

This download is a preprint of an article whose final and definitive form is to be published in the International Journal of Geographical Information Science © 2008? [copyright Taylor & Francis]; The International Journal of Geographical Information Science is available online at <http://www.informaworld.com>

**RESEARCH ARTICLE**

**A semantic registry using a Feature Type Catalogue instead of ontologies to support spatial data infrastructures**

KRISTIN M. STOCK\*†‡, ROB ATKINSON§‡, CHRIS HIGGINS◇, MARK SMALL◇, ANDREW WOOLF∞, KEIRAN MILLARD∩ and DAVID ARCTURAΔ

\*†Centre for Geospatial Science, University of Nottingham. [kristin.stock@nottingham.ac.uk](mailto:kristin.stock@nottingham.ac.uk)

‡Social Change Online, UK.

§CSIRO Land and Water, Australia. [rob.atkinson@csiro.au](mailto:rob.atkinson@csiro.au)

◇EDINA, University of Edinburgh, UK. [firstname.lastname@ed.ac.uk](mailto:firstname.lastname@ed.ac.uk)

∞Science and Technology Facilities Council, UK. [a.woolf@rl.ac.uk](mailto:a.woolf@rl.ac.uk)

∩HR Wallingford, UK. [k.millard@hrwalligford.co.uk](mailto:k.millard@hrwalligford.co.uk)

ΔOpen Geospatial Consortium Interoperability Institute. [darctur@ogcii.org](mailto:darctur@ogcii.org)

*(Received 31 December 2007)*

## **Abstract**

The use of a semantically rich registry containing a Feature Type Catalogue (FTC) to represent the semantics of geographic feature types including operations, attributes and relationships between feature types is required to realise the benefits of Spatial Data Infrastructures (SDIs). Specifically, such information provides a more complete representation of the semantics of the concepts used in the SDI, and enables advanced navigation, discovery and utilisation of discovered resources.

The presented approach creates an FTC implementation in which attributes, associations and operations for a given feature type are encapsulated within the FTC, and these conceptual representations are separated from the implementation aspects of the web services that may realise the operations in the FTC. This differs from previous approaches that combine the implementation and conceptual aspects of behaviour in a web service ontology, but separate the behavioural aspects from the static aspects of the semantics of the concept or feature type.

These principles are demonstrated by the implementation of such a registry using open standards. The ebXML Registry Information Model (ebRIM) was used to incorporate the FTC described in ISO 19110 by extending the Open Geospatial Consortium ebRIM Profile for the Web Catalogue Service (CSW) and adding a number of stored queries to allow the FTC component of the standards-compliant registry to be interrogated. The registry was populated with feature types from the marine domain, incorporating objects that conform to both the object and field views of the world.

The implemented registry demonstrates the benefits of inheritance of feature type operations, attributes and associations, the ability to navigate around the FTC and the advantages of separating the conceptual from the implementation aspects of the FTC. Further work is required to formalise the model and include axioms to allow enhanced semantic expressiveness and the development of reasoning capabilities.

*Keywords:* Semantics; Registries; Spatial data infrastructure; Ontology

## **1. Introduction**

A registry (also known as a catalogue) (or set of interoperable and distributed registries) is a core component of an effective Spatial Data Infrastructure (SDI). Such a registry contains information about the resources (for example, web services and data) available within the SDI, and enables discovery and execution of these resources. Resource discovery and execution may be performed by users in an attempt to achieve a particular goal.

This paper presents research incorporating a semantically-rich Feature Type Catalogue (FTC) in a geospatial registry for an SDI. The semantically-rich FTC includes information about the feature types, their attributes and operations, and the various types of relationships between those feature types. The FTC operations are linked to the web services that implement the operations, and the relationships can be of a predefined set of types, or can be of user-defined type. Thus the FTC provides a picture of the semantics of the geographic features used in the SDI and the way that they interact with the SDI resources themselves (data sets and web services).

