_1">
            <input message="tns:FolderManagerEndpoint_setEnvironment"/>
            <output message="tns:FolderManagerEndpoint_setEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
    </portType>
    <portType name="RuleManagerEndpoint">
        <operation name="create" parameterOrder="DTO_1">
            <input message="tns:RuleManagerEndpoint_create"/>
            <output message="tns:RuleManagerEndpoint_createResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="createShortcut" parameterOrder="String_1 String_2">
            <input message="tns:RuleManagerEndpoint_createShortcut"/>
            <output message="tns:RuleManagerEndpoint_createShortcutResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="delete" parameterOrder="String_1">
            <input message="tns:RuleManagerEndpoint_delete"/>
            <output message="tns:RuleManagerEndpoint_deleteResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:EntityReferencedException" name="EntityReferencedException"/>
        </operation>
        <operation name="force" parameterOrder="DTO_1">
            <input message="tns:RuleManagerEndpoint_force"/>
            <output message="tns:RuleManagerEndpoint_forceResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="get" parameterOrder="String_1">
            <input message="tns:RuleManagerEndpoint_get"/>
            <output message="tns:RuleManagerEndpoint_getResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getByName" parameterOrder="String_1 String_2">
            <input message="tns:RuleManagerEndpoint_getByName"/>
            <output message="tns:RuleManagerEndpoint_getByNameResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getEnvironment">
            <input message="tns:RuleManagerEndpoint_getEnvironment"/>
            <output message="tns:RuleManagerEndpoint_getEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
```

```
        <operation name="getEnvironmentValue" parameterOrder="String_1">
            <input message="tns:RuleManagerEndpoint_getEnvironmentValue"/>
            <output message="tns:RuleManagerEndpoint_getEnvironmentValueResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="getPath" parameterOrder="String_1">
            <input message="tns:RuleManagerEndpoint_getPath"/>
            <output message="tns:RuleManagerEndpoint_getPathResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="getTarget" parameterOrder="String_1">
            <input message="tns:RuleManagerEndpoint_getTarget"/>
            <output message="tns:RuleManagerEndpoint_getTargetResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="set" parameterOrder="DTO_1">
            <input message="tns:RuleManagerEndpoint_set"/>
            <output message="tns:RuleManagerEndpoint_setResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:StaleDataException" name="StaleDataException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="setEnvironment" parameterOrder="arrayOfEnvironmentVariable_1">
            <input message="tns:RuleManagerEndpoint_setEnvironment"/>
            <output message="tns:RuleManagerEndpoint_setEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
    </portType>
    <portType name="SessionManagerEndpoint">
        <operation name="abort" parameterOrder="String_1">
            <input message="tns:SessionManagerEndpoint_abort"/>
            <output message="tns:SessionManagerEndpoint_abortResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="closeSession" parameterOrder="String_1">
            <input message="tns:SessionManagerEndpoint_closeSession"/>
            <output message="tns:SessionManagerEndpoint_closeSessionResponse"/>
        </operation>
        <operation name="create" parameterOrder="DTO_1">
            <input message="tns:SessionManagerEndpoint_create"/>
            <output message="tns:SessionManagerEndpoint_createResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="createShortcut" parameterOrder="String_1 String_2">
            <input message="tns:SessionManagerEndpoint_createShortcut"/>
            <output message="tns:SessionManagerEndpoint_createShortcutResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        </operation>
        <operation name="delete" parameterOrder="String_1">
            <input message="tns:SessionManagerEndpoint_delete"/>
            <output message="tns:SessionManagerEndpoint_deleteResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:EntityReferencedException" name="EntityReferencedException"/>
        </operation>
        <operation name="force" parameterOrder="DTO_1">
            <input message="tns:SessionManagerEndpoint_force"/>
            <output message="tns:SessionManagerEndpoint_forceResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
```

