Open Geospatial Consortium Inc.

Date: 07.05.2006

Reference number of this OpenGIS® Project Document: **OGC 06-002r1**

Version: 0.5

Category: OGC® Discussion Paper

Editor: Joshua Lieberman

Geospatial Semantic Web Interoperability Experiment Report

Warning

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Discussion Paper
Document stage:	Approved for Public Release
Document language:	English

Contents

i. Preface

This document describes the results of an interoperability experiment addressing an issue(s) relating to the application of an OGC specification.

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained on the OGC website.

This is an OGC Discussion Paper for review bythe community. It is a working draft document and may be updated, replaced by other documents at any time. This document reflects is work in progress and is not an official position of the OGC membership.

This document was developed by the GSW.IE group as part of the OGC Interoperability Program initiative: Geospatial Semantic Web Interoperability Experiment (GSW.IE). The authors of this document are GSW.IE members.

ii. Submitting organizations

This Interoperability Program Report is being submitted to the OGC Interoperability Program by the following organizations:

Northrup Grumman Information Technology / TASC

BBN Technology

National Geospatial Intelligence Agency

Institut für Geoinformatik, Universität Münster

iii. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

CONTACT	COMPANY	ADDRESS	PHONE	EMAIL
Joshua Lieberman	Traverse Technologies	72 Maple Street, West Newbury, MA	+1 (617) 395-7766	jlieberman@traversetechnologies.com
Todd Pehle	NGIT / TASC	1010 Market Street, Suite 1740, St. Louis, MO 63101	+1 (314) 259-7841	todd.pehle@ngc.com
Chuck Morris	NGIT / TASC	1010 Market Street, Suite 1740, St. Louis, MO 63101	+1 (314) 259-7857	chuck.morris@ngc.com
Dave Kolas	BBN	1300 North 17th Street, Suite	+1 (703) 284-	dkolas@bbn.com

	Technologies	400 Arlington, VA 22209	1200	
Mike Dean	BBN Technologies	1300 North 17th Street, Suite 400 Arlington, VA 22209	+1 (703) 284-1200	mdean@bbn.com
Michael Lutz	IfGi	Institut für Geoinformatik Westfälische Wilhelms-Universität Robert-Koch-Str. 26-28 D-48149 Münster	+49 (0) 251 83-34707	m.lutz@uni-muenster.de
Florian Probst	IfGi	Institut für Geoinformatik Westfälische Wilhelms-Universität Robert-Koch-Str. 26-28 D-48149 Münster	+49 (0) 251 83-30058	f.probst@uni-muenster.de
Eva Klien	IfGi	Institut für Geoinformatik Westfälische Wilhelms-Universität Robert-Koch-Str. 26-28 D-48149 Münster	+49 (0) 251 83-39764	klien@uni-muenster.de

iv. Revision history

Date	Release	Author	Paragraph modified	Description
January 23, 2006	0.0	Josh Lieberman		First complete draft of document
May 5, 2006	0.5	Josh Lieberman		Responses to TASC comments
Aug. 21,2006		Carl Reed and Greg Buehler	Numerous	Prepare for posting to OGC public web site.

v. Future Work

The work reported in this document is a preliminary effort to extend existing OGC services, encodings, and architectures with Semantic Web technologies. Almost all results, therefore, are preliminary products that will benefit from further development, testing, piloting, and deployment. In particular, it is envisioned that ontologies refined from this work, as well as service profiles will be the subject of future OGC public documents and OGC compliant implementations.

Foreword

This document describes the results of an experiment applying Semantic Web technologies to implement and extend OGC specifications.

This report is informative, and is not a normative (draft or final) OGC Implementation Specification. This report thus does not include any compliance clauses or other specification-specific information.

This report often uses the terminology defined in the OGC Abstract Specification.

Introduction

The GSW Interoperability Experiment brought together a number of threads in semantics, Web technology, and geospatial reasoning. Participation in the experiment was solicited both from a wide spectrum of OGC members, and from other relevant organizations such as W3C, universities, and commercial semantic research groups.

The principal areas of compilation, problem solving, and implementation for this experiment were:

- ✓ Development of geospatial, domain, and other ontologies covering the knowledge and operations domains of the demonstration use case
- ✓ Development of a reference Web architecture for exchanging information with formal semantics and processing information queries.
- ✓ Choice and refinement of one or more query languages and predicates for expressing geospatial queries.
- ✓ Development of interfaces and profiles that support the interaction of components in the Web architecture and can be used to extend existing OGC Web Services components.
- ✓ Implementation of components and information stores sufficient to carry out the experimental demonstration.

An important challenge for this experiment and for follow-on work has involved bridging a gap between the technology of distributed computing and the technology of semantic processing. The initial architecture for this work emphasized the discovery and consumption of distributed geospatial services to gather together semantic artifacts for reasoning within an application. Later versions will begin to explore the distribution of machine reasoning and semantic processing among remote service components, as ongoing work here and elsewhere makes this feasible.

1 References

OWL
W3C Ontology Web Language, available online: http://www.w3.org/2004/OWL/

SA
Topic 12, Service Architecture. OGC Abstract Specification, available online: http://www.opengis.org/public/abstract/01-112.pdf

SESAME
SESAME Users Manual, available online: http://www.openrdf.org/doc/sesame/users/index.html

SESAME_HTTP
SESAME http interface, available online: http://www.openrdf.org/doc/sesame/users/ch08.html#d0e2475

SPARQL
SPARQL query specification, available online: http://www.w3.org/TR/rdf-sparql-query/.
SPARQL protocol specification, available online: http://www.w3.org/TR/rdf-sparql-protocol/

SWRL
W3C Semantic Web Rules Language, submission available online: http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

Catalogue
OpenGIS® Catalogue Service Interfaces Implementation Specification Version 2.0.1, http://portal.opengeospatial.org/files/?artifact_id=5929&version=2

WMS
OpenGIS® Web Map Service Interfaces Implementation Specification Version 1.1.1, available online: http://portal.opengeospatial.org/files/?artifact_id=1081&version=1&format=pdf

WFS
OpenGIS® Web Feature Service Interfaces Implementation Specification Version 1.1, available online: https://portal.opengeospatial.org/files/?artifact_id=8339

2 Relationship to Other Activities

The GSW.IE sought particularly to make connections between two lines of OGC activity:

- Geospatial encodings, service specifications and architectures which are potentially applicable to a broad spectrum of geospatial data and applications;

- Restriction to particular profiles and vocabularies for those specifications in order to make them meaningful and implementable within a particular information community.

More specifically, the experiment addressed the use of the Web Feature Service and Catalogue Service / Web service specifications, as well as the GML and Filter Encoding specifications. Information communities considered included geospatial intelligence and aeronautical navigation.

3 Problem Statement

This interoperability experiment addressed several important steps towards the development of a Geospatial Semantic Web (GSW), where discovery, query, and consumption of geospatial content are based on formal semantic specification.

What problem in general does the Geospatial Semantic Web solve? The World Wide Web has made a world of online content globally accessible with one mouse click. The Geospatial Web has extended this capability to an immense variety of geospatial information and concepts. The farther that information and services travel from their origin, though, the less likely that remote users will also have access to the local context in which to find, evaluate, and interpret such resources. The success of syntactically interoperable Web services connections has created significant semantic gaps in what can be utilized. Both users and their client software may lack the tools to work with the diverse information which Geospatial Web standards have made nominally obtainable.

The Semantic Web seeks to make the meaning as accessible as the material, by enabling connections–which are both logical and (machine) actionable–between concepts which a user presently understands and those which may be new and foreign. The Geospatial Semantic Web extends this capability to both content and concepts that are specifically spatial, temporal, and geographic in nature, giving both people and machines true access to a wider range of knowledge.

Many pieces of the GSW puzzle have already been worked extensively in the last several years. Our experiment sought specifically to test the extension and implementation of WFS /FE services where these elements may be assembled to implement a distributed end-to-end geospatial query/discovery use case. Significant elements of this experiment have included:

- Development / encoding of formal and connected geospatial ontologies, including feature type descriptions;
- Geospatial service interfaces which can provide service information formulated in the OWL-S semantic expression language referencing those ontologies;
- Geospatial service interfaces which can operate on requests for ontologically expressed service and content descriptions;

- Semantic query language processing interfaces for WFS /FE or other services which operate on combinations of the above ontologies, service information and queries;
- Processes for discovering semantically expressed geospatial information and services;
- Implemented WFS / FE and other components for use in processing geospatial queries

Two test demonstrations were developed by the GSW IE participants of an end-to-end semantic geospatial query case, where WFS /FE data and services relevant to the query are discovered through knowledgebase reasoning. Parts of this report include material for developing new semantic profiles of the WFS / FE and CS/W specifications which reflect the results of the experiment. This is a small but significant contribution to advancement of the Geospatial Semantic Web.

3.1 Use Case

The specific query we set out to answer in this experiment was:

> *"Which airports in a specified area of the world are capable of receiving a C-5 cargo plane?"*

Given a user able to construct such a query, a known information service with known request syntax and data that directly answers this question, the query is of course trivial. In our specific use case, the user may be familiar with the general concepts of the query, but not know the specific terms to use in a successful query. The specific services with relevant information may not be known in advance, and there may not be an existing dataset attribute which explicitly indicates which airports have the specific capability requested.

In the real world of the present, such a user would typically hunt around for information sounding syntactically familiar, then turn to searching for other people to help her with unfamiliar datasets and terminology, to whom she would laboriously explain the query, searching for terms familiar to both of them.

Our plan was to develop semantic forms of the critical information for this search process, including Web Feature Service information, GML Feature Schemas, and Spatial / Non-spatial Filter Encodings using OWL / OWL-S. This information could then be used with semantic processing technologies to implement the above query use case in a distributed computing scenario.

3.2 Semantic Interoperability

There is no general agreement on what exactly constitutes semantic interoperability, although lack of it it is commonly held to be a problem. Some elements of semantic interoperability include:

- Top of the interoperability stack for distributed computing
- Shared concepts
- Formal, common knowledge representation, usually in graph form, generically termed "ontologies"
- Actionable (i.e. machine-process-able) relationships between similar / connected concepts using inference, reasoning, pattern matching, rule processing, etc.
- Shared models for distributed processes, such as interface contracts and virtual machines.

3.3 Semantic processing roles in a distributed computing architecture

The GSW.IE was specifically interested in the application of semantic technology to OWS service architectures. In this type of environment as illustrated in Figure 1, roles for semantic processing include:

- Cross-domain resource *discovery: Services are published through a well-known trader. A client discovers relevant served resources through the trader, which tells it how to bind to the server. In the case where the published resources are represented in a different vocabulary or knowledge domain from that of the client user, semantic processing enables such heterogenous resources to be discovered where text search alone is not sufficient.*

- Heterogeneous resource *query:Clients bind to and query services for relevant resources. Semantic query establishes connections from the query knowledge domain to multiple heterogeneous resource domains across multiple heterogeneous service interfaces.*

- Resource *translation* from one domain to another*: Basic publish-find-bind architectures provide for discovery and retrieval of distributed resources. For actual utilization of heterogeneous resources, those resources may require an additional step of translation through a broker function from their native knowledge domain into a knowledge domain accessible to the retrieving client and user.*

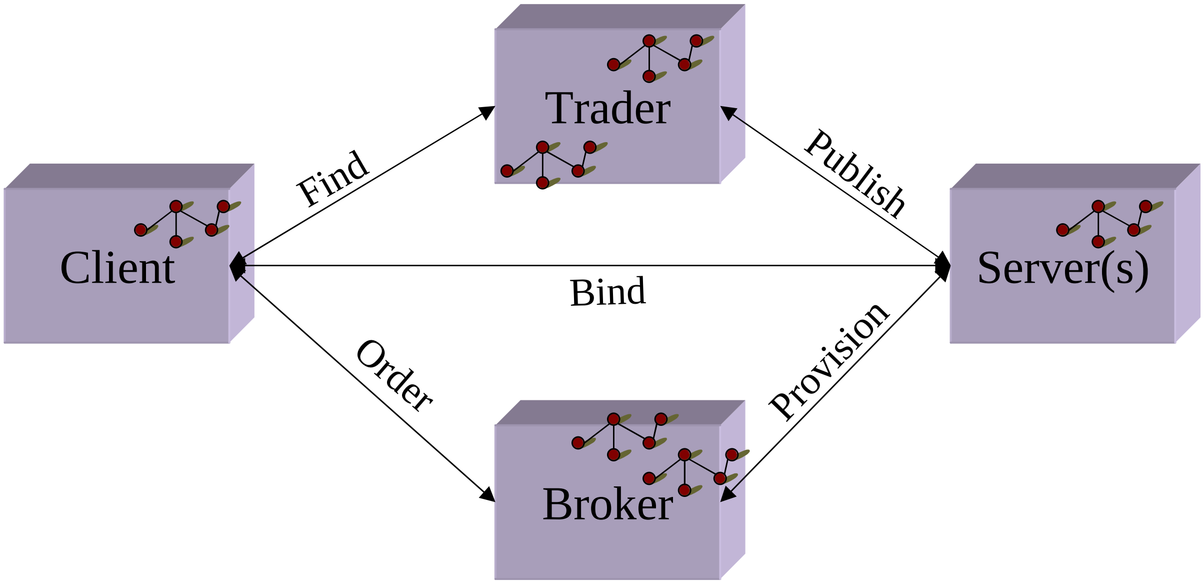


Figure 1. Service-oriented Architecture with Broker role extension.

Each of the interactions shown may potentially be enhanced with formal semantics and semantic processing.

4 Experiment Design and Activities

4.1 Overview

The GSW.IE addressed some of the following experiments around the use of a semantically enabled Geospatial Web:

- Experiment #1: Construct a feature dataset ontology using OWL in order to describe 5-6 sample feature collections relevant to the GEOINT use case.
- Experiment #2: Construct an OWS service description ontology using OWL-S in order to describe WFS services.
- Experiment #3: Translate realistic GEOINT request (see the Use Case above) into a semantic query language encoding and GEOINT query domain ontology.
- Experiment #4: Perform an end-to-end geospatial semantic query in a non-distributed environment using the results of Experiment #1 - Experiment #3; the result of this query process will be WFS service requests for information which satisfy the query.
- Experiment #5: Implement and test an OWS capabilities type and/or interface which returns service / content descriptions such as in Experiment #1 and Experiment #2 from WFS service instances.
- Experiment #6: Perform an end-to-end geospatial semantic query as in Experiment #4 but involving remote requests for metadata as in Experiment #5.

The goal of this sequence of experiments was a step-wise increase in the complexity of the architecture being tested, beginning with a largely centralized use of ontologies and knowledge bases, proceeding to more distributed forms of processing.

4.2 Multiple ontology development

An essential part of the experimental design involved use of multiple ontologies representing domains of knowledge or communities of information which remain separate but which may be linked and traversed by the geo-semantic query process. Figure 3 shows the conceptual layout of ontology “types” which were defined and developed during the experiment.