The inclusion of such an FTC within a geospatial registry for an SDI provides a number of benefits:

- it offers a vocabulary of terms that creators of web services and data sets can reference;
- it provides semantic information about the feature types (a lightweight ontology) used in the registry, providing assistance with the interpretation of registry contents, and offering opportunities for reasoning;
- it provides a structure that shows the relationships between feature types and thus assists with discovery of related feature types;
- it includes the operations for a feature type, thus providing extra semantic information; and
- it provides an important link between the implementation aspects of an SDI (the web services and data sets), and the expression of their semantics, but keeps the two aspects separate.

Previous research in the field of semantics and ontologies has addressed the representation and use of ontologies for geographic features (Smith *et al.* 1998, Grenon *et al.* 2004), and previous research in the field of SDIs has discussed the use of registries for information and discovery (Lutz *et al.* 2006) and the representation of semantics in registries (Dogac *et al.* 2005). The current research extends this work. The contribution of the research is to present a method for linking semantic representations of geographic features and the SDI resources that implement them. This method adopts a different model from previous research approaches that separate the semantic representation of behaviour from that of static ontological aspects, but do not separate implementation from conceptual aspects of the concepts. The proposed method separates implementation and conceptual aspects, but represents behaviour as an intrinsic part of a feature type (concept) and thus as contributing to the representation of its semantics.

The outline of the paper is as follows: Section 2 summarises related work in the areas of semantics and ontologies, SDI registries and semantic registries to provide background for the current research. Section 3 justifies the method. Section 4 describes the content of a geospatial registry in general, and introduces the content of the semantically-rich FTC within the registry. Section 5 presents the semantic registry that was implemented to demonstrate the approach described in this research. The registry created for this research was based on open standards and as such conforms to basic principles of SDI design. Finally, Section 6 evaluates the approach, discusses the outcomes of the work and describes future extension work.

## **2. Related Work**

The work described in this paper combines research from two fields: SDIs and semantics. SDI research has focussed on the elements necessary to create an SDI, including the registry, while semantics research has focussed on techniques to represent semantics and then use them to integrate and discover information. Recent research has started to explore the storage of semantic information within a registry, and is extended by the current work.

### **2.1 Spatial Data Infrastructures and Registries**

An SDI includes the data sources, systems, network linkages, standards and institutional arrangements involved in delivering spatially related information from many different sources to a set of potential users (Coleman *et al.* 1994). Such an infrastructure enables interoperability between multiple suppliers and consumers of information, as well as the actual provision of such information, and enables improved access, quality and efficiency of management of data.

Previous work with SDIs has focussed on issues of data access and availability, and many national and multinational organisations have instigated SDI implementations (for example, the European Union and the United Nations). More recent work, particularly in a research context, is attempting to automate user interaction with the SDI (Lutz *et al.* 2006, Lutz *et al.* 2007), and also to provide functionality that can automatically discover, access and invoke resources without direct user intervention (Yue *et al.* 2007).

Registries are a core component of an SDI. They provide information about the objects that exist within the SDI. Such objects include data and web services, and the information about them stored in the registry includes descriptions about the way data is structured, general metadata and quality descriptions and descriptions of what the objects mean (semantics). The inclusion of descriptions of object semantics in the registry in an appropriate form are particularly important if more advanced automation approaches are to be employed within an SDI. Such semantic information must be available to applications to enable discovery and orchestration.

### **2.2 Feature Type Catalogues, Semantics and Ontologies**

The term Feature Type Catalogue is specific to geographic information, and was defined in International Standard ISO 19110 (International Standards Organisation 2005) using concepts from International Standard ISO 19109 (International Standards Organisation 2002). Feature Type Catalogues are formal structures for representing the categories of geographic features or geographic concepts, and may provide support for any geographic information system or data sharing exercise, because they identify the concepts with which the system deals.

FTCs vary in the amount of information that they contain, as well as their semantic richness. Such richness refers to the amount of meaning they can express about the feature type concerned. Expressiveness variation may occur in the structure and in the content of the FTC:

- FTCs may take the form of a simple list of feature types with little internal structure (for example, a controlled vocabulary), while semantically-richer FTCs may take the form of a taxonomy, including hierarchical relationships. At the semantically-richest end (in terms of structure), an FTC may include other types of associations between feature types (not only hierarchical relationships).