```
        <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        <fault message="tns:IllegalParentException" name="IllegalParentException"/>
        <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
    </operation>
    <operation name="get" parameterOrder="String_1">
        <input message="tns:SessionManagerEndpoint_get"/>
        <output message="tns:SessionManagerEndpoint_getResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getByName" parameterOrder="String_1 String_2">
        <input message="tns:SessionManagerEndpoint_getByName"/>
        <output message="tns:SessionManagerEndpoint_getByNameResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getEnvironment">
        <input message="tns:SessionManagerEndpoint_getEnvironment"/>
        <output message="tns:SessionManagerEndpoint_getEnvironmentResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
    </operation>
    <operation name="getEnvironmentValue" parameterOrder="String_1">
        <input message="tns:SessionManagerEndpoint_getEnvironmentValue"/>
        <output message="tns:SessionManagerEndpoint_getEnvironmentValueResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
    </operation>
    <operation name="getErrorXML" parameterOrder="String_1">
        <input message="tns:SessionManagerEndpoint_getErrorXML"/>
        <output message="tns:SessionManagerEndpoint_getErrorXMLResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getOGCMetadata" parameterOrder="String_1 String_2">
        <input message="tns:SessionManagerEndpoint_getOGCMetadata"/>
        <output message="tns:SessionManagerEndpoint_getOGCMetadataResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        <fault message="tns:NoResultsException" name="NoResultsException"/>
    </operation>
    <operation name="getPath" parameterOrder="String_1">
        <input message="tns:SessionManagerEndpoint_getPath"/>
        <output message="tns:SessionManagerEndpoint_getPathResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getProgress" parameterOrder="String_1">
        <input message="tns:SessionManagerEndpoint_getProgress"/>
        <output message="tns:SessionManagerEndpoint_getProgressResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getResults" parameterOrder="String_1 String_2 int_3 int_4">
        <input message="tns:SessionManagerEndpoint_getResults"/>
        <output message="tns:SessionManagerEndpoint_getResultsResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
        <fault message="tns:NoResultsException" name="NoResultsException"/>
    </operation>
    <operation name="getStatus" parameterOrder="String_1">
        <input message="tns:SessionManagerEndpoint_getStatus"/>
        <output message="tns:SessionManagerEndpoint_getStatusResponse"/>
        <fault message="tns:ContainerException" name="ContainerException"/>
        <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
    </operation>
    <operation name="getTarget" parameterOrder="String_1">
        <input message="tns:SessionManagerEndpoint_getTarget"/>
        <output message="tns:SessionManagerEndpoint_getTargetResponse"/>
```

```
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="openSession" parameterOrder="String_1">
            <input message="tns:SessionManagerEndpoint_openSession"/>
            <output message="tns:SessionManagerEndpoint_openSessionResponse"/>
        </operation>
        <operation name="pause" parameterOrder="String_1">
            <input message="tns:SessionManagerEndpoint_pause"/>
            <output message="tns:SessionManagerEndpoint_pauseResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:SessionOperationException" name="SessionOperationException"/>
        </operation>
        <operation name="publishOGCMetadata" parameterOrder="String_1 String_2">
            <input message="tns:SessionManagerEndpoint_publishOGCMetadata"/>
            <output message="tns:SessionManagerEndpoint_publishOGCMetadataResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:NoResultsException" name="NoResultsException"/>
        </operation>
        <operation name="rewind" parameterOrder="String_1 String_2">
            <input message="tns:SessionManagerEndpoint_rewind"/>
            <output message="tns:SessionManagerEndpoint_rewindResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:SessionOperationException" name="SessionOperationException"/>
        </operation>
        <operation name="run" parameterOrder="String_1">
            <input message="tns:SessionManagerEndpoint_run"/>
            <output message="tns:SessionManagerEndpoint_runResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:SessionOperationException" name="SessionOperationException"/>
        </operation>
        <operation name="set" parameterOrder="DTO_1">
            <input message="tns:SessionManagerEndpoint_set"/>
            <output message="tns:SessionManagerEndpoint_setResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
            <fault message="tns:StaleDataException" name="StaleDataException"/>
            <fault message="tns:NameNotUniqueException" name="NameNotUniqueException"/>
            <fault message="tns:EntityNotFoundException" name="EntityNotFoundException"/>
            <fault message="tns:IllegalParentException" name="IllegalParentException"/>
            <fault message="tns:ReferencesNotFoundException" name="ReferencesNotFoundException"/>
        </operation>
        <operation name="setEnvironment" parameterOrder="arrayOfEnvironmentVariable_1">
            <input message="tns:SessionManagerEndpoint_setEnvironment"/>
            <output message="tns:SessionManagerEndpoint_setEnvironmentResponse"/>
            <fault message="tns:ContainerException" name="ContainerException"/>
        </operation>
        <operation name="updateFeature" parameterOrder="String_1 String_2 arrayOfString_3
arrayOfAttributeValue_4 arrayOfAttributeValue_5">
            <input message="tns:SessionManagerEndpoint_updateFeature"/>
            <output message="tns:SessionManagerEndpoint_updateFeatureResponse"/>
        </operation>
    </portType>
    <binding name="ActionManagerEndpointBinding" type="tns:ActionManagerEndpoint">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="create">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
```