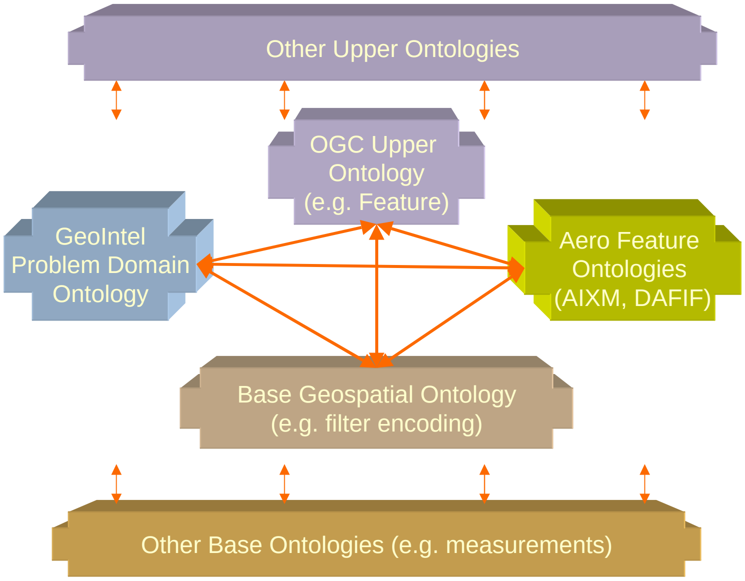

Other Upper Ontologies
OGC Upper Ontology (e.g. Feature)
GeoIntel Problem Domain Ontology
Aero Feature Ontologies (AIXM, DAFIF)
Base Geospatial Ontology (e.g. filter encoding)
Other Base Ontologies (e.g. measurements)

Figure 3. Conceptual design of multiple ontologies relevant to the experiment use case, showing their relationship to one another and to the semantic query process.

Just as the communities they represent remain separate but overlapping, so these ontologies are expected to maintain separate identities but be linked (e.g. via subsumption or mapping) into a knowledgebase (either physical or virtual) capable of providing answers to cross-domain queries.

Domain ontologies

Any ontology representing specialized knowledge may be termed a *domain* ontology. We make a practical distinction here between two types of GSW domain ontologies

Problem ontology – ontology for a domain in which questions are asked or applications are created, also termed *query*, *application,* or *public* ontology.In Figure 3, this type of ontology is represented by the GeoIntel ontology.

Feature ontology – ontology for a domain in which source geoinformation exists, also termed *data*, *source*, or *private* ontology.In Figure 3, this type is represented by multiple aero feature ontologies

Upper versus base ontologies

Base Ontology - A commonly used and possibly standardized ontology to represent a particular domain or area which is of general interest. An example of this would be a particular standard vocabulary, such as for units of measure

Upper Ontology - An ontology designed to link other ontologies by providing a more abstract definition of concepts that appear in the other ontologies. This creates a tree in which other ontology concepts become subconcepts of the elements of the upper ontology. Concepts such as Feature or SpatiallyRelated might belong to an upper ontology.

These two types of ontologies are not mutually exclusive. A particular ontology may have greater or lesser characteristics of each type. However, they both tend to be ways that ontologies are linked to one another, and there are times when it makes sense to choose between the two approaches. Upper ontologies tend to represent the universe at the most abstract level; They tend to be broad, but may not be deep enough to meet the needs of an application, and it can be hard to come to agreement about what should or should not be in them. This is not to say that it is easy to come to consensus about how to represent things within one specific domain (geospatial data, for example), just that there could be a higher probability of success here.

The use of common base ontologies makes software specialization significantly less challenging. If there were a standard base geospatial ontology that defined representations for geometries, mapping software could make use of data from any ontology as long as it linked to the base geospatial ontology for describing its geometries. The Friend of a Friend (FOAF), contact information ontology is another example of this. Software designed to seek out contact information could handle data represented primarily in any ontology if it referenced foaf where appropriate. None of this is to say that we exclude the use of upper ontologies, but we tend to use them for different purposes. For example, Classification Upper Ontologies are particularly useful for human understanding of the meaning of an ontology.

Ontology Language

For a number of reasons, including experience of the participants, availability of software tools, and affinity for use in distributed systems, ontologies were represented for the purpose of the experiment in the RDF and RDFS offshoot OWL DL (i.e. OWL builds on RDF and is represented in it). Other information elements were also represented in RDF, such as rules (SWRL).

There are choices as well of languages to be used for querying ontologies. The focus of the IE was the use of SeRQL since it is currently supported by the SESAME platform, but preliminary use was also made of SPARQL, an emerging W3C standard which is similar to SeRQL.

4.3 Information sets

While a distributed computing investigation must operate on a number of levels, the ultimate object of this activity is clearly information interoperability. A number of the types and sets of information which were created and/or operated upon in the GSW IE are described below.

4.3.1 OWL and Ontologies

An ontology is itself a conceptual entity which may be carried in any number of expressions (e.g. vocabularies) and encodings (e.g. textual forms). Given experimental priorities of available software technology and applicability to Web / distributed computing paradigms, OWL DL [OWL] was chosen as the most appropriate language for expressing ontologies created and/or utilized in the experiment, encoded generally in the form of RDF XML for portability.

OWL DL alone proved not to be sufficiently expressive for some of the geospatial and cross-domain knowledge needed to process the experimental query efficiently. Additional processing rules were therefore devised by the participants to fill this gap. In the case of the BBN semantic middleware, rules were added to the BBN knowledgebase by making use of the developing SWRL specification. SWRL combines OWL with a means of expressing logical evaluation rules. A sample rule and matching expression are shown below:

```
IF
 ?airport  is an Airport
 ?airport comprisesRunway ?runway
 ?runway is a Runway
 ?runway runwayLength ?length
 ?length  > 6000
THEN
 ?airport is a C5CapableAirport

<ruleml:imp>
  <ruleml:_body>
   <swrlx:classAtom>
    <owlx:Class owlx:name="&airport-ont;#Airport"/>
    <ruleml:var>airport</ruleml:var>
   </swrlx:classAtom>
   <swrlx:individualPropertyAtom swrlx:property="&airport-
ont;#comprisesRunway">
    <ruleml:var>airport</ruleml:var>
    <ruleml:var>runway</ruleml:var>
   </swrlx:individualPropertyAtom>
   <swrlx:classAtom>
    <owlx:Class owlx:name="&airport-ont;#Runway"/>
    <ruleml:var>runway</ruleml:var>
   </swrlx:classAtom>
   <swrlx:datavaluedPropertyAtom swrlx:property="&airport-ont;#runwayLength">
    <ruleml:var>runway</ruleml:var>
    <ruleml:var>length</ruleml:var>
   </swrlx:datavaluedPropertyAtom>
   <swrlx:builtinAtom swrlx:builtin="&swrlb;#greaterThanOrEqual">
    <ruleml:var>length</ruleml:var>
    <owlx:DataValue owlx:datatype="&xsd;int">6000</owlx:DataValue>
   </swrlx:builtinAtom>
   <!-- additional constraints -->
  </ruleml:_body>
  <ruleml:_head>
   <swrlx:classAtom>
    <owlx:Class owlx:name="&this;#C5CapableAirport"/>
    <ruleml:var>airport</ruleml:var>
   </swrlx:classAtom>
  </ruleml:_head>
 </ruleml:imp>
```

4.3.1.1 Feature representation

There are a number of knowledge representation issues which were dealt with during the course of this experiment in order to construct feature domain ontologies which are both accurate and useful. Issues are described here under two main headings.

Representing the representation

Ontologies are normally used to represent what is known about a real domain and are most useful when such a direct representation is asserted. Representing geographic knowledge is complicated by the fact that this knowledge already has a level of geometric indirection, as geographic features are represented inexactly by geometric entities such as points, lines, and polygons. The ontologist is then faced with a choice whether to derive classes from geographic entities (e.g. “cities”), feature entities (“city point features”) or both. The latter is more accessible in terms of geospatial reasoning and query (e.g. “within 50 km of city”), but the former is more accessible to reasoning on geopolitical concepts (e.g. “neighborhood belongs to city”).

In order to avoid myriad complications which attend this sort of multiple inheritance, the choice was generally made in this experiment to represent geographic entities directly and express feature “aspects” of those entities by means of property relationships (e.g. “hasLocation”). This was generally an illustration that there are no globally “true” or “correct” answers in ontology development (at least accessible to human ontologists), only useful ones for the perceived range of ontology application.

Shared vocabulary

There are means to directly and semi-automatically translate many data models, particularly if they are expressed in XML Schema or UML, directly into an ontology language such as OWL. Web-accessible archives contain many of these generated ontologies. The challenge is that many entity properties from these data models end up in the ontology as string or other literal types, which cannot generally be compared with each other in order to bridge gaps between disparate ontologies. Experiment participants spent some time refining the “working set” ontologies to make greater use of common property concepts from “base” ontologies such as one for measurements. This approach, although requiring greater upfront effort, generally allows the linking of multiple ontologies into a common query / reasoning framework to proceed more smoothly and easily.

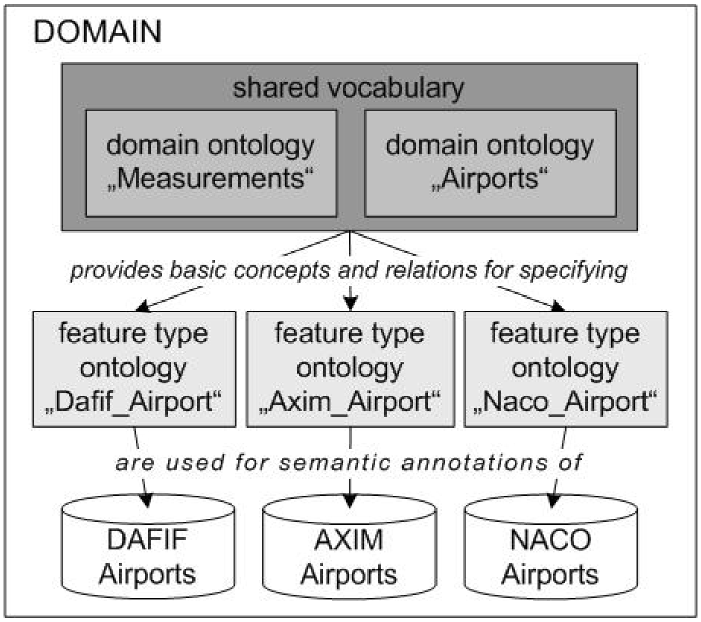

DOMAIN
shared vocabulary
domain ontology „Measurements“
domain ontology „Airports“
provides basic concepts and relations for specifying
feature type ontology „Dafif_Airport“
feature type ontology „Axim_Airport“
feature type ontology „Naco_Airport“
are used for semantic annotations of
DAFIF Airports
AXIM Airports
NACO Airports

Figure 8. Illustration of feature representation

4.3.1.2 Multiple distributed feature sources

In the case of multiple, distributed feature data sources, there is an additional layer of representation, both as an (instance) ontology, and as a Web service" represented in this case by an OWL-S description.

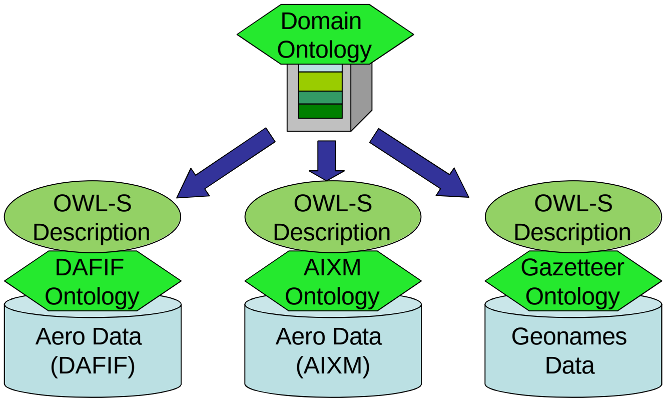


Figure 8. Illustration of feature source distribution

4.3.2 Query languages

Example SeRQL statement for experiment test query

```
SELECT airport, city FROM
   {airport}   rdf:type    {c5:C5CapableAirport};
        filter:satisfiesFilter {filter},
   {filter} rdf:type {filter:DWithin};
          filter:referenceGeometry {city};
          filter:radius {“50000"^^xsd:int},
   {city}   rdf:type {cities:City};
       filter:satisfiesFilter {cityFilter},
   {cityFilter} rdf:type {filter:PropertyIsLike};
            filter:property {"NAME"};
             filter:literal {"Saint Louis"}
using namespace
c5 = <http://gsw.projects.semwebcentral.org/2005/03/airport/c5#>,
filter = <http://gsw.projects.semwebcentral.org/2005/03/geofilter/filter-ont#>,
cities = <http://gsw.projects.semwebcentral.org/2005/03/airport/cities-ont#>
```

Query Distribution

A consequence of working with distributed and heterogeneous information sources is the convolution of each single query into a graph of multiple consequent queries as the initiatl query touches concepts which are represented in remote stores. The example in Figure 9 shows how the “C5 capable Airport” concept leads to a sequence of queries against multiple datasets representing multiple feature ontologies possible offered by multiple services. Each arrow in the diagram may represent multiple service queries which determine through description and typing information whether the features in a particular dataset are relevant to the initial query. Depending on the scale of a real distributed information system and the knowledge about them already held by the query processor, there may be tens or hundreds of consequent queries for a single initial query.

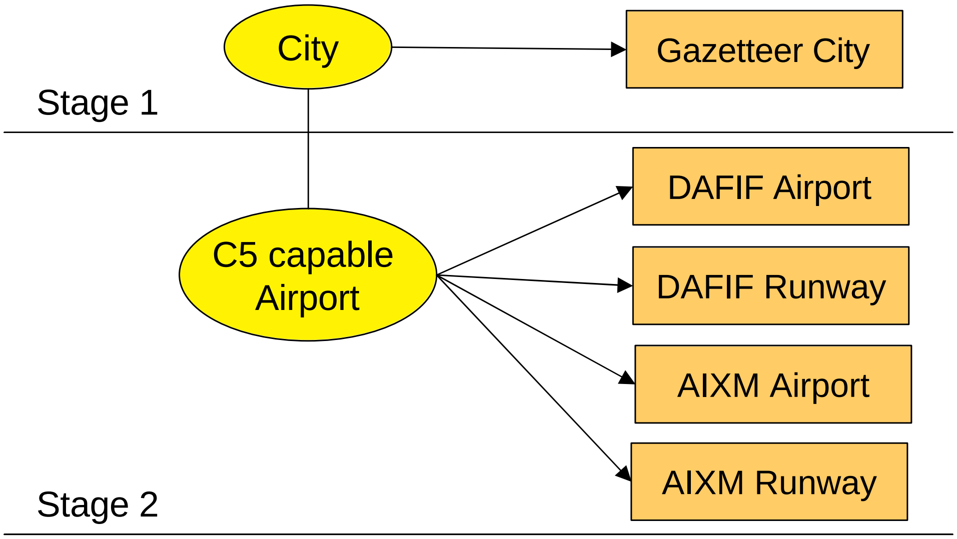


Figure 9. Query plan example: subqueries consequent on evaluation of the initial experiment query.