- FTCs may also vary in the amount of information they contain about the feature types concerned. Basic FTCs include textual definitions and their attributes, while semantically-richer FTCs may also contain sets of operations that a feature type has, together with the characteristics of the attributes and operations and constraints.

FTCs can be considered to exist on a spectrum of semantic richness. As additional elements are added to FTCs, they move closer to full ontologies on the semantic spectrum (McGuinness 2003). When containing the full set of elements described in this paper (flexible and extensible types of relationships, operations, attributes and relationships between operations and attributes for a given feature type), they may be considered a lightweight ontology. At this point, they differ mainly from heavy weight ontologies in the ability to express axioms and the level of mathematical formality in the definitions. Elements within FTCs are often expressed with natural language or Unified Modeling Language (UML) (Object Management Group 2007), rather than the more formal ontology languages, and do not usually have clearly defined methods for reasoning attached to them (Gomez-Perez *et al.* 2004).

### **2.3 Registries and Ontologies**

Research into the expression of semantics in registries to date has addressed the mapping of OWL constructs to ebRIM registries (Dogac *et al.* 2005) and the inclusion of DAML-S web service ontologies in UDDI registries (Dogac *et al.* 2002).

Semantics have been combined with registries (but not included within them) in the spatial domain with the use of ontologies to find concepts that are related to a specified search concept, followed by searching for related services in an OGC-compliant registry (Bernard *et al.* 2004, Vogele *et al.* 2004, Klien *et al.* 2006).

More recent work on spatial semantic registries has combined a range of these different aspects by using OWL-S ontologies and related application ontologies to express semantics and then registering the ontologies in a registry using the OGC ebRIM Profile for the Web Catalogue Service (CSW) (Martell 2005). This research then adds semantic searching capabilities to identify concepts that have a subsumption relationship with the queried concept and discover and orchestrate suitable web services (Yue *et al.* 2007).

In all of these approaches, the services are separated from the domain objects of interest to varying degrees. Some approaches confine the use of ontologies to the searching process (before the registry and services are identified), while others use OWL-S, in which the link between services and domain objects is made through inputs, outputs, preconditions and effects (Elenius *et al.* 2005). Figure I illustrates this model. These approaches are usually driven by the functional needs of the systems concerned, focussing on discovery and querying of services that relate to a particular domain object using a simple link between service and domain object.

### **3. Justification of the Approach**

The current research proposes an approach to the modelling of the semantics of resources in an SDI that is an alternative to the previous approaches in that it does not use ontologies or ontology languages. The approach is instead focussed around the OGC model of feature types, and considers a feature type to be the central conceptual unit that has attributes to represent its characteristics, associations to represent its relationships with other feature types and operations to represent its behaviour. In combination, these three elements represent the semantics of the feature type. Figure II illustrates this model.

This approach is feature-centric rather than service-centric, and is thus conceptually different from the approaches taken by other researchers. It considers a web service to be an implementation of an operation that forms an integral part of the behaviour of a feature type, and thus an essential component of the semantic representation of that feature type (Rugg *et al.* 1997, Kuhn 1994). That is, the feature type stores the conceptual semantics, and the web service is simply an implementation of those semantics, with necessary binding and invocation information attached to the web service.

This approach has two benefits. Firstly, it provides an encapsulated representation of the semantics of a concept (or feature type) that includes the important behavioural aspects of those semantics. Secondly, it clearly separates conceptual semantic aspects from technical and operational aspects, which has a number of benefits from a practical perspective in the implementation of an SDI. These benefits are each discussed in more detail below.

### ***3.1 The Feature Type Model as an Encapsulated Semantic Representation***

A number of research efforts in recent years have identified the need for behaviour to be included in the semantic representation of a concept in order to give a more complete picture of the semantics of geographic features. The type of behaviour modelled has included actions and activities (Kuhn 2001, Timpf 2002), scenarios and applications of use (referred to as context) (Camara *et al.* 2000, Cai 2007) and pragmatics, including the origins, effects and uses of geographic entities (Brodaric 2007). These various approaches have all pointed to the shortcomings of static object ontologies for semantic representation, and shown that the categorisation of geographic features into concepts is often determined by the uses to which the features are put. In this way, behavioural aspects can provide rich information to aid in interpretation and use of geographic features (Kuhn 1994, Rugg *et al.* 1997).