```
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="createShortcut">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="delete">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="EntityReferencedException">
                <soap:fault name="EntityReferencedException" use="literal"/>
            </fault>
        </operation>
        <operation name="force">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="get">
```

```
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="getByName">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="getEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="getEnvironmentValue">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="getPath">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
```

```
        <operation name="getTarget">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="set">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="StaleDataException">
                <soap:fault name="StaleDataException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="setEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
    </binding>
    <binding name="ActionMapManagerEndpointBinding" type="tns:ActionMapManagerEndpoint">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="create">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
```

```
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="createShortcut">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="delete">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="EntityReferencedException">
                <soap:fault name="EntityReferencedException" use="literal"/>
            </fault>
        </operation>
        <operation name="force">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="get">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
```

```
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getByName">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getEnvironment">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="getEnvironmentValue">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="getPath">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getTarget">
        <soap:operation soapAction=""/>
        <input>
```

```
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="set">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="StaleDataException">
                <soap:fault name="StaleDataException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="setEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
    </binding>
    <binding name="DataStoreManagerEndpointBinding" type="tns:DataStoreManagerEndpoint">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="create">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
```

```
        <fault name="ReferencesNotFoundException">
            <soap:fault name="ReferencesNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="createShortcut">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="delete">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="EntityReferencedException">
            <soap:fault name="EntityReferencedException" use="literal"/>
        </fault>
    </operation>
    <operation name="deriveIdentityMapping">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
    </operation>
    <operation name="deriveReverseMapping">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
    </operation>
    <operation name="deriveTargetSchema">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
    </operation>
    <operation name="fetchSchema">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
```

```
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="force">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="get">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="getByName">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="getEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
```

```
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="getEnvironmentValue">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="getPath">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getStoreSpecs">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="getTarget">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="set">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
```

```
            <fault name="StaleDataException">
                <soap:fault name="StaleDataException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="setEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="testConnection">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
    </binding>
    <binding name="DiscoveryManagerEndpointBinding" type="tns:DiscoveryManagerEndpoint">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="create">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="createShortcut">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
```

```
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="delete">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="EntityReferencedException">
            <soap:fault name="EntityReferencedException" use="literal"/>
        </fault>
    </operation>
    <operation name="force">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="NameNotUniqueException">
            <soap:fault name="NameNotUniqueException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="IllegalParentException">
            <soap:fault name="IllegalParentException" use="literal"/>
        </fault>
        <fault name="ReferencesNotFoundException">
            <soap:fault name="ReferencesNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="get">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
```