4.3.3   OWL-S service information

A critical element of the use of services in distributed computing environments is the service description. Whether such a description is available directly from a service, such as with OWS services, or from a 3rd party in the case of most WSDL descriptions, service information is the key both to discovering and consuming useful services in any service-oriented architecture.

Descriptions or “capabilities” for OWS services have generally consisted of custom XML documents incorporating some elements of standard metadata for geoinformation. Several efforts have been underway to specify more complete and actionable ontology-based descriptions in order to facilitate the implementation of “Semantic Web Services”. One effort in particular is OWL-S, which extends OWL to represent three aspects of a Web service (or any other service, really):

- Service profile – thematic information on the type, role, and context of a service
- Service grounding – syntactical information for service binding and consumption

- Service process – a conceptual or general model of the process by which a service relates inputs to outputs. In contrast to the notion of an opaque interface, it is not really practical to make use of a service or build a client for it without some external or public model of what the service does. This is the newest and most problematic part of a semantically enabled service description.

OWL-S also makes use of elements of WSDL [WSDL] in order to describe the syntax of binding to a particular service.

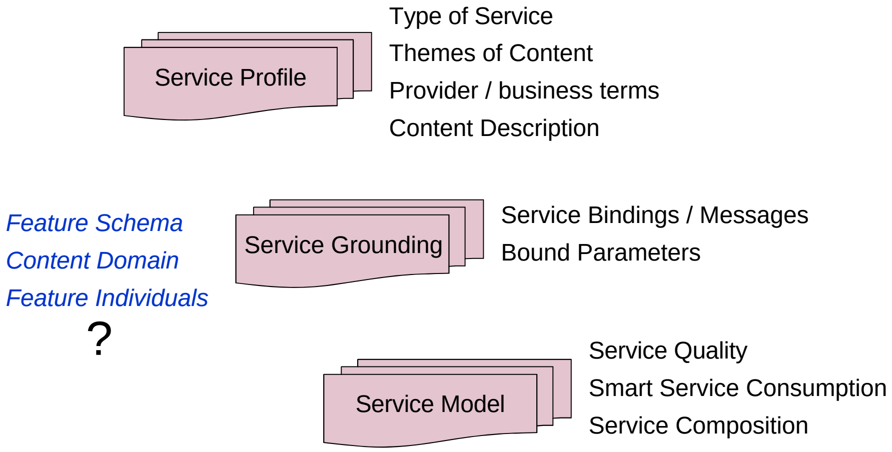


Figure n. Conceptual elements of an OWL-S service description

In principal, an OWL-S description can be utilized by a generic service client to bind effectively to that service, hence the concept of an “OWL-S Virtual Machine” which represents such a distributed service as a local process in an environment such as a knowledgebase. This approach is discussed further as an aspect of the BBN semantic middleware.

The following box shows a short example of an OWL-S description for a WFS serving DAFIF features

```
<service:Service rdf:ID="Demo3DafifService">
  <service:presents rdf:resource="#Demo3DafifProfile"/>
  <service:describedBy rdf:resource="#Demo3Dafif_GetFeature"/>
  <service:supports rdf:resource="#WsdlGrounding"/>
</service:Service>
<process:AtomicProcess rdf:ID="Demo3Dafif_GetFeature">
   <service:describes rdf:resource="#Demo3DafifService" />
   <process:hasInput rdf:resource="#Demo3Dafif_GetFeature_IN"/>
   <process:hasOutput rdf:resource="#Demo3Dafif_GetFeature_Dafif_ARPTS_OUT"/>
   <process:hasOutput rdf:resource="#Demo3Dafif_GetFeature_Dafif_RWYS_OUT"/>
</process:AtomicProcess>
<process:Input rdf:ID="Demo3Dafif_GetFeature_IN">
   <process:parameterType>&wfs;#GetFeature</process:parameterType>
```

```
</process:Input>
<process:Output rdf:ID="Demo3Dafif_GetFeature_Dafif_ARPTS_OUT">
  <process:parameterType>&dafif;#DAFIF_ARPTS</process:parameterType>
</process:Output>
<process:Output rdf:ID="Demo3Dafif_GetFeature_Dafif_RWYS_OUT">
  <process:parameterType >&dafif;#DAFIF_RWYS</process:parameterType>
</process:Output>
```

4.3.4 GML features / descriptions

Several feature datasets were used in the experiment. Three principal datasets served as the material for “feature ontologies” The challenge (and opportunity) they posed for semantic discovery and query was that each contained partial information in differently named fields about place features relevant to the experimental query.

FeatureType	ISO 19115 Metadata File	Keywords used in the ISO 19115 Metadata
Dafif_Airport	dafif_airportsV1.xml	Airport, **Aerodrome**, dafifairports.owl#Dafif_Airport
Dafif_Airport	dafif_airportsV2.xml	Airport, Dafif, Runway, dafifairportsV2.owl#Dafif_Airport
AXIM_Airport	axim_airports.xml	Airfields, **Aerodrome**, Transportation
Facilities	Transportation_facilities.xml	Transportation, Airport, Railwaystation, Underground, Busstop

Figure 9. Some comparisons of fields in various aeronav datasets used in the experiment

4.3.4.1 AIXM

AIXM stands for **A**eronautical **I**nformation E**X**change **M**odel and consists of both a data model and an exchange format for aeronav information. Various datasets have been developed after this model.

4.3.4.2 DAFIF

DAFIF stands for **D**igital **A**eronautical **F**light **I**nformation **F**ile, a public domain, global aviation database. Once a publically accessible dataset, it is no longer updated as a publicly available resource, due mainly to security concerns.

4.3.4.3 GEONAMES

GEONAMES is a database of worldwide geographic place names maintained by NGA, available both as datasets and from GNS – GEOnet Names Server, including where available both geographic and UTM point coordinates.

4.4 Semantic query stack

The initial semantic query stack targeted by the experiment is illustrated in Figure 2.

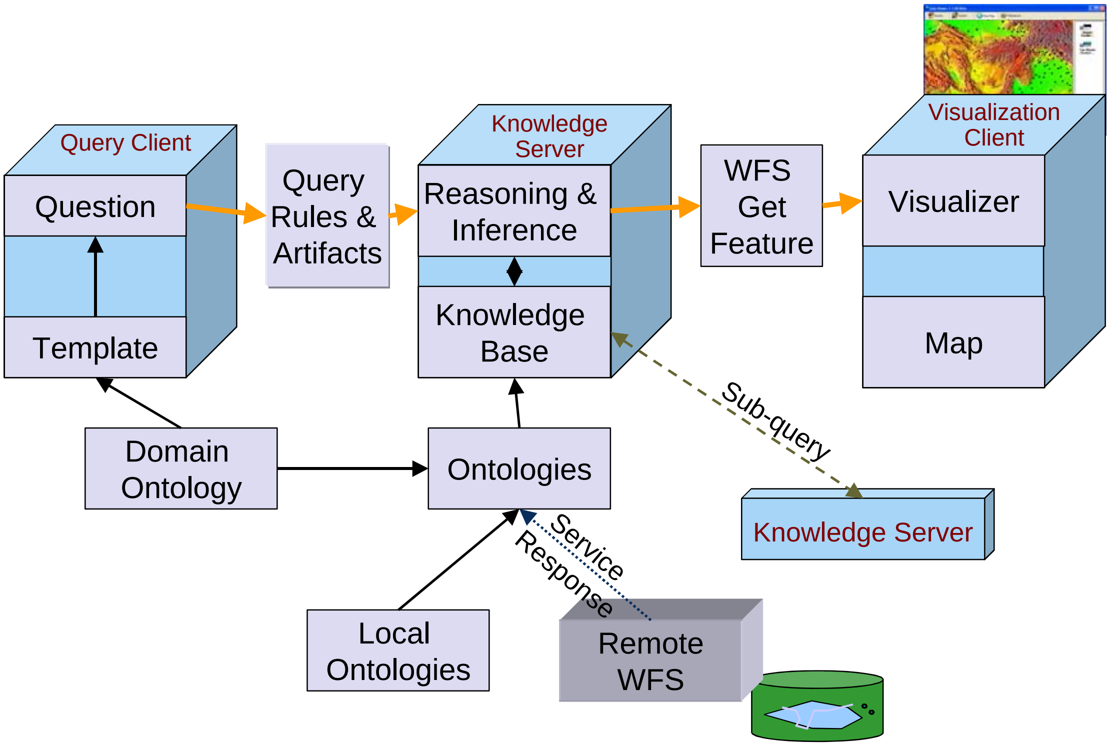


Figure 2. Conceptual semantic query stack, showing formulation of a domain query, reasoning and inference across multiple ontologies, and map visualization of the query result. Sub-queries shown by a dashed line were not within the scope of the present work.

The "Knowledge Server" component is shown only in the most general terms and was implemented in more than one way by different GSW.IE participants. For example, the TASC-BBN component fetched feature instances directly while the IfGi client performed this function.

4.5 Component interfaces

In order to implement a distributed computing system, it is necessary to define or choose interfaces by which the components communicate with each other. The experiment focused on two OGC service specifications and two similar "Knowledge Service" query interfaces. Forms of these interfaces were then implemented in different component and architectural configurations by the participants.

4.5.1 Web Feature Service (WFS)

The GSW.IE specifically looked at what it means to "semantically enable" a Web Feature Service. The appropriate level of enablement represents a balance between a view of the WFS as a primary source of data and the likelihood that the provider of feature data through a WFS is likely the party most knowledgeable about the features and most able to describe them fully. Four levels of enablement were considered, although not all were implemented in the experiment:

- WFS service description in OWL-S; this enables semantic discovery of WFS capabilities and/or use of OWL/S virtual machines to consume WFS services.

- WFS feature type description (e.g. DescribeFeature) in OWL; this is usually a necessary enhancement in order for the tightly coupled contents of WFS to be discoverable through semantic processing whether by the WFS component or elsewhere.

- WFS feature instances returned in OWL; this step is not generally strictly necessary in order to visualize features, for example, as other representations (e.g. GML) may suffice for this purpose; it may be needed, however, in order to perform domain translations or to determine whether particular features actually answer a query

- WFS support for an (additional) OWL / RDF query language such as SPARQL; in this case the graph matching, rule processing and other forms of semantic query processing would be accomplished behind the WFS service interface.

These forms of service interface enhancement are not necessarily achieved in an originating server component but may represent a proxy, façade, or translation layer somewhere else in the services architecture. Similar approaches were implemented during the course of the IE.

4.5.2 CS/W

A catalogue or registry interface is a central component feature of any service-oriented architecture. The experiment began with considering a catalogue as a means of discovering other semantically enhanced services and information. During the course of the experiment it gradually became clear that the OGC CS/W service interface could also be enhanced in various ways to provide a connection between OGC services and various forms of semantic graph manipulation technology. Several levels of semantic enablement were again considered and variously implemented:

- Existing catalog resources, such as WFS service instances or feature types, which have been "decorated" (e.g. as keywords) with searchable semantic concepts such as OWL class identifiers. This enables a non-enhanced CS/W client to use the concepts resulting from a separate semantic query or knowledgebase operation in a "normal" CS/W query.

- Addition of semantic catalogue resources, as for example records which represent OWL concepts, relationships, or subgraphs. At perhaps the lowest level, individual RDF triples could be registered as CS/W catalogue records. It is likely that some hierarchy of record types (e.g. triple- subgraph-graph)would prove most useful in this regard (c.f. classification-classification_node-classification_scheme in the ebRIM profile of CS/W).

- CS/W enhanced query to support semantic processing; in such a case the discovery-oriented aspects of knowledgebase processing and query would be invoked with CS/W operations, specifically GetRecord. Such an enhanced interface could be syntactically compliant with the CS/W specification, but involve query operators that make use of graph relationships in addition to more straightforward textual, spatial, or category matching.

- CS/W enhancement to support an OWL / RDF query language such as SPARQL; in this case the CS/W interface would be used as a rather thin veneer on a variety of knowledgebase processing operations. It is likely that additional operations would be required to support actions such as inferencing, in much the same way that non-semantically-enhanced CS/W implementations require additional operations if certain distributed catalogue administration functions are to be supported.

- CS/W brokering enhancements for semantic translation; while the OGC Catalogue 2.0 specification does include conceptual ordering operations, these have not really been explicitly defined in either the CS/W protocol or related information profiles. On the other hand, ontology languages such as OWL make less of an absolute distinction between resource collections, definitions, and individuals than some other information forms, so specific brokering operations may not be as important in order for a knowledgebase type of component implementing a CS/W interface to provide semantic translation capabilities.

4.5.3 SESAME HTTP

As part of its API, the SESAME open-source semantic processing framework [SESAME] provides a simple HTTP interface [SESAME_HTTP] with operations which allow a Web client to list the available graph repositories and execute queries in SeRQL against them. Operations available through this interface include:

- *ListRepositories*
- *EvaluateTableQuery*
- *EvaluateGraphQuery*
- *ExtractRDF*
- *UploadData*
- *UploadURL*
- *ClearRepository*
- *RemoveStatements*

As can be seen from this list, the SESAME http interface provides some capabilities for information about, query of, and transactions on RDF graph stores, but does not provide all the controls on semantic processing which are possible within the platform.

4.5.4 SPARQL

SPARQL [SPARQL] is a standard semantic query language emerging from the W3C. It is also combined with a specification for a relatively simple access protocol with HTTP and SOAP bindings. This access protocol has but one interface and one operation: *Query*. This operation simply provides the means of conveying the text of a SPARQL query to a SPARQL query processor and receiving the response. All of the information on what the client is actually requesting is contained within the SPARQL message itself, and so the interface carries no information / semantics on what is being requested, in contrast to other more method-rich interfaces.

4.6 Software selection and designs

This clause of the report describes components which were deployed by GSW IE participants.

4.6.1 BBN knowledge server

The BBN semantic middleware component is based upon the open-source processing platform SESAME and the reasoning engine JENA, as well as their own graph store, DAMLDB. The component architecture is shown in Figure 4.