Although the need for modelling of behaviour in information systems to enhance understanding and management is well recognised, most approaches to the representation of that behaviour use an approach that separates the behavioural from static aspects of the concepts concerned. The current research differs from these approaches in that it considers behaviour to be a fundamental component of a concept, rather than considering behaviour and static aspects of concepts as inherently different and distinct. This model of behaviour as an intrinsic part of the definition of a concept is supported by research into the foundations of knowledge and understanding. Aristotle's work on substances and accidents identifies substances (things or bodies) as carriers of accidents (qualities, events, processes) – accidents have no independent existence (Smith 1988). Grenon *et al.*'s (2004) SNAP ontology describes a number of dependent entities that exist only as part of independent entities. Dependent entities include qualities (attributes), functions (operations), roles, dispositions, powers, liabilities and conditions. Linguistics research by Dahlgren (1988) identifies the language that people use most often when describing the properties of a particular concept, and many of these dimensions relate to behaviour rather than static aspects of a concept (for example, behaviour, function, manner, operation). Linguistics research further supports the thesis that the way objects are structured is closely related to the operations that can be performed on them (Jackendoff 1983). In the knowledge representation field, Minsky's seminal work highlights the need for an intimate connection between factual and procedural knowledge, and the consequent Frames paradigm for knowledge representation reflects the idea of a prototypical concept that encapsulates several different types of information, including behavioural aspects (Minsky 1975). The model of behaviour as an intrinsic part of the definition of a concept also reflects the object oriented paradigm that is used in computer science programming and analysis and design (Worboys

1994, Rumbaugh 1995), in which components in an information system are viewed as entities with their own attributes, methods and inheritance relationships with other entities.

In addition to the theoretical support for an encapsulated model of the semantics of a concept, the approach provides a number of practical benefits from the point of view of both access and maintenance of SDI content. Access to content is enhanced through the application of a consistent model for feature semantics that encapsulates all of the feature aspects (behaviour, static attributes and associations), so there is no need to navigate through separate structures to identify the full semantic description of the feature type concerned. This offers significant benefits in terms of easy and convenient discovery of the different components that relate to a feature type, and the ability to link the attributes of a feature type to the operations that use or affect those attributes.

The approach also has practical benefits in terms of the management of SDI content. Governance of such content is of critical importance in an SDI, in which multiple parties contribute resources. The encapsulation of the feature type with its attributes, operations and associations ensures that governance of content can be divided by feature type, and different parties can be made responsible for management a feature type as a unit. In contrast, the service-centric approach separates the governance of the web service ontology from that of the static content ontology to which it relates, so management of content across an SDI is more difficult.

### **3.2 *The Feature Type Model as a Separation of Semantic and Implementation Aspects***

The encapsulated approach to the definition of the semantics of a concept in which operations are an intrinsic part of a concept differs from other approaches in that it *does not* separate behavioural aspects from static aspects of a concept. However, the proposed approach also differs from other approaches in that it *does* separate conceptual from implementation aspects of the concept. The FTC itself provides a semantic representation of the concepts of interest, while the information about the web services that implement the feature type operations is contained in another part of the registry.

One of the benefits of this separation is that the semantics of the behaviour of a feature type are not dependent on such behaviour being implemented as a web service. The existing ontology-based approaches (for example, using OWL-S or WSMO) (see Figure I) represent the behaviour of a concept as a web service and model it in a web service ontology. However, a concept's behavioural semantics still exist whether or not they are implemented. Also, a particular web service may represent only part of an operation, or only a particular role of the operation, so the representation of behaviour is limited if it is only expressed through web services.

This separation between conceptual and implementation aspects also has benefits for practical implementation in a registry. The conceptual or semantic description of objects may be best represented using mechanisms that incorporate reasoning and inference tools, while the implementation aspects may be best represented using traditional registry tools that are focussed around efficient invocation of services. Future research by the authors is exploring these possibilities.