```
<operation name="getByName">
    <soap:operation soapAction=""/>
    <input>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </input>
    <output>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </output>
    <fault name="ContainerException">
        <soap:fault name="ContainerException" use="literal"/>
    </fault>
    <fault name="EntityNotFoundException">
        <soap:fault name="EntityNotFoundException" use="literal"/>
    </fault>
</operation>
<operation name="getDiscoverySpecs">
    <soap:operation soapAction=""/>
    <input>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </input>
    <output>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </output>
    <fault name="ContainerException">
        <soap:fault name="ContainerException" use="literal"/>
    </fault>
</operation>
<operation name="getEnvironment">
    <soap:operation soapAction=""/>
    <input>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </input>
    <output>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </output>
    <fault name="ContainerException">
        <soap:fault name="ContainerException" use="literal"/>
    </fault>
</operation>
<operation name="getEnvironmentValue">
    <soap:operation soapAction=""/>
    <input>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </input>
    <output>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </output>
    <fault name="ContainerException">
        <soap:fault name="ContainerException" use="literal"/>
    </fault>
</operation>
<operation name="getPath">
    <soap:operation soapAction=""/>
    <input>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </input>
    <output>
        <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
    </output>
    <fault name="ContainerException">
        <soap:fault name="ContainerException" use="literal"/>
    </fault>
    <fault name="EntityNotFoundException">
        <soap:fault name="EntityNotFoundException" use="literal"/>
    </fault>
</operation>
<operation name="getTarget">
    <soap:operation soapAction=""/>
```

```
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="set">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="StaleDataException">
                <soap:fault name="StaleDataException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="setEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
    </binding>
    <binding name="FolderManagerEndpointBinding" type="tns:FolderManagerEndpoint">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="create">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
```

```
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="createShortcut">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="delete">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="EntityReferencedException">
                <soap:fault name="EntityReferencedException" use="literal"/>
            </fault>
        </operation>
        <operation name="force">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="get">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
```

```
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getByName">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getContents">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getEnvironment">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="getEnvironmentValue">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="getPath">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
```

```
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="getTarget">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="recursiveCopy">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
        </operation>
        <operation name="set">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="StaleDataException">
                <soap:fault name="StaleDataException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
```

```
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="setEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
    </binding>
    <binding name="RuleManagerEndpointBinding" type="tns:RuleManagerEndpoint">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="create">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="createShortcut">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="delete">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
```

```
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="EntityReferencedException">
                <soap:fault name="EntityReferencedException" use="literal"/>
            </fault>
        </operation>
        <operation name="force">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="IllegalParentException">
                <soap:fault name="IllegalParentException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="get">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="getByName">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="getEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
```

```
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="getEnvironmentValue">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="getPath">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getTarget">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="set">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="StaleDataException">
            <soap:fault name="StaleDataException" use="literal"/>
        </fault>
        <fault name="NameNotUniqueException">
            <soap:fault name="NameNotUniqueException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="IllegalParentException">
            <soap:fault name="IllegalParentException" use="literal"/>
        </fault>
        <fault name="ReferencesNotFoundException">
            <soap:fault name="ReferencesNotFoundException" use="literal"/>
```

```
            </fault>
        </operation>
        <operation name="setEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
    </binding>
    <binding name="SessionManagerEndpointBinding" type="tns:SessionManagerEndpoint">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="abort">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="closeSession">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
        </operation>
        <operation name="create">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="NameNotUniqueException">
                <soap:fault name="NameNotUniqueException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="ReferencesNotFoundException">
                <soap:fault name="ReferencesNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="createShortcut">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
```

```
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="delete">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="EntityReferencedException">
            <soap:fault name="EntityReferencedException" use="literal"/>
        </fault>
    </operation>
    <operation name="force">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="NameNotUniqueException">
            <soap:fault name="NameNotUniqueException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="IllegalParentException">
            <soap:fault name="IllegalParentException" use="literal"/>
        </fault>
        <fault name="ReferencesNotFoundException">
            <soap:fault name="ReferencesNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="get">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getByName">
        <soap:operation soapAction=""/>
```