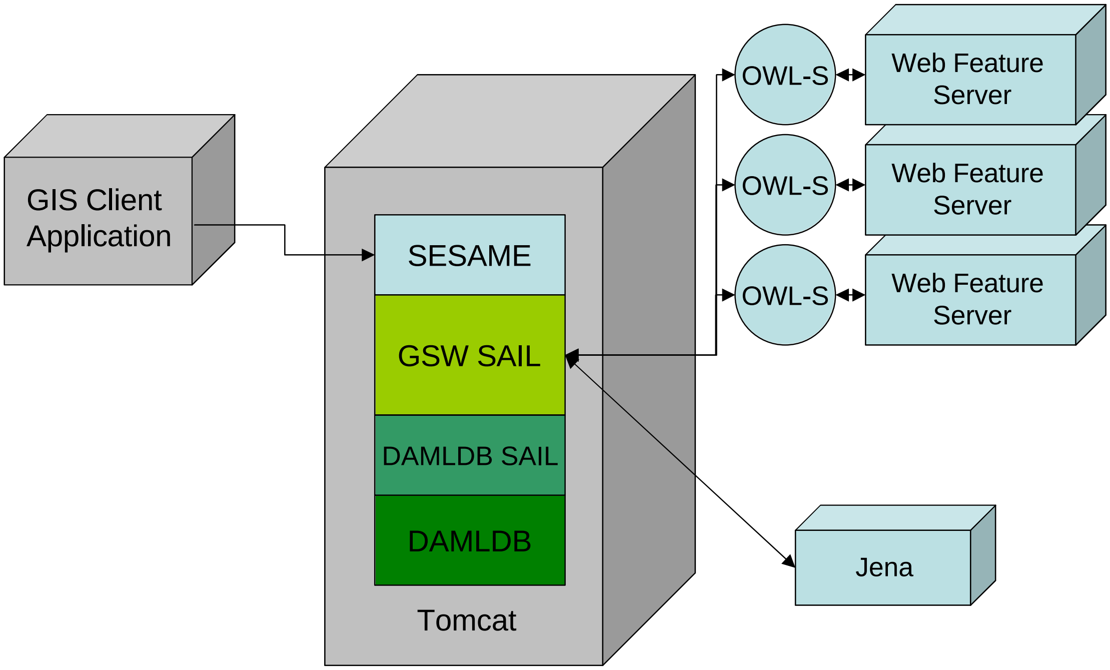


Figure 4. Component and sub-component architecture of BBN knowledge middleware

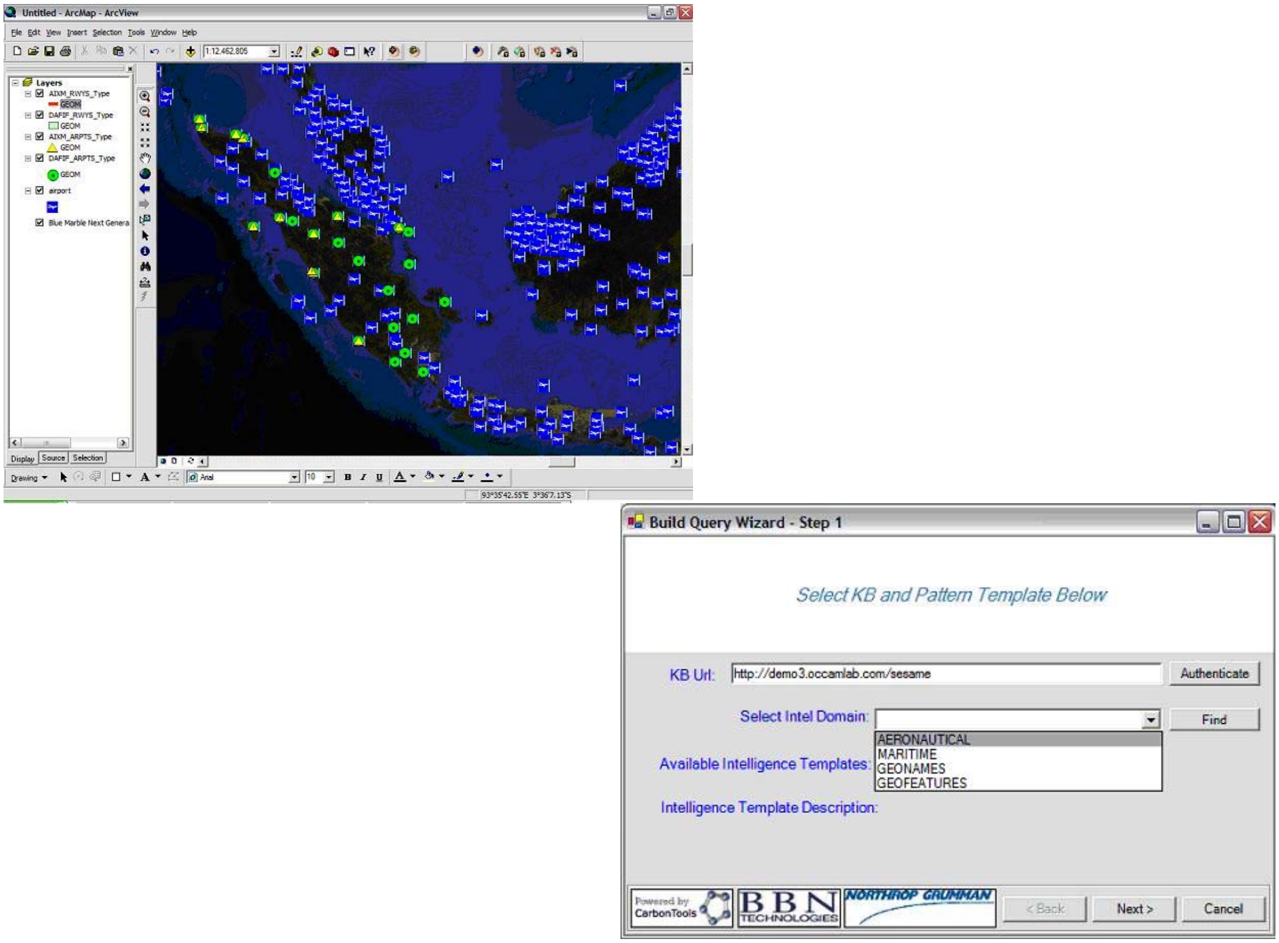
In addition, a custom GSW SAIL (SESAME **S**torage **A**nd **I**nference **L**ayer add-in) was developed for specialized geospatial processing, such as evaluation of geospatial rules, construction of OGC Filter query expressions and connection to remote Web Feature Servers through an OWL-S virtual machine.

4.6.2 NGT feature server

NGIT / TASC participants provided access to aeronavigation geodata (e.g. AIXM, DAFIF) through an open-source Web Feature Server component (Geoserver) hosted at their OccamLab facility.

4.6.3 NGT client

NGIT / TASC designed and developed a semantic query / map visualization client based on ArcMap and the Carbon Toolkit. This client component is able to make SeRQL queries to the BBN knowledge server as well as GetFeature queries to the WFS component.



Select KB and Pattern Template Below

KB Url: http://demo3.occamlab.com/sesame

Select Intel Domain:

Available Intelligence Templates: AERONAUTICAL, MARITIME, GEONAMES, GEOFEATURES

Intelligence Template Description:

Figure 5. Images of map viewer and query wizard screen from NGIT / TASC semantic map client

4.6.4 IfGi client

Participants from IfGi developed a browser-based client implementation which incorporated both the construction of a semantic query and processing of a knowledgebase to return concepts matching the submitted query.

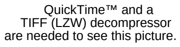


Figure 6. Browser screeen from IfGi semantic query client

4.6.5 IfGi catalog

IfGi deployed other OGC components as shown in the architecture diagram in Figure 7. Components included a CS/W catalog with “concept-decorated” metadata records, as well as an ontology-based reasoner component and Web Feature Server.

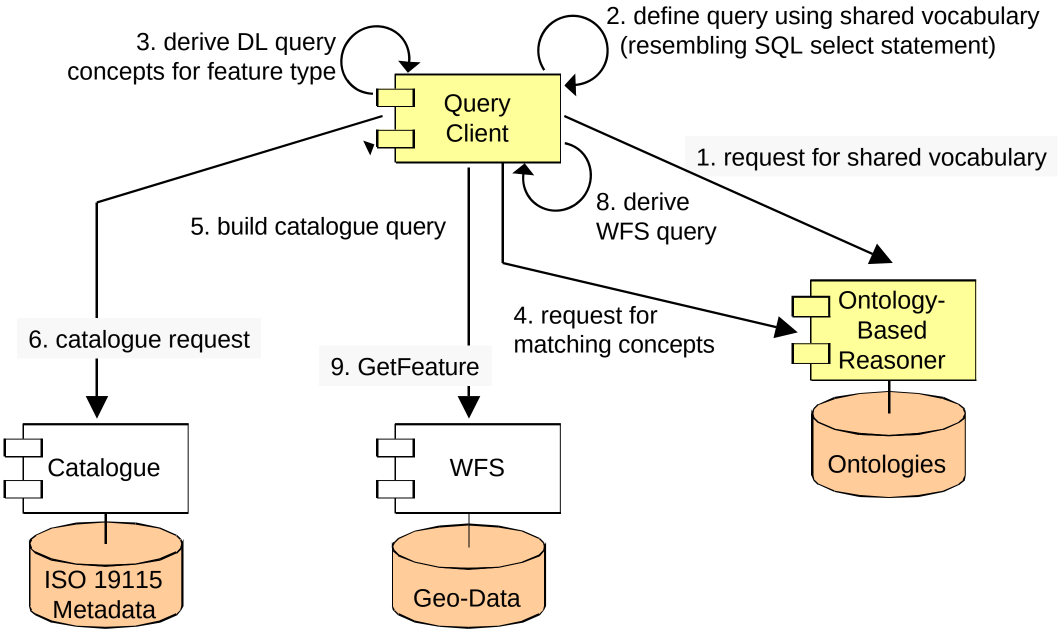

3. derive DL query concepts for feature type
2. define query using shared vocabulary (resembling SQL select statement)
Query Client
1. request for shared vocabulary
8. derive WFS query
5. build catalogue query
4. request for matching concepts
Ontology-Based Reasoner
6. catalogue request
9. GetFeature
Ontologies
Catalogue
WFS
ISO 19115 Metadata
Geo-Data

Figure 7. Architecture for IfGi semantic query sequence.

4.7 Experimental scenario

Since the precise make-up of semantic processing components was itself the subject of investigation in this experiment, interoperability between such components was not the principal goal of experimental activities. Instead, the experimental scenario involved multiple participant groups (2) beginning with a common query, a common set of ontologies, and a common set of feature collections hosted on WFS servers. The scenario goal was to successfully process the query given the other common sets of information and derive similar result feature sets. Analysis of the experimental results below then revealed common patterns in the query processing as subjects for future interoperability work.

5 Results

There were a number of results from this experiment. They include:

- A record of 6 months of deliberation and development in the GSW IE wiki (http://portal.opengeospatial.org/wiki/twiki/bin/view/GSWie/WebHome),
- Two test deployments of query-processing software (BBN/TASC and IfGi),
- Initial development of several geospatial ontologies,
- An initial semantic query architecture,
- Concepts for profiles of WFS and CS/W specifications

Some of these results are further discussed below.

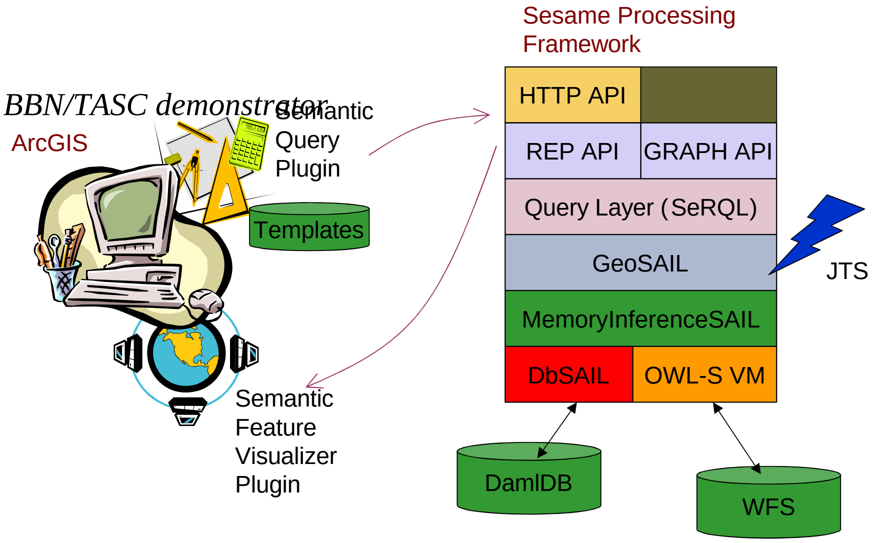
5.1 Demonstrations

Participants from BBN and NGIT/TASC have demonstrated the use of their Semantic Query client, Semantic Geospatial middleware, and multiple distributed WFS aeronav feature collections. Participants from IfGi Muenster have demonstrated their Query Concept client and Concept Decoration geospatial catalog, along with consequent access to relevant distributed WFS feature collections.

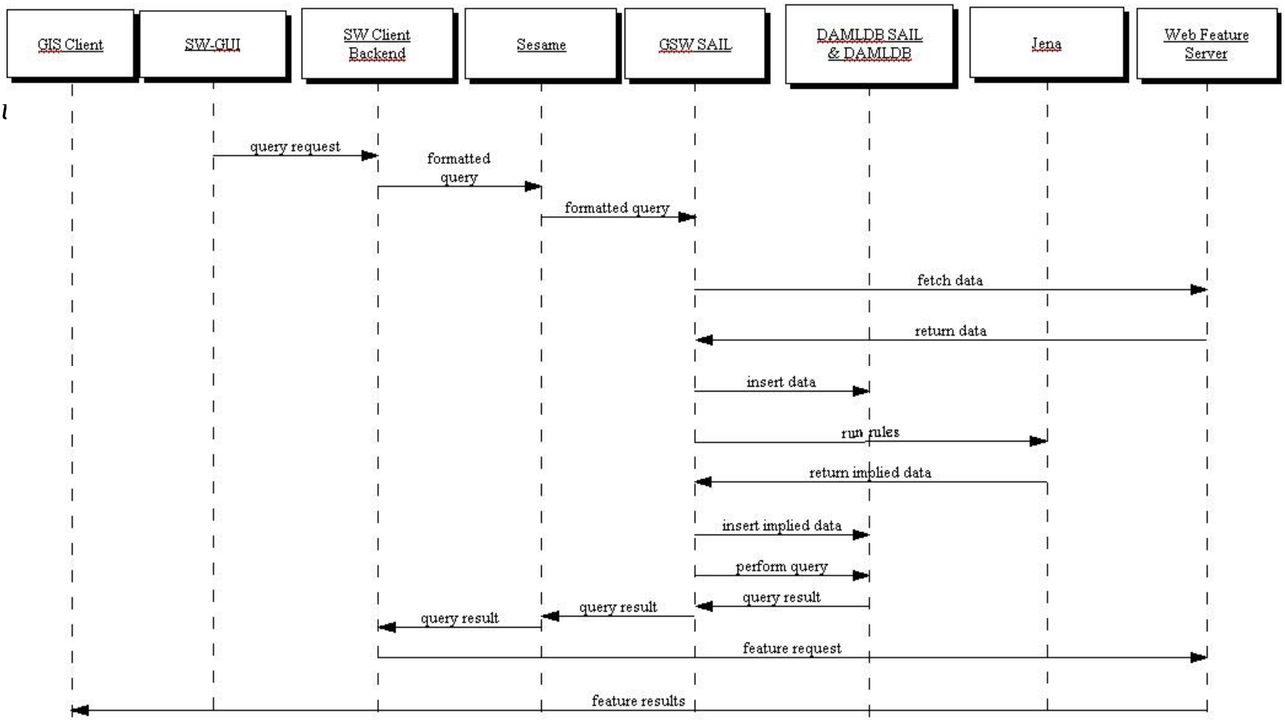
5.1.1 BBN/TASC

Figure 8 below depicts the final deployment configuration of the BBN - NGIT/TASC demonstrator, including combined query and visualization components as ArcGIS plug-ins, the GeoSAIL component within the SESAME middleware platform, and use of an OWL-S virtual machine to incorporate WFS contents within the query process. Figure 9 shows the sequence used by the demonstrator in processing the test query.

Figure 8. Deployment diagram of BBN/TASC demonstrator



Figı



5.1.2 IfGi

The demonstrated IfGi query sequence consists of the following steps:

1. **Select domain of interest.** In the first step the user has to select one or more domains of interest, i.e. he chooses the domain vocabulary he needs for formulating a query.

2. **Select concept and properties.** In the second step, the user selects the concept and properties that he is interested in. In our example, the user wants to know the name of an airport.

3. **Define constraints**. In the third step, the user defines the constraint the requested airport should meet. This is the most complex task during the process of query formulation. Finally the user has created a query for an airport that has a runway, which length is greater than 2000m. In the following steps, this query is used for both, the discovery of suitable WFS for providing this kind of information and for the retrieval of the actual feature from the WFS.

4. **Generate logic statement for semantic matchmaking.** For the task of discovering a suitable WFS, we want to take advantage of logical reasoning services that enable an automated matchmaking. Thus, we first have to translate the query statement displayed in step 4 into a logic statement.

5. **Submit semantic query for discovery**. In step 5 the results of the semantic matchmaking are displayed. The reasoning service has calculated that the QUERY concept is a subclass of the DafifAirport? concept defined in dafif.owl. (dafif.owl describes the runway and airport feature types offered by the DAFIF WFS). Consequently, the DAFIF WFS has been identified as a semantic match for the query and can now be selected by the user to be used for retrieving the requested information.

6. **Generate & submit WFS request for retrieval**. In the 6th step the GetFeature request has been generated based on the query statement from step 4. This also requires the translation from the ontological vocabulary used in Dafif.owl into the terms used in the GML schema of the DAFIF WFS. By submitting the WFS request the results can be displayed immediately.

The example has been simplified by basing the query formulation on measurements.owl and airports.owl only. This hopefuly provides enough insight for illustrating the principle behind the approach. However, an example of how the entire query would look like for the Airport/Runaway use case is provided in the queries.owl file

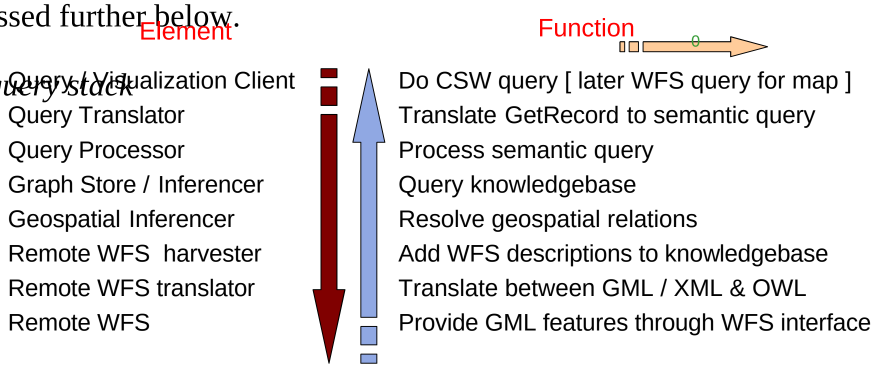
5.2 Architectural Model

Even given the fairly well defined and developed OWS geospatial services architecture, the topic of an architectural model for geosemantic services was perhaps the most vigorously debated area of the entire experiment. This vigor illustrates above all the ubiquitous potential applicability of semantic technology and representations. There were so many different ways in which processing steps could be distributed and so many different points at which transformations to/from semantic representations of information could prove useful, that there was plenty of room for individual preferences, available technical capabilities, and nuance of requirements.

The more persistent possibilities have been more-or-less combined and summarized in the semantic query stack described below. This stack defines a set of (almost) atomic processing steps which can be implemented in several varieties of distributed components (two of which were demonstrated in the experiment). Standard interfaces could be defined between any two of these processing steps, although it is likely that future development will identify a limited number of discrete components which will be of interest for widespread deployment. The generalized elements and their functions are summarized in Figure 10.

The processing sequences for some of the use cases implemented or contemplated in the experiment are shown in Figure 11. and discussed further below.

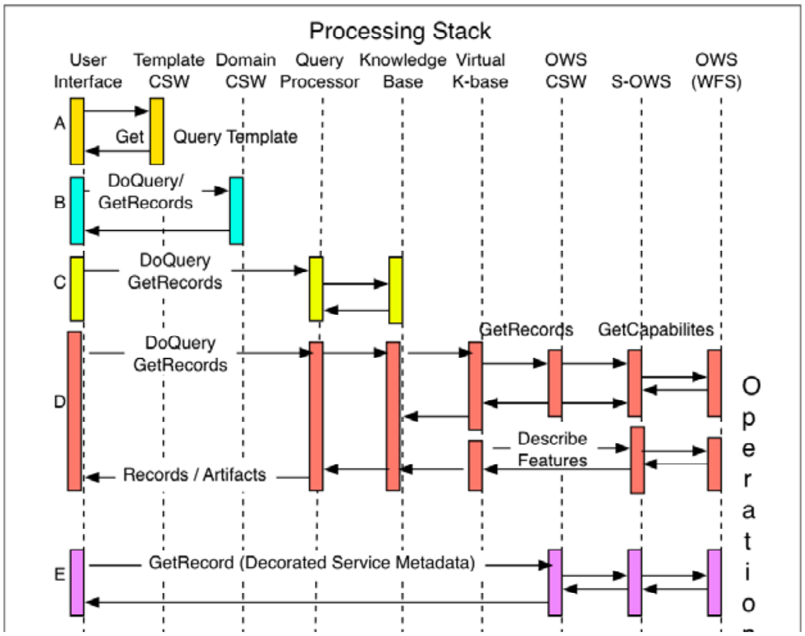
Figure 10. Typical geosemantic query stack



Element	Function
Query / Visualization Client	Do CSW query [later WFS query for map]
Query Translator	Translate GetRecord to semantic query
Query Processor	Process semantic query
Graph Store / Inferencer	Query knowledgebase
Geospatial Inferencer	Resolve geospatial relations
Remote WFS harvester	Add WFS descriptions to knowledgebase
Remote WFS translator	Translate between GML / XML & OWL
Remote WFS	Provide GML features through WFS interface

Standardized Web Services interfaces can be (or have been) defined between any two of these stack elements

Figure 11. Generalized geosema



Further use cases and representative component distributions are shown in Figure 12. Also shown in Figure 12 are the types of ontologies most closely involved with the processing steps shown. This gives an impression of where domain knowledge and technical knowledge are most likely to require synchronization.

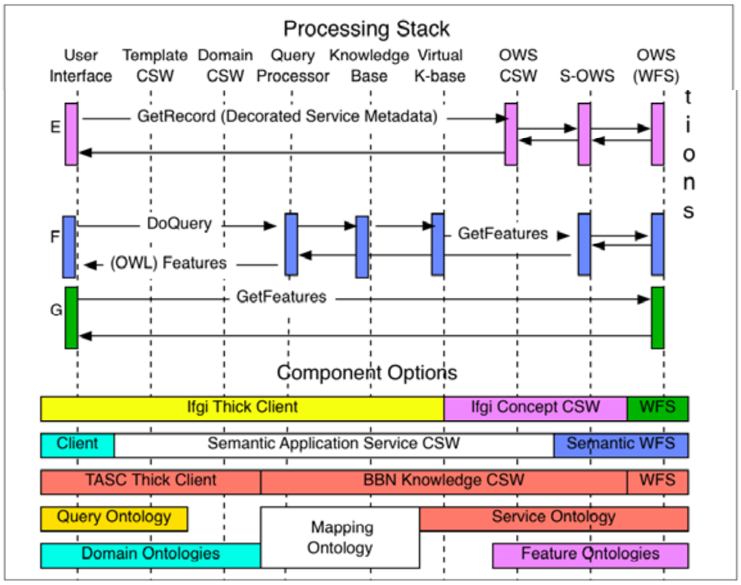


Figure 12. Generalized geosemantic query sequence: broker scenarios and alternative component configurations

5.2.1 Notes on the generalized query sequence:

5.2.1.1 Atomic services

1. User Interface - the application interface for the user

2. Template catalog - service for discovery & retrieval of query template ontologies for user interface configuration

3. Domain catalog - service for discovery & retrieval of domain ontologies familiar to the user with which construct queries.

4. Query processor - service to process queries (either in DoQuery? or GetRecord? requests, by performing reasoning and other operations on a knowledgebase.

5. Virtual K(nowledge)-base - knowledgebase service which represents and accesses remote knowledge as needed to support query processing in the knowledgebase.

6. OWS CSW - Service to harvest and catalog remote OWS services and contents (e.g. WFS and features.

7. S-OWS - service which represents an OWS service (e.g. WFS) but is able to respond to requests with responses in a semantic format (e.g. OWL feature ontology and/or OWLS-S description).

8. OWS - normal OGC Web service such as a WFS.

5.2.1.2 Operation Sequences

1. Discover and retrieve a query template ontology to implement in the client

2. Discover and retrieve domain ontology (or ontologies) which are used to "fill in the blanks" in the query

3. Issue query (shown here as a SPARQL-like DoQuery? request) against a fixed knowledgebase component

4. Issue query (either DoQuery? or csw:GetRecord) against a virtual knowledgebase which incorporates both feature and service ontologies as needed from remote OWS services (WFS +/- CSW here) with harvesting and translation of both GetCapabilities? and DescribeFeature?.

5. Issue GetRecord? request for (concept-annotated) feature and service records in an otherwise "normal" CSW Catalog.

6. Issue a DoQuery? request against a virtual knowledgebase which in term fetches - translates - returns matching feature individuals from matching remote WFS's.

7. Issue GetRecord? request against remote WFS and return GML

5.2.1.3 Distributed Components

Shown here are three possible componentizations and corresponding knowledge components which are processed by them. The pattern of semantic technologies as catalog-like mediation services between user-centered client applications and specialized remote resources should be clear.

1. Ifgi thick knowledge client -> Concept-decorated Service Catalog -> Web Feature Service(s)

2. Thin client -> query/domain/virtualK-base catalog(s) -> Semantic Web Feature Service(s) (WFS or WFS facade(s) returning OWL).

3. TASC thick client -> BBN GeoSESAME middleware / catalog (Sesame + SAIL's) -> Web Feature Service(s)

Query and domain (c.f. public) ontologies on the user (left) side are mapped / overlapped using a knowledgebase "mapping" or "base" ontology to feature and OWS service (c.f. private) ontologies on the resource (right) side.

5.2.2 Further observations on GSW architecture

TBD

5.3 WFS Profile

As discussed in Clause 3.5.1, there are a number of ways in which a semantic profile of WFS can be defined and implemented. Semantic enablement of WFS in this experiment was only accomplished implicitly, through façade or proxy elements in the semantic middleware components deployed by the participants, for example the GeoSAIL and OWL-S VM elements in GeoSESAME, or the concept-decorated service information in the IfGi catalog component. Nevertheless, the experiment has provided some insight into issues involved in selecting and refining future semantic WFS profiles.

5.3.1 Type I and type II profiles

Specification profiles in general are usually distinguished as Type I (only restrictive) where the information and/or operations are restricted from their nominally specified domains, or as Type II (restrictive and additive) where additional options, elements, or operations may be used to expand the nominal specification.

Almost all of the considered WFS profiles fall within the Type II category since they include additional options, formats, and/or operations. The exception may only be a compatibility profile in which the nominal capabilities, feature schemas, and GML features are restricted for maximal ease of translation to semantic representations such as OWL.

5.3.2 Levels of generalization

An important principle of distributed computing is that of appropriate generalization or level of detail. In other words, don't pull a gigabyte of features over the network to assess whether they meet a local need when the same can be done with 100 bytes of descriptive metadata. In the semantic processing realm, the same could be termed "Just-in Time Knowledge". Any useful semantic profile of WFS must be able to supply economical levels of detail – just enough content for query processing or inference to continue, but not more. This can present an interesting problem of matching the operations and formats of existing WFS' (i.e. GetCapabilities -> GetFeatureInfo -> GetFeature) to the almost stepless degrees of specificity which are possible in OWL and other processable graph-type knowledge representations.

5.3.3 Where to put the knowledge

Work in the experiment with semantically represented features and metadata has demonstrated the value of a proxy or façade approach to semantic enablement. In other words, existing feature service instances and their feature collections can be fronted by components which represent themselves as enhanced services and perform whatever processing and translation is required to interact successfully with the original WFS. The advantage of this approach is that it enables separation between the provision of features (the original WFS) and the provision of semantic technology (e.g. the OWL-aware or SPARQL-aware proxy). A disadvantage from the point of view of knowledge management, is that the original WFS may not be able to convey enough information to the proxy for generating an accurate knowledge representation, yet the needed human domain expertise is probably back with the providers of the original WFS. The technical concerns do not (necessarily) separate along the same lines as the domain concerns. This problem manifests itself both in the GML / OWS translator of the GeoSESAME platform and in the IfGi Concept-decorated Catalog.

The conclusion here is that there is probably no one semantic WFS profile (either integral or proxied) which suits all use cases. It will probably be important to define and implement 3-4 profile levels in order to meet the needs of real distributed geosemantic systems.

5.3.4 Case for semantic query language capability in a WFS

Suppose a WFS has a feature that it represents in GML like this:

```
<city fid="1">
   <name>Metropolis</name>
   <size>100000</size>
   <area>60</area>
   <geometry>
      <point>
         <x>-100</x>
         <y>50</y>
      </point>
   </geometry>
</city>
```

It may be important to redefine this semantically in OWL to take advantage of some standard semantic definitions. Something like this:

```
<city rdf:ID="city1">
   <name>Metropolis</name>
   <std1:population>100000</std1:population>
   <std2:area>
      <std2:units>sqkm</std2:units>
      <std2:value>60</std2:value>
   </std2:area>
   <geometry>
      <point>
         <x>-100</x>
         <y>50</y>
      </point>
   </geometry>
</city>
```

Now suppose a client draws all the cities in a particular area. It “understands” this semantic definition of population and wants to use it to determine how large a dot to draw on the map. All is well and good. It can do a GetFeature request using the BBOX operator in a standard OGC filter, and get the features back in the OWL it understands how to draw.

But suppose the client also wants to only retreive cities with a population greater than 10000. The client would be stuck because it has no way of knowing that the "size" GML property is the same thing as the "std1:population" OWL property, so it can't construct the appropriate OGC Filter.