```
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="getEnvironment">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="getEnvironmentValue">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
        </operation>
        <operation name="getErrorXML">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
        </operation>
        <operation name="getOGCMetadata">
            <soap:operation soapAction=""/>
            <input>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </input>
            <output>
                <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
            </output>
            <fault name="ContainerException">
                <soap:fault name="ContainerException" use="literal"/>
            </fault>
            <fault name="EntityNotFoundException">
                <soap:fault name="EntityNotFoundException" use="literal"/>
            </fault>
            <fault name="NoResultsException">
                <soap:fault name="NoResultsException" use="literal"/>
```

```
        </fault>
    </operation>
    <operation name="getPath">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getProgress">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getResults">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="NoResultsException">
            <soap:fault name="NoResultsException" use="literal"/>
        </fault>
    </operation>
    <operation name="getStatus">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="getTarget">
        <soap:operation soapAction=""/>
        <input>
```

```
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="openSession">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
    </operation>
    <operation name="pause">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="SessionOperationException">
            <soap:fault name="SessionOperationException" use="literal"/>
        </fault>
    </operation>
    <operation name="publishOGCMetadata">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="NoResultsException">
            <soap:fault name="NoResultsException" use="literal"/>
        </fault>
    </operation>
    <operation name="rewind">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
```

```
        <fault name="SessionOperationException">
            <soap:fault name="SessionOperationException" use="literal"/>
        </fault>
    </operation>
    <operation name="run">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="SessionOperationException">
            <soap:fault name="SessionOperationException" use="literal"/>
        </fault>
    </operation>
    <operation name="set">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
        <fault name="StaleDataException">
            <soap:fault name="StaleDataException" use="literal"/>
        </fault>
        <fault name="NameNotUniqueException">
            <soap:fault name="NameNotUniqueException" use="literal"/>
        </fault>
        <fault name="EntityNotFoundException">
            <soap:fault name="EntityNotFoundException" use="literal"/>
        </fault>
        <fault name="IllegalParentException">
            <soap:fault name="IllegalParentException" use="literal"/>
        </fault>
        <fault name="ReferencesNotFoundException">
            <soap:fault name="ReferencesNotFoundException" use="literal"/>
        </fault>
    </operation>
    <operation name="setEnvironment">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </output>
        <fault name="ContainerException">
            <soap:fault name="ContainerException" use="literal"/>
        </fault>
    </operation>
    <operation name="updateFeature">
        <soap:operation soapAction=""/>
        <input>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
        </input>
        <output>
            <soap:body namespace="http://laserscan.com/RadiusStudio" use="literal"/>
```