So to avoid situations like this, I think it is necessary that the constraint language can use the OWL definitions directly. A standard semantic query language like SPARQL seems like it would be the best solution.

5.4 CS/W Profile

The concept of a specification profile is defined in ISO 19106, but the basic idea is to follow a set of rules for specializing, extending, and / or restricting an existing specification or protocol. In the case of the OGC Catalog 2.0 specification, there is a fairly well defined layering.

- Base specification defines conceptual operations, the basic (record) information model, and a core set of queryable properties, mainly from Dublin Core.
- A protocol layer defined how the conceptual operations may be implemented. There is an HTTP (called Catalog Services / Web or CS/W) and a CORBA profile defined in the base specification document.
- An information profile then defines more specifically the information model (record definitions and relationships) with which a catalog will be concerned. Examples of these include FGDC or ISO 19115/19119 profiles.

An information profile generally includes:

- Definitions of record types for catalog holdings - recordtypes
- Properties which can be queried to discover holdings (e.g. Dublin Core +) and their relationships to the recordtypes.
- Result models (e.g. element sets) and formats (e.g. XML) which can be requested from the catalog.
- One or more query languages (e.g. Common Query Language +) which can be used to define holdings of interest.
- Definitions of sources of information from which to build holdings - e.g. service capabilities

The ebRIM profile of CS/W is a little different because it defines not a model of the specific content to be catalogued, but a catalog management information model which can be used to catalog a variety of information resources. The ebRIM model explicitly accounts for common catalog information which allows resources to be discovered and retrieved. By pulling the information relevant to discovery out of the resource itself, an ebRIM-based catalog is not dependent on being able to reach "into" the resource format itself. ebRIM objects include:

- Catalogued resource - "extrinsicObject"

- Queryable properties - "slots"
- Resource relationships - "associations"
- Resource classifications - "classifications - classification nodes - classification schemes"

The concept of the ebRIM management model is important to semantic catalogs because it specifically enables access to content by defining how a resource may be discovered (e.g. slots, associations, classifications) separately from the discoverable resource itself. The added-value discovery information need not come from the resource itself or even the community which provides the resource. A classification particularly may characterize a resource in a completely different way than that resource is described by its own "community".

This leaves us some challenges in defining a "Semantic CS/W Profile"

- Reconcile the records-oriented base CS/W model with semantic graph objects such as OWL ontologies.
- Define (separately) the information model of registered resources and the information model of additional knowledge which adds value to the discovery process.

In general, the catalog role of separating and connecting the resources to be discovered from the information used for discovering them, fits well with our semantic middleware role of reasoning from query domains to data domains.

5.4.1 Semantic registry information model

This RIM attempts to reconcile the records-oriented base CS/W model with the semantic graph model by modeling each node and each relationship in the graph as a separate OWL Entity record. The "usedBy" and "impliedBy" relations are intended to link the OWL entity definitions to potential sources of instance data. For instance, if you have a WFS serving airports, you would have a Dataset record for the airport featuretype. You might also have OWLEntity records for an Airport class, a locatedAt property, and a Location class, etc. All of these OWLEntity records would use the usedBy relation to point back to the Dataset record. This would support both type reasoning and instance reasoning.

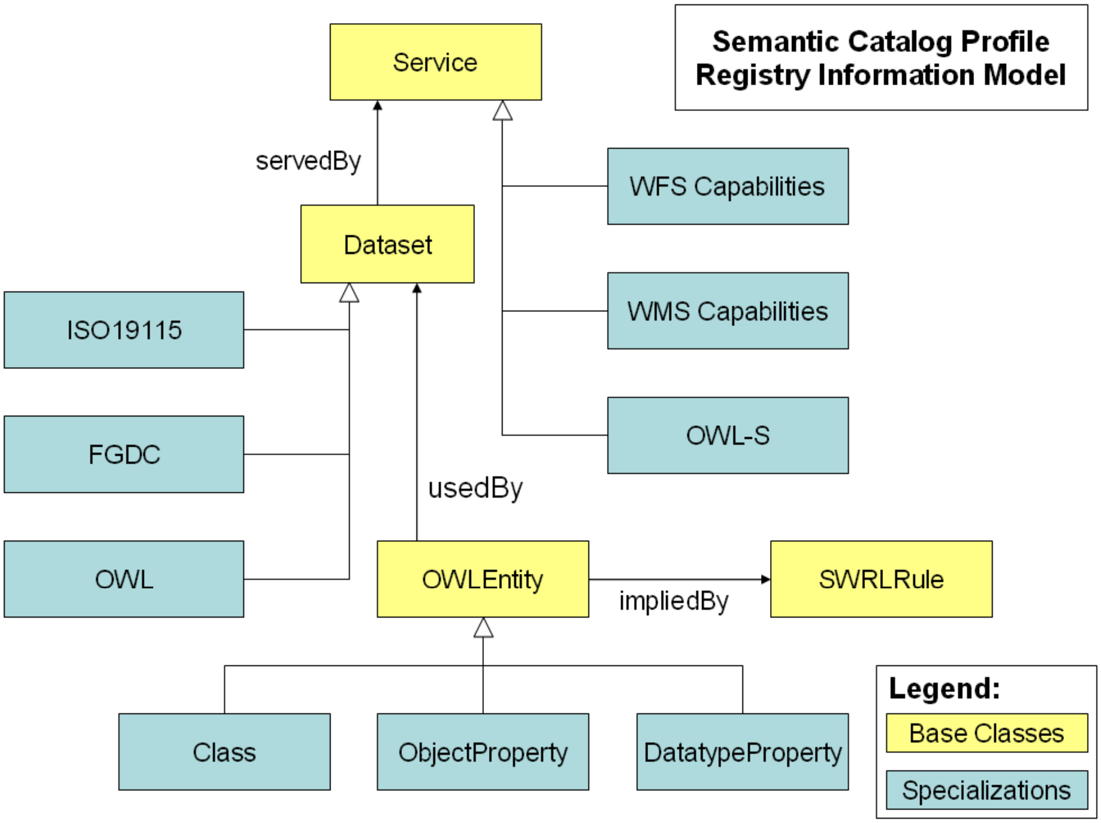


Figure 13. Object model for proposed semantic catalog information model

Type reasoning:

User starts by searching OWLEntity classes by keyword "airport", which turns up the Airport class record. The user traces the usedBy relationship back to the airport Dataset record to find the featuretype name and a link to the service, which he can use to make traditional WFS queries.

Instance reasoning:

User starts by searching OWLEntity classes by keyword "airport", which turns up the Airport class record. The user then queries for airport classes using a SPARQL query. The usedBy relationship information lets the catalog know which featuretypes from which WFS's it needs to retrieve data from to resolve the query. Similarly, the impliedBy relationships let the catalog know which SWRL rules it needs to evaluate to resolve the query.

How are the usedBy relationships created? Some of them are inserted manually or created by harvesting, but others are the result of reasoning. For example, you might manually add the relationships "{Airport Class} usedBy {Airport Dataset}" and "{Airfield Class} usedBy {Airfield Dataset}". But if the OWL definition of the Airfield specifies that it is a type of Airport, the catalog would automatically add the relationship "{Airport Class} usedBy {Airfield Dataset}".

5.5 GML OWL

Several activities were undertaken in this experiment around the idea of representing GML in the OWL language. This is not particularly far-fetched since early versions of GML were represented in RDF and GML still carries the object-property design pattern of RDF. Three stages of this representation might be described:

- Representation of metadata for GML features (e.g. FGDC, ISO 19115/139) in OWL. The IfGi implementation took this approach to representing particular feature types in their query concept. The potential for using

- Representation of GML schemas in OWL. There have been various efforts to render GML schema elements in OWL. The BBN geospatial SAIL infers an OWL description of (simple) GML features which it is able to access from remote WFS'. A more generalized representation is beyond the scope of this report. It has been complicated, among other things, by the fact that the GML feature "model" is not necessarily dependent on XML Schema but is largely expressed in XML Schema.

- Representation of GML feature instances in OWL. There was debate during the experiment on the importance of this representation. The participants ended up identifying some use cases in which discovering the correct feature collection to query as GML is a sufficient result of semantic processing; they identified other use cases where there are advantages in having an OWL representation of individual features, for determining whether they in fact satisfy the query, for participating in further processing (e.g. spatial buffering) and for relating the vocabulary of the features back to the query vocabulary (brokering).

Initial investigations suggest that there is a substantial profile of the GML 3.1.1 schema which could be represented in OWL without great difficulty. It also appears feasible, again for a GML profile, that substantially the same XML encoding for feature instances / individuals could serve to express both GML and GML/OWL. This would be a very worthwhile goal for future work.

5.6 Filter / geospatial ontology

The filter/geospatial ontologyshown in Annex A together with rules expresssions ("GeoSWRL") was used in the experiment both to process geospatial relationships and to construct WFS filter queries.

6 Conclusions and next steps

As stated at the beginning of this report, each of the activities and accomplishments of this experiment may be considered preliminary or initial steps towards GSW development. A number of conclusions and next steps are listed below; the lists are by no means exclusive. The overall value of the experiment is largely in enabling future activities to be undertaken with greater clarity of purpose, means, and expectation.

Concluding observations

- Semantic interoperability between participants was itself difficult to achieve.

- Experiment revealed many interesting and open research questions.

- Also discovered some disparities between rigor and utility
- Useful ontologies are not easy or quick to develop.
- Ontologies are typically not universal or immutable. Nature of a useful ontology depends on its purpose and context.
- GSW encompasses a wide range of use cases and distributive models. Some are compatible with existing OGC specifications and some are not (yet).
- This experiment has not "tested" new specifications, but has provided important material for defining them.
- Semantic processing has usable existing technology and great potential but requires extensive outreach, modest initial expectations, and testing against specific use cases for further GSW development.
- Assessment of semantic interoperability is still a work in progress:
- Semantic problems in geo-spatial data need to be clarified in order to be addressed
- There won't be one method solving all semantic problem types (man's-eye view).
- There are use cases for representing reality directly, as well as indirecty through representations / models

Next Steps

- Test implementations of (proposed) standard semantic OWS service profiles
- Refine and enrich developed ontologies (e.g. GML)
- Access, utilize, harmonize with other ontologies
- Apply methodology to new discovery and query use cases
- Refine and implement broker role in SOA: provision with translation

7 Terms and definitions

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

Domain
Area of common understanding characterized by a shared vocabulary

Knowledge
Awareness, understanding, and/or appreciation of information

Ontology
Controlled vocabulary which specifies a conceptualisation of a knowledge domain

Semantics
Meanings assigned to symbols and sets of symbols in a language, or by extension to expressions of data. Complement or opposite of syntax. In the context of this report, *meaning* and *understanding* refer specifically to computing capabilities for appropriate decision and action.

Semantic Web Services
Web services whose interfaces, descriptions, processes, and contents are expressed in a manner which can be understood by a machine.

Annex A. OWL ontologies

GML Ontology v.1

```
<?xml version='1.0'?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://gsw.projects.semwebcentral.org/2005/03/geofilter/gml-ont#"
  xml:base="http://gsw.projects.semwebcentral.org/2005/03/geofilter/gml-ont"
>
    <owl:Ontology rdf:about=""/>
    <owl:Class rdf:ID="Geometry"/>
    <owl:Class rdf:ID="Point">
        <rdfs:subClassOf rdf:resource="#Geometry"/>
    </owl:Class>

    <owl:DatatypeProperty rdf:ID="latitude">
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    </owl:DatatypeProperty>

    <owl:DatatypeProperty rdf:ID="longitude">
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    </owl:DatatypeProperty>
</rdf:RDF>
```

GML Ontology v.2

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:geom="http://babyz.uni-muenster.de/gswie/geomTypes.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns="http://babyz.uni-muenster.de/gswie/gml.owl#"
  xml:base="http://babyz.uni-muenster.de/gswie/gml.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://babyz.uni-
muenster.de/gswie/geomTypes.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Feature"/>
  <owl:Class rdf:ID="GeographicFeature">
    <rdfs:subClassOf rdf:resource="#Feature"/>
  </owl:Class>
  <owl:Class rdf:ID="FeatureCollection">

    <rdfs:subClassOf rdf:resource="#Feature"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="hasLocation">
    <rdfs:domain rdf:resource="#GeographicFeature"/>
    <rdfs:range rdf:resource="http://babyz.uni-
muenster.de/gswie/geomTypes.owl#Point"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasBoundingBox">
```

```
    <rdfs:domain rdf:resource="#GeographicFeature"/>
  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:ID="hasMember">
    <rdfs:domain rdf:resource="#FeatureCollection"/>
    <rdfs:range rdf:resource="#Feature"/>
  </owl:ObjectProperty>
</rdf:RDF>

<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://babyz.uni-muenster.de/gswie/geomTypes.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://babyz.uni-muenster.de/gswie/geomTypes.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Composite">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Complex"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="MultiSurface">
    <rdfs:subClassOf>

      <owl:Class rdf:ID="MultiPrimitive"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#MultiPrimitive">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Aggregate"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Aggregate">

    <rdfs:subClassOf>
      <owl:Class rdf:ID="Object"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="CompositeCurve">
    <rdfs:subClassOf rdf:resource="#Composite"/>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="OrientableCurve"/>
    </rdfs:subClassOf>

  </owl:Class>
  <owl:Class rdf:ID="CompositeSolid">
    <rdfs:subClassOf rdf:resource="#Composite"/>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Solid"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Curve">
    <rdfs:subClassOf>

      <owl:Class rdf:about="#OrientableCurve"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="DirectPosition">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >hold the coordinates for a posiiton within some cooridnate reference
system</rdfs:comment>
    <rdfs:subClassOf>
```

```
      <owl:Class rdf:ID="Geometry"/>
    </rdfs:subClassOf>

    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >0</owl:minCardinality>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="coordinateReferenceSystem"/>
            </owl:onProperty>

          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:about="#coordinateReferenceSystem"/>
            </owl:onProperty>
            <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:maxCardinality>
          </owl:Restriction>
        </owl:intersectionOf>

      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="OrientableSurface">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="OrientablePrimitive"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Complex">

    <rdfs:subClassOf>
      <owl:Class rdf:about="#Object"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Primitive">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Object"/>
    </rdfs:subClassOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >abstract root class of the geometric primitives
defines the basic "boundary" operations that ties the primitives in each
dimension together</rdfs:comment>

  </owl:Class>
  <owl:Class rdf:ID="Coordinate">
    <rdfs:subClassOf rdf:resource="#Geometry"/>
  </owl:Class>
  <owl:Class rdf:about="#OrientableCurve">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#OrientablePrimitive"/>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:ID="SC_CRS">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="directPosition"/>
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >0</owl:minCardinality>
      </owl:Restriction>

    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#Geometry"/>
  </owl:Class>
  <owl:Class rdf:about="#Object">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >root class of the geometric object taxonomy
supports interfaces common to all geographically referenced geometric
objects</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#Geometry"/>
  </owl:Class>
  <owl:Class rdf:about="#Solid">

    <rdfs:subClassOf rdf:resource="#Primitive"/>
  </owl:Class>
  <owl:Class rdf:ID="MultiPoint">
    <rdfs:subClassOf rdf:resource="#MultiPrimitive"/>
  </owl:Class>
  <owl:Class rdf:ID="CompositePoint">
    <rdfs:subClassOf rdf:resource="#Composite"/>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Point"/>

    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="MultiSolid">
    <rdfs:subClassOf rdf:resource="#MultiPrimitive"/>
  </owl:Class>
  <owl:Class rdf:ID="CompositeSurface">
    <rdfs:subClassOf rdf:resource="#OrientableSurface"/>
    <rdfs:subClassOf rdf:resource="#Composite"/>
  </owl:Class>

  <owl:Class rdf:ID="Surface">
    <rdfs:subClassOf rdf:resource="#OrientableSurface"/>
  </owl:Class>
  <owl:Class rdf:about="#Point">
    <rdfs:subClassOf rdf:resource="#Primitive"/>
  </owl:Class>
  <owl:Class rdf:ID="MultiCurve">
    <rdfs:subClassOf rdf:resource="#MultiPrimitive"/>
  </owl:Class>

  <owl:Class rdf:about="#OrientablePrimitive">
    <rdfs:subClassOf rdf:resource="#Primitive"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="coordinates">
    <rdfs:domain rdf:resource="#DirectPosition"/>
    <rdfs:range rdf:resource="#Coordinate"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="position">
    <rdfs:domain rdf:resource="#Point"/>

    <rdfs:range rdf:resource="#DirectPosition"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#directPosition">
    <rdfs:domain rdf:resource="#SC_CRS"/>
    <rdfs:range rdf:resource="#DirectPosition"/>
    <owl:inverseOf>
```

```
      <owl:ObjectProperty rdf:about="#coordinateReferenceSystem"/>
    </owl:inverseOf>
  </owl:ObjectProperty>

  <owl:ObjectProperty rdf:about="#coordinateReferenceSystem">
    <owl:inverseOf rdf:resource="#directPosition"/>
    <rdfs:range rdf:resource="#SC_CRS"/>
    <rdfs:domain rdf:resource="#DirectPosition"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="latitude">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:domain rdf:resource="#Coordinate"/>
  </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:ID="longitude">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:domain rdf:resource="#Coordinate"/>
  </owl:DatatypeProperty>
</rdf:RDF>
```

Airport Ontology

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://gsw.projects.semwebcentral.org/2005/03/airport/airport-ont#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:meas="http://babyz.uni-muenster.de/gswie/measurements.owl#"
  xml:base="http://gsw.projects.semwebcentral.org/2005/03/airport/airport-ont">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="RefuelingService">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Service"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="availableFuelType"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="FuelType"/>
  <owl:Class rdf:ID="PCN">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >What is PCN?</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="STARS">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="LandingProcedure"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Airport">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >included constraint "name >= 1" (to compensate for deletion of
airportName)</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:FunctionalProperty rdf:ID="name"/>
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:FunctionalProperty rdf:ID="isLocatedAt"/>
        </owl:onProperty>
        <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class rdf:ID="Runway"/>
        </owl:someValuesFrom>
        <owl:onProperty>
          <owl:TransitiveProperty rdf:ID="hasPart"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="ComplexFeature"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#ComplexFeature">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Feature"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Apron">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="SimpleFeature"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="EmergencyService">
    <rdfs:subClassOf rdf:resource="#Service"/>
  </owl:Class>
  <owl:Class rdf:ID="RepairType"/>
  <owl:Class rdf:ID="NAVAID">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#SimpleFeature"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="NDB">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="NAVAIDService"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Taxiway">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#SimpleFeature"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ILS">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#NAVAIDService"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="MLS">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#NAVAIDService"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="SID">
    <rdfs:subClassOf rdf:resource="#LandingProcedure"/>
  </owl:Class>
  <owl:Class rdf:ID="OxygenService">
    <rdfs:subClassOf rdf:resource="#Service"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="availableOxygenType"/>
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Runway">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >changed cardinality constraint on isTerminatedBy to "= 2"
included constraint "name >= 1" (to compensate for deletion of
runwayName)</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="isTerminatedBy"/>
        </owl:onProperty>
        <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >2</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
        <owl:onProperty>
          <owl:FunctionalProperty rdf:about="#name"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
        <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#length"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#SimpleFeature"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Obstruction">
```

```
    <rdfs:subClassOf>
      <owl:Class rdf:about="#SimpleFeature"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="TACAN">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#NAVAIDService"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Lighting">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >why is the constraint "availableLightingType >= 1"  specified (cf. other
SimpleFeatures)?</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#SimpleFeature"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="availableLightingType"/>
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="NOTAMService">
    <rdfs:subClassOf rdf:resource="#Service"/>
  </owl:Class>
  <owl:Class rdf:ID="RunwayEnd">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#SimpleFeature"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#NAVAIDService">
    <rdfs:subClassOf rdf:resource="#Service"/>
  </owl:Class>
  <owl:Class rdf:ID="Location">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >What is a location? Is it the same as a (specific kind of) point (as it
has a lat and long)?
Should this concept not rather be in the geo(metry) domain
ontology?</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:FunctionalProperty rdf:ID="elevationReferencePoint"/>
        </owl:onProperty>
        <owl:allValuesFrom
rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#elevation"/>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="LightingType"/>
  <owl:Class rdf:ID="OxygenType"/>
  <owl:Class rdf:ID="VOR">
    <rdfs:subClassOf rdf:resource="#NAVAIDService"/>
  </owl:Class>
  <owl:Class rdf:ID="C5">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="PlaneType"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="WeatherService">
    <rdfs:subClassOf rdf:resource="#Service"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
        <owl:onProperty>
          <owl:DatatypeProperty rdf:ID="availableDataSource"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#SimpleFeature">
    <rdfs:subClassOf rdf:resource="#Feature"/>
  </owl:Class>
  <owl:Class rdf:about="#PlaneType">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Changed concept name to PlaneType - is this correct (cf. fuelType,
tailNumber)?</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="RepairService">
    <rdfs:subClassOf rdf:resource="#Service"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="availableRepairType"/>
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:ObjectProperty rdf:about="#availableLightingType">
    <rdfs:domain rdf:resource="#Lighting"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >changed range to LightingType (in analogy to FuelType)</rdfs:comment>
    <rdfs:range rdf:resource="#LightingType"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="requiresFuel">
    <rdfs:range rdf:resource="#FuelType"/>
    <rdfs:domain rdf:resource="#PlaneType"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="providesNAVAIDService">
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:ID="providesService"/>
    </rdfs:subPropertyOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Are NAVAIDs airports? If not this should not be defined as a subproperty
of providesService.</rdfs:comment>
    <rdfs:range rdf:resource="#NAVAIDService"/>
    <rdfs:domain rdf:resource="#NAVAID"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="services">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >what is the meaning of this role?
define range!</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#providesService">
    <rdfs:domain rdf:resource="#Airport"/>
    <rdfs:range rdf:resource="#Service"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Are all things that provide services airports? If not the domain should be
deleted.</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="isLitBy">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Runway"/>
          <owl:Class rdf:about="#RunwayEnd"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="#Lighting"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="runwayWeightBearingCapacity">
    <rdfs:range>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#WeightQuantity"/>
          <owl:Restriction>
            <owl:allValuesFrom rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Tonne"/>
            <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </rdfs:range>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Runway"/>
          <owl:Class rdf:about="#RunwayEnd"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:subPropertyOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#weight"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >defined as subproperty of meas:weight
unitOfMeasure defined as Tonne in the range definition (change if
required)</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#availableOxygenType">
    <rdfs:domain rdf:resource="#OxygenService"/>
    <rdfs:range rdf:resource="#OxygenType"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >changed range toOxygenType (in analogy to FuelType)</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#availableRepairType">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >changed range to RepairType (in analogy to FuelType)</rdfs:comment>
    <rdfs:range rdf:resource="#RepairType"/>
    <rdfs:domain rdf:resource="#RepairService"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#isTerminatedBy">
    <rdfs:range rdf:resource="#RunwayEnd"/>
    <rdfs:domain rdf:resource="#Runway"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="isObstructedBy">
    <rdfs:domain rdf:resource="#Runway"/>
    <rdfs:range rdf:resource="#Obstruction"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Are all things that can be obstructed runways? If not, delete
domain.</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasPCN">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >What is a PCN?</rdfs:comment>
    <rdfs:range rdf:resource="#PCN"/>
    <rdfs:domain rdf:resource="#Runway"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="maxTirePressure">
    <rdfs:range>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#PressureQuantity"/>
          <owl:Restriction>
            <owl:allValuesFrom rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Hektopascal"/>
            <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </rdfs:range>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >defined as subproperty of meas:pressure
unitOfMeasure defined as Hektopascal in the range definition (change if
required)</rdfs:comment>
    <rdfs:domain rdf:resource="#PCN"/>
    <rdfs:subPropertyOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#pressure"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="servicesTaxiway">
    <rdfs:range rdf:resource="#Taxiway"/>
    <rdfs:domain rdf:resource="#Apron"/>
    <rdfs:subPropertyOf rdf:resource="#services"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#availableFuelType">
    <rdfs:domain rdf:resource="#RefuelingService"/>
    <rdfs:range rdf:resource="#FuelType"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="providesLandingProcedure">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Airport"/>
          <owl:Class rdf:about="#Runway"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="#LandingProcedure"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="runwaySurface">
    <rdfs:domain>
```

```
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Runway"/>
          <owl:Class rdf:about="#RunwayEnd"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >shouldn't this be an ObjectProperty?</rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="icaoCode">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Airport"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="notamFileIdentifier">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#NOTAMService"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="trafficPatternAltitude">
    <rdfs:domain rdf:resource="#Airport"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >What is the meaning of this role?</rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="timeConversion">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >What is the meaning of this role?</rdfs:comment>
    <rdfs:domain rdf:resource="#Airport"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="#availableDataSource">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >deleted WeatherService from domain</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="navaidFrequency">
    <rdfs:domain rdf:resource="#NAVAIDService"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >shouldn't this be an ObjectProperty?</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="iataCode">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Airport"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="navaidIdentifier">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#NAVAID"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >shouldn't this have int as a range (or are these alphanumeric
IDs)?</rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="navaidChannel">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >shouldn't this be an ObjectProperty?</rdfs:comment>
    <rdfs:domain rdf:resource="#NAVAIDService"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="tailNumber">
```

```
    <rdfs:domain rdf:resource="#PlaneType"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Is that an attribute of a particular plane or of a type of plane? (cf. 
fuelType)</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:TransitiveProperty rdf:about="#hasPart">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >changed name from comprises to hasPart
actually, this should be part of ;-) an ontology on "partonomy"...
deleted subroles -- and changed the constraints in airport concept accordingly
defined role to be transitive</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </owl:TransitiveProperty>
  <owl:SymmetricProperty rdf:ID="intersectsRunway">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain rdf:resource="#Runway"/>
    <rdfs:range rdf:resource="#Runway"/>
    <owl:inverseOf rdf:resource="#intersectsRunway"/>
  </owl:SymmetricProperty>
  <owl:SymmetricProperty rdf:ID="servicesRunway">
    <owl:inverseOf rdf:resource="#servicesRunway"/>
    <rdfs:domain rdf:resource="#Taxiway"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="#Runway"/>
    <rdfs:subPropertyOf rdf:resource="#services"/>
  </owl:SymmetricProperty>
  <owl:FunctionalProperty rdf:ID="navaidName">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >can this be changed to simply "name"?</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="#NAVAID"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="pavementSubgradeCategory">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >shouldn't this be an ObjectProperty?</rdfs:comment>
    <rdfs:domain rdf:resource="#PCN"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#elevationReferencePoint">
    <rdfs:domain rdf:resource="#Location"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >range should be some kind of point from the geo(metry) 
ontology</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="pcnNumber">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >shouldn't this have int as a range?</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="#PCN"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="latitude">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
    <rdfs:domain rdf:resource="#Location"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#name">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >there should be only one kind of name (for airports and runways and other
things)
therefore also deleted domain</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="longitude">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="#Location"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="pavementType">
    <rdfs:domain rdf:resource="#PCN"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >shouldn't this be an ObjectProperty?</rdfs:comment>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:ID="pavementEvaluationMethod">
    <rdfs:domain rdf:resource="#PCN"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >shouldn't this be an ObjectProperty?</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:FunctionalProperty rdf:about="#isLocatedAt">
    <rdfs:range rdf:resource="#Location"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >deleted domain because also things that are not airports or NAVAIDServices
can have a location. Should this role not rather be in the geo(metry) domain
ontology?</rdfs:comment>
  </owl:FunctionalProperty>
</rdf:RDF>
```

Filter Ontology

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
    <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
    <!ENTITY gml-ont
'http://gsw.projects.semwebcentral.org/2005/03/geofilter/gml-ont'>
]>
<rdf:RDF
    xmlns="http://gsw.projects.semwebcentral.org/2005/03/geofilter/filter-ont#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:gml="http://gsw.projects.semwebcentral.org/2005/03/geofilter/gml-
ont#"
    xml:base="http://gsw.projects.semwebcentral.org/2005/03/geofilter/filter-
ont">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Location"/>
  <owl:Class rdf:ID="Filter"/>
  <owl:Class rdf:ID="SpatialOp">
    <rdfs:subClassOf rdf:resource="#Filter"/>
  </owl:Class>
  <owl:Class rdf:ID="ComparisonOp">
```

```
    <rdfs:subClassOf rdf:resource="#Filter"/>
  </owl:Class>
  <owl:Class rdf:ID="LogicOp">
    <rdfs:subClassOf rdf:resource="#Filter"/>
  </owl:Class>
  <owl:Class rdf:ID="BinarySpatialOp">
    <rdfs:subClassOf rdf:resource="#SpatialOp"/>
    <rdfs:subClassOf>
       <owl:Restriction>
           <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</owl:cardinality>
           <owl:allValuesFrom rdf:resource="&gml-ont;#Geometry"/>
           <owl:onProperty>
               <owl:DatatypeProperty rdf:about="operand"/>
           </owl:onProperty>
       </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Equals">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Disjoint">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Touches">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Within">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Overlaps">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Crosses">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Intersects">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Contains">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="DistanceBuffer">
    <rdfs:subClassOf rdf:resource="#SpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="DWithin">
    <rdfs:subClassOf rdf:resource="#DistanceBuffer"/>
  </owl:Class>
  <owl:Class rdf:ID="Beyond">
    <rdfs:subClassOf rdf:resource="#DistanceBuffer"/>
  </owl:Class>
  <owl:Class rdf:ID="BBox">
    <rdfs:subClassOf rdf:resource="#DistanceBuffer"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="within"/>
  <owl:DatatypeProperty rdf:ID="radius">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="referenceGeometry">
    <rdfs:range
rdf:resource="http://gsw.projects.semwebcentral.org/2005/03/geofilter/gml-
ont#Geometry"/>
```