```
            </output>
        </operation>
    </binding>
    <service name="studio">
        <port binding="tns:ActionManagerEndpointBinding" name="ActionManagerEndpointPort">
            <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
        </port>
        <port binding="tns:ActionMapManagerEndpointBinding" name="ActionMapManagerEndpointPort">
            <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
        </port>
        <port binding="tns:DataStoreManagerEndpointBinding" name="DataStoreManagerEndpointPort">
            <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
        </port>
        <port binding="tns:DiscoveryManagerEndpointBinding" name="DiscoveryManagerEndpointPort">
            <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
        </port>
        <port binding="tns:FolderManagerEndpointBinding" name="FolderManagerEndpointPort">
            <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
        </port>
        <port binding="tns:RuleManagerEndpointBinding" name="RuleManagerEndpointPort">
            <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
        </port>
        <port binding="tns:SessionManagerEndpointBinding" name="SessionManagerEndpointPort">
            <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
        </port>
    </service>
</definitions>
```

Annex C Rules to English Pseudo-Prose Stylesheet

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="html"/>
    <!-- ====================================================================== -->
    <!-- Top-level stylesheet template for rules -->
    <xsl:template match="Rule">
        <html>
            <head>
                <title>Rule</title>
            </head>
            <body>
                <h1>Rule Description</h1>
                <table align="center" width="90%" bgColor="#ffffe0" border="1">
                    <tbody>
                        <tr><td><tt><xsl:apply-templates/></tt></td></tr>
                    </tbody>
                </table>
                <hrule/>
                <h1>Rule Structure</h1>
                <table align="center" width="90%" bgColor="#ffffe0" border="1">
                    <tbody>
                        <tr><td><b><tt><xsl:apply-templates mode="tree"/></tt></b></td></tr>
                    </tbody>
                </table>
            </body>
        </html>
    </xsl:template>
    <xsl:template match="RootPredicate">
        <xsl:value-of select="concat('Check for ',@classLabel,' objects ',@objLabel,' that ')"/>
        <xsl:apply-templates/>
    </xsl:template>
    <!-- ====================================================================== -->
    <!-- Templates for tree structure display rules -->
    <xsl:template match="*" mode="tree">
        <a name="#">
            <xsl:attribute name="title">
                <xsl:apply-templates select="."/>
            </xsl:attribute>
            <xsl:value-of select="name(.)"/>
        </a>
        <div style="position: relative; margin-left: 16px;">
            <xsl:apply-templates mode="tree"/>
        </div>
    </xsl:template>
    <!-- ====================================================================== -->
    <!-- Templates for values in rules -->
    <xsl:template match="StaticValue">
        <xsl:value-of select="@value"/>
    </xsl:template>
    <xsl:template match="DynamicValue">
        <xsl:choose>
            <xsl:when test="@objRef!=''">
                <xsl:value-of select="concat(@classRef,':',@objRef,'.',@propName)"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="concat(@classRef,'.',@propName)"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:template>
    <xsl:template match="BuiltinFnValue">
        <xsl:value-of select="concat(@fnName,'(')"/>
        <xsl:apply-templates select="./*[1]"/>
```

```
        <xsl:for-each select="./*[position()!=1]">
            <xsl:value-of select="string(',')"/>
            <xsl:apply-templates select="."/>
        </xsl:for-each>
        <xsl:value-of select="string(')')"/>
    </xsl:template>
    <xsl:template match="AggregateValue">
        <xsl:value-of select="concat(@fnName,'(')"/>
        <xsl:apply-templates select="./*[1]"/>
        <xsl:for-each select="./*[position()!=1 and position()!=last()]">
            <xsl:value-of select="string(',')"/>
            <xsl:apply-templates select="."/>
        </xsl:for-each>
        <xsl:value-of select="concat(') over all ',@classLabel,' objects ',@objLabel,' for which ')"/>
        <xsl:apply-templates select="./*[last()]"/>
    </xsl:template>
    <!-- ========================================================================= -->
    <!-- Templates for relations in rules -->
    <xsl:template match="EqualsRelation">
        <xsl:text>equals</xsl:text>