```
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="satisfiesFilter">
    <rdfs:range rdf:resource="#Filter"/>  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="operand"/>
  <owl:Class rdf:ID="BinaryLogicOp">
    <rdfs:subClassOf rdf:resource="#LogicOp"/>
  </owl:Class>
  <owl:Class rdf:ID="UnaryLogicOp">
    <rdfs:subClassOf rdf:resource="#LogicOp"/>
  </owl:Class>
  <!--<owl:Class rdf:ID="PropertyIs"-->
</rdf:RDF>
```

Filter-2 Ontology

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
    <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
    <!ENTITY gml-ont
'http://gsw.projects.semwebcentral.org/2005/03/geofilter/gml-ont'>
]>
<rdf:RDF
    xmlns="http://gsw.projects.semwebcentral.org/2005/03/geofilter/filter-ont#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:gml="http://gsw.projects.semwebcentral.org/2005/03/geofilter/gml-
ont#"
    xml:base="http://gsw.projects.semwebcentral.org/2005/03/geofilter/filter-
ont">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Location"/>
  <owl:Class rdf:ID="Filter"/>
  <owl:Class rdf:ID="SpatialOp">
    <rdfs:subClassOf rdf:resource="#Filter"/>
  </owl:Class>
  <owl:Class rdf:ID="ComparisonOp">
    <rdfs:subClassOf rdf:resource="#Filter"/>
  </owl:Class>
  <owl:Class rdf:ID="LogicOp">
    <rdfs:subClassOf rdf:resource="#Filter"/>
  </owl:Class>
  <owl:Class rdf:ID="BinarySpatialOp">
    <rdfs:subClassOf rdf:resource="#SpatialOp"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</owl:cardinality>
            <owl:allValuesFrom rdf:resource="&gml-ont;#Geometry"/>
            <owl:onProperty>
                <owl:DatatypeProperty rdf:about="operand"/>
            </owl:onProperty>
        </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Equals">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Disjoint">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Touches">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Within">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Overlaps">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Crosses">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Intersects">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="Contains">
    <rdfs:subClassOf rdf:resource="#BinarySpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="DistanceBuffer">
    <rdfs:subClassOf rdf:resource="#SpatialOp"/>
  </owl:Class>
  <owl:Class rdf:ID="DWithin">
    <rdfs:subClassOf rdf:resource="#DistanceBuffer"/>
  </owl:Class>
  <owl:Class rdf:ID="Beyond">
    <rdfs:subClassOf rdf:resource="#DistanceBuffer"/>
  </owl:Class>
  <owl:Class rdf:ID="BBox">
    <rdfs:subClassOf rdf:resource="#DistanceBuffer"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="within"/>
  <owl:DatatypeProperty rdf:ID="radius">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="referenceGeometry">
    <rdfs:range
rdf:resource="http://gsw.projects.semwebcentral.org/2005/03/geofilter/gml-
ont#Geometry"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="satisfiesFilter">
    <rdfs:range rdf:resource="#Filter"/>  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="operand"/>
  <owl:Class rdf:ID="BinaryLogicOp">
    <rdfs:subClassOf rdf:resource="#LogicOp"/>
  </owl:Class>
  <owl:Class rdf:ID="UnaryLogicOp">
    <rdfs:subClassOf rdf:resource="#LogicOp"/>
  </owl:Class>
  <!--<owl:Class rdf:ID="PropertyIs"-->
</rdf:RDF>
```

Measurement Ontology

```
<?xml version="1.0"?>
<rdf:RDF
```

```
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://dummy.com/ontology">
  <owl:Ontology rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl"/>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Hektopascal">
    <owl:equivalentClass>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Millibar"/>
    </owl:equivalentClass>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#PressureUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Kilometer">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Degree">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#SlopeUnit"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#DirectionUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#PressureQuantity">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
        </owl:onProperty>
        <owl:allValuesFrom>
          <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#PressureUnit"/>
        </owl:allValuesFrom>
      </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#SlopeQuantity">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
    </rdfs:subClassOf>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:allValuesFrom>
```

```
          <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#SlopeUnit"/>
        </owl:allValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
        </owl:onProperty>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#CubicMeter">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#VolumeUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Millimeter">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#UKPound">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#WeightUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Hectare">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#AreaUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Dezimeter">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#SlopeUnit">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Unit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#SquareMeter">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#AreaUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Are">
    <rdfs:subClassOf>
```

```
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#AreaUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Meter">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#USPound">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#WeightUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#AreaQuantity">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
        </owl:onProperty>
        <owl:allValuesFrom>
          <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#AreaUnit"/>
        </owl:allValuesFrom>
      </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthQuantity">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
        </owl:onProperty>
        <owl:allValuesFrom>
          <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit"/>
        </owl:allValuesFrom>
      </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Foot">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Nanometer">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#VolumeUnit">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Unit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#InchesOfMercuryColumn">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#PressureUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#VolumeQuantity">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
        </owl:onProperty>
        <owl:allValuesFrom rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#VolumeUnit"/>
      </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#WeightQuantity">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:allValuesFrom>
          <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#WeightUnit"/>
        </owl:allValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
        </owl:onProperty>
      </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Acre">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#AreaUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Mikrometer">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#AreaUnit">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Unit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Atmosphere">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#PressureUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Percent">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#SlopeUnit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Pascal">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#PressureUnit"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#DirectionQuantity">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
        </owl:onProperty>
        <owl:allValuesFrom>
          <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#DirectionUnit"/>
        </owl:allValuesFrom>
      </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Unit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#WeightUnit">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Unit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Kilogram">
```

```
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#WeightUnit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#PressureUnit">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Unit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#DirectionUnit">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Unit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Tonne">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#WeightUnit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#SquareFoot">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#AreaUnit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty>
              <owl:DatatypeProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#value"/>
            </owl:onProperty>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure"/>
            </owl:onProperty>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Millibar">
    <owl:equivalentClass rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Hektopascal"/>
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#PressureUnit"/>
  </owl:Class>
  <owl:Class rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#Centimeter">
    <rdfs:subClassOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#LengthUnit"/>
  </owl:Class>
  <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#weight">
    <rdfs:range rdf:resource="http://babyz.uni-muenster.de/gswie/measurements.owl#WeightQuantity"/>
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#quantity"/>
    </rdfs:subPropertyOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#width">
    <rdfs:range rdf:resource="http://babyz.uni-muenster.de/gswie/measurements.owl#LengthQuantity"/>
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#quantity"/>
    </rdfs:subPropertyOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#orientation">
    <rdfs:range rdf:resource="http://babyz.uni-muenster.de/gswie/measurements.owl#DirectionQuantity"/>
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#quantity"/>
    </rdfs:subPropertyOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#slope">
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#quantity"/>
    </rdfs:subPropertyOf>
    <rdfs:range rdf:resource="http://babyz.uni-muenster.de/gswie/measurements.owl#SlopeQuantity"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#elevation">
    <rdfs:range rdf:resource="http://babyz.uni-muenster.de/gswie/measurements.owl#LengthQuantity"/>
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#quantity"/>
    </rdfs:subPropertyOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#length">
    <rdfs:range rdf:resource="http://babyz.uni-muenster.de/gswie/measurements.owl#LengthQuantity"/>
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#quantity"/>
    </rdfs:subPropertyOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#pressure">
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:about="http://babyz.uni-muenster.de/gswie/measurements.owl#quantity"/>
    </rdfs:subPropertyOf>
    <rdfs:range rdf:resource="http://babyz.uni-muenster.de/gswie/measurements.owl#PressureQuantity"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#quantity">
    <rdfs:range rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#unitOfMeasure">
    <rdfs:range rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Unit"/>
    <rdfs:domain rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#volume">
    <rdfs:subPropertyOf rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#quantity"/>
    <rdfs:range rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#VolumeQuantity"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#value">
    <rdfs:domain rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements.owl#Quantity"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="http://babyz.uni-
muenster.de/gswie/measurements.owl#count">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  </owl:DatatypeProperty>
</rdf:RDF>
```

DAFIF Airport Ontology example

```
<?xml version="1.0"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Klien (Institut für
Geoinformatik) -->
<rdf:RDF xmlns:run="http://babyz.uni-muenster.de/gswie/dafifrunways.owl#"
xmlns="http://babyz.uni-muenster.de/gswie/dafifairportsV2.owl#"
xmlns:airp="http://babyz.uni-muenster.de/ontologies/airport-ont-ifgi.owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:meas="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
xmlns:dc="http://purl.org/dc/elements/1.1/" xml:base="http://babyz.uni-
muenster.de/gswie/dafifairportsV2.owl">
    <owl:Ontology rdf:about="">
        <owl:imports rdf:resource="http://babyz.uni-
muenster.de/ontologies/airport-ont-ifgi.owl"/>
        <owl:imports rdf:resource="http://babyz.uni-
muenster.de/gswie/dafifrunways.owl"/>
    </owl:Ontology>
    <owl:Class rdf:ID="Inactive">
        <rdfs:subClassOf>
            <owl:Class rdf:ID="AirportUsage"/>

        </rdfs:subClassOf>
        <rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">ACTIVE AIRPORTS HAVING
```

```
PERMANENT TYPE SURFACE RUNWAYS WITH LESS THAN THE MINIMUM FACILITIES REQUIRED
FOR A, B, OR C AIRPORTS ABOVE. AIRPORTS UNDER CONSTRUCTION WITH NO RUNWAY YET
USABLE, ARE INCLUDED IN THIS CATEGORY. (FOR THE UNITED STATES THE FAA TERM
"INACTIVE" WILL BE INCORPORATED INTO THIS CATEGORY). (AAFIF)</rdfs:comment>
    </owl:Class>
    <owl:Class rdf:ID="ActiveJoint">
        <rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">ACTIVE JOINT (CIVIL AND
MILITARY). AIRPORTS JOINTLY CONTROLLED, USED AND/OR OPERATED BY BOTH CIVIL AND
MILITARY AGENCIES. THE MILITARY AGENCIES MUST BE PERMANENT OPERATIONAL FLIGHT
LINE TYPE TENANTS, WITH OR WITHOUT AIRCRAFT STATIONED ON THE AIRPORT.
FACILITIES ARE THE SAME AS FOR CIVIL (A) AIRPORTS. (AAFIF)</rdfs:comment>
        <rdfs:subClassOf rdf:resource="#AirportUsage"/>
    </owl:Class>
    <owl:Class rdf:ID="ActiveCivil">

        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
ACTIVE CIVIL AIRPORTS CONTROLLED AND OPERATED BY CIVIL AUTHORITIES PRIMARILY
FOR USE BY CIVIL AIRCRAFT, ALTHOUGH MILITARY MAY HAVE LANDING PRIVILEGES AND/OR
CONTRACT RIGHTS. MINIMUM FACILITIES ARE AVAILABLE WHICH INCLUDE: CONTROL TOWER
(OR SOME SIMILAR CONTROL SYSTEM), SUCH AS A FLIGHT SERVICE STATION (FSS) WHICH
ISSUES CLEARANCES AND ADVISORIES WHEN THERE IS NO TOWER OR THE TOWER IS NOT IN
OPERATION. THE FSS CAN ALSO HAVE A REMOTE COMMUNICATION OUTLET (RCO) OR CAN BE
COLOCATED WITH A UNICOM (AERONAUTICAL ADVISORY STATION). THE COMMON TRAFFIC
ADVISORY FREQUENCY (CTAF) ADVISES ON KNOWN TRAFFIC AROUND AIRPORTS WITH NO
TOWER. LIGHTING (MAY BE FLARE POTS, ETC.), POL AND FACILITIES FOR
ORGANIZATIONAL MAINTENANCE OR BETTER. (AAFIF)</rdfs:comment>
        <rdfs:subClassOf rdf:resource="#AirportUsage"/>
    </owl:Class>
    <owl:Class rdf:ID="ActiveMilitary">
        <rdfs:subClassOf rdf:resource="#AirportUsage"/>
        <rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">ACTIVE MILITARY AIRPORTS
CONTROLLED AND OPERATED BY MILITARY AUTHORITIES PRIMARILY FOR USE BY MILITARY
AIRCRAFT, ALTHOUGH CIVIL AIRCRAFT MAY HAVE LANDING PRIVILEGES AND/OR CONTRACTS
RIGHTS. FACILITIES ARE THE SAME AS FOR CIVIL (A)
AIRPORTS.(AAFIF)</rdfs:comment>
    </owl:Class>

    <owl:Class rdf:ID="Dafif_Airport">
        <owl:equivalentClass>
            <owl:Class>
                <owl:intersectionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="http://babyz.uni-
muenster.de/ontologies/airport-ont-ifgi.owl#Airport"/>
                    <owl:Restriction>
                        <owl:someValuesFrom rdf:resource="#AirportUsage"/>
                        <owl:onProperty>
                            <owl:ObjectProperty rdf:ID="hasUsage"/>

                        </owl:onProperty>
                    </owl:Restriction>
                    <owl:Restriction>
                        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
                        <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#hasElevation"/>
                    </owl:Restriction>
                    <owl:Restriction>
                        <owl:someValuesFrom>

                            <owl:Restriction>
                            <owl:allValuesFrom rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#Foot"/>
                            <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#hasUnitOfMeasure"/>
                        </owl:Restriction>
                    </owl:someValuesFrom>
                    <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#hasElevation"/>
                </owl:Restriction>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/ontologies/airport-ont-ifgi.owl#hasPhysicalPart"/>

                    <owl:someValuesFrom rdf:resource="http://babyz.uni-
muenster.de/gswie/dafifrunways.owl#Dafif_Runway"/>
                </owl:Restriction>
            </owl:intersectionOf>
        </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```

DAFIF Runway Ontology example

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://babyz.uni-muenster.de/gswie/dafifrunways.owl#"
    xmlns:airp="http://babyz.uni-muenster.de/ontologies/airport-ont-ifgi.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:meas="http://babyz.uni-muenster.de/gswie/measurements051014.owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://babyz.uni-muenster.de/gswie/dafifrunways.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://babyz.uni-muenster.de/ontologies/airport-
ont-ifgi.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Dafif_Runway">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://babyz.uni-
muenster.de/ontologies/airport-ont-ifgi.owl#Runway"/>

          <owl:Restriction>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
            <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#hasLength"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#hasWidth"/>
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</owl:cardinality>
          </owl:Restriction>

          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="http://babyz.uni-
muenster.de/ontologies/airport-ont-ifgi.owl#AirportID"/>
            <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/ontologies/airport-ont-ifgi.owl#hasID"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#hasLength"/>
            <owl:someValuesFrom>
              <owl:Restriction>
                <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#hasUnitOfMeasure"/>

                <owl:allValuesFrom rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#Foot"/>
              </owl:Restriction>
            </owl:someValuesFrom>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#hasWidth"/>
            <owl:someValuesFrom>
              <owl:Restriction>
                <owl:allValuesFrom rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#Foot"/>

                <owl:onProperty rdf:resource="http://babyz.uni-
muenster.de/gswie/measurements051014.owl#hasUnitOfMeasure"/>
              </owl:Restriction>
            </owl:someValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```