    </xsl:template>
    <xsl:template match="NotEqualsRelation">
        <xsl:text>does not equal</xsl:text>
    </xsl:template>
    <xsl:template match="LessRelation">
        <xsl:text>is less than</xsl:text>
    </xsl:template>
    <xsl:template match="GreaterEqualsRelation">
        <xsl:text>is greater than or equal to</xsl:text>
    </xsl:template>
    <xsl:template match="ContainsRelation">
        <xsl:text>contains</xsl:text>
    </xsl:template>
    <xsl:template match="SpatialEqualsRelation">
        <xsl:text>equals</xsl:text>
    </xsl:template>
    <xsl:template match="SpatialDisjointRelation">
        <xsl:text>is disjoint from</xsl:text>
    </xsl:template>
    <xsl:template match="SpatialIntersectsRelation">
        <xsl:text>intersects</xsl:text>
    </xsl:template>
    <xsl:template match="SpatialTouchesRelation">
        <xsl:text>touches</xsl:text>
    </xsl:template>
    <xsl:template match="SpatialOverlapsRelation">
        <xsl:text>overlaps</xsl:text>
    </xsl:template>
    <xsl:template match="SpatialCrossesRelation">
        <xsl:text>crosses</xsl:text>
    </xsl:template>
    <xsl:template match="SpatialWithinRelation">
        <xsl:text>is contained within</xsl:text>
    </xsl:template>
    <xsl:template match="SpatialContainsRelation">
        <xsl:text>contains</xsl:text>
    </xsl:template>
    <xsl:template match="SpatialWithinDistRelation">
        <xsl:value-of select="concat('is within a distance of ',@distance,' of')"/>
    </xsl:template>
    <xsl:template match="SpatialBeyondRelation">
        <xsl:value-of select="concat('is farther than ',@distance,' from')"/>
    </xsl:template>
    <!-- ========================================================================= -->
    <!-- Templates for predicates in rules -->
    <xsl:template match="RelationalPredicate">
        <xsl:apply-templates select="./*[1]"/>
        <xsl:value-of select="string(' ')"/>
```

```
    <xsl:apply-templates select="./*[2]"/>
    <xsl:value-of select="string(' ')"/>
    <xsl:apply-templates select="./*[3]"/>
</xsl:template>
<xsl:template match="ReferentialPredicate">
    <xsl:choose>
        <xsl:when test="@objRef1!="">
            <xsl:value-of select="concat(@classRef1,':',@objRef1)"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="@classRef1"/>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:value-of select="string(' refers to ')"/>
    <xsl:choose>
        <xsl:when test="@objRef2!="">
            <xsl:value-of select="concat(@classRef2,':',@objRef2)"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="@classRef2"/>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:value-of select="concat(' via ',@refName)"/>
</xsl:template>
<xsl:template match="RangePredicate">
    <xsl:apply-templates select="./*[1]"/>
    <xsl:text> is in the range </xsl:text>
    <xsl:choose>
        <xsl:when test="@minInclusive='true'"><xsl:value-of select="string('[')"/></xsl:when>
        <xsl:otherwise><xsl:value-of select="string('(')"/></xsl:otherwise>
    </xsl:choose>
    <xsl:apply-templates select="./*[2]"/>
    <xsl:text>,</xsl:text>
    <xsl:apply-templates select="./*[3]"/>
    <xsl:choose>
        <xsl:when test="@minInclusive='true'"><xsl:value-of select="string(']')"/></xsl:when>
        <xsl:otherwise><xsl:value-of select="string(')')"/></xsl:otherwise>
    </xsl:choose>
</xsl:template>
<xsl:template match="AndPredicate">
    <xsl:apply-templates select="./*[1]"/>
    <xsl:for-each select="./*[position()>1]">
        <xsl:value-of select="string(' and ')"/>
        <xsl:apply-templates select="."/>
    </xsl:for-each>
</xsl:template>
<xsl:template match="OrPredicate">
    <xsl:apply-templates select="./*[1]"/>
    <xsl:for-each select="./*[position()>1]">
        <xsl:value-of select="string(' or ')"/>
        <xsl:apply-templates select="."/>
    </xsl:for-each>
</xsl:template>
<xsl:template match="ConditionalPredicate">
    <xsl:value-of select="string('if ')"/>
    <xsl:apply-templates select="./*[1]"/>
    <xsl:value-of select="string(' then ')"/>
    <xsl:apply-templates select="./*[2]"/>
    <xsl:if test="count(./*)>2">
        <xsl:value-of select="string(' else ')"/>
        <xsl:apply-templates select="./*[3]"/>
    </xsl:if>
</xsl:template>
<xsl:template match="ExistsPredicate">
    <xsl:choose>
        <xsl:when test="@n=1">there is </xsl:when>
        <xsl:otherwise>there are </xsl:otherwise>
    </xsl:choose>
```

```
        <xsl:value-of select="concat(@qualifier,' ',@n,' ',@classLabel)"/>
        <xsl:choose>
            <xsl:when test="@n=1"> object </xsl:when>
            <xsl:otherwise> objects </xsl:otherwise>
        </xsl:choose>
        <xsl:value-of select="concat(@objLabel,' for which ')"/>
        <xsl:apply-templates/>
    </xsl:template>
</xsl:stylesheet>
```