### **3.3 *Why not Ontologies?***

The approach described in this paper does not use ontologies. This is not because ontologies themselves are not suited to the tasks described in this paper or are inherently flawed, but rather because the currently popular applications and approaches to the use of such ontologies are not consistent with a model that encapsulates

behaviour and static aspects within the same ontology. Current applications of ontologies divide ontologies up according to purpose (for example, domain ontologies, task ontologies, web service ontologies) (Gomez-Perez *et al.* 2004), but this conflicts with the model proposed herein. In particular, the current approaches separate web service ontologies (for example, OWL-S, WSML) from static ontologies to represent objects in a domain.

## **4 Semantic Registry Design**

The semantic registry described in this research is a model for the representation of semantic information in a registry. This Section describes the model in conceptual terms, describing the content and the types of structure it uses, and is independent of any implementation, while Section 5 provides a specific implementation of the model using open standards and includes more detail about how the method was implemented.

### **4.1 The Registry Content**

A registry is a repository for information that enables the management, access, discovery and use of resources within an SDI. Much of this content would not be considered semantic (in that it does not describe meaning in direct terms), and is used instead to manage and execute the information stored in an SDI. Registry content may potentially encompass a wide range of different types of information, including but not limited to the following:

- Metadata about resources available in the SDI, including their names, keywords, titles and quality statements (for example, as included in ISO 19115 (International Standards Organisation 2003)).
- References to the resources that may be accessed by SDI users. Most importantly, this includes data sets and web services, but may include a wide range of other resources as required by the SDI (for example, documents and images).
- Service bindings, describing how a web service in the SDI can be invoked. These are usually machine-readable.
- Specifications that resources use. For example, a description of the OGC Web Feature Service specification (Vretanos 2005) may be included if there are web services that adopt that specification, as it provides information about how the parameters of calls to the web service are to be interpreted.
- Profiles of specifications providing more restricted or specialised versions of specifications. In either case, the information assists in interpretation and execution of resources.
- Schemas (UML, XML and GML) describing the data layers included in the SDI or used by web services included in the SDI. These schemas describe the syntactic structure of the data concerned. Again, this assists the SDI in interpretation and use of the resources concerned.
- Default presentation and styling information describing values preferred by the supplier of the web service or data layer for how data items may be presented (but users are sometimes given the option to override these settings).
- User interface information, including information to allow automatic interaction with users (for example, generation of user interfaces via XSL stylesheets). This may include models for the parameters that users might be prompted for when a service is executed (for example, the styles chosen to display data in an OGC Web Map Service (de la Beaujardiere 2006)).



## **4.2 The Feature Type Catalogue**

The types of content described in Section 4.1 are included in a typical registry adhering to a registry standard like eBRIM, and include the components required to allow simple discovery and invocation of resources within an SDI. The current work extends such content with the inclusion of semantic information to provide for more advanced discovery and service chaining. The structure of the extended content reflects that of the ISO 19110 FTC, and includes feature types and their attributes, operations and associations.

Feature Types are included in the registry as a container for a collection of information about a particular class of geographic feature. This information includes at least a name and description for the feature type, but may also include attributes, operations and associations.

Each feature type may have a range of attributes representing the static characteristics of the feature type and being equivalent to adjectives in natural language terms. Examples of attributes include size, construction material, width or spatial or temporal aspects of the feature type.

Each feature type may also have a set of operations describing its behaviour, functionality or use. These operations are attached to the feature type, and may or may not be implemented as web services. If they are implemented as web services, there is a link from the operation to the web service (specifically the service binding describing how they can be executed and machine-readable), but this is an implementation link and does not form part of the semantic definition of the feature type. Operations may also be linked to the attributes that they interact with using one of the following mechanisms:

- **triggeredBy:** any change in an attribute value or the change to a specific value causes the operation to occur. For example, a flood level attribute reaching a certain value may trigger an operation to open a flood gate.
- **affects:** the operation affects the value of an attribute. For example, an operation that reseals a road may affect the road surface material attribute value.
- **observes:** the operation observes an attribute value. For example, an operation that reports the status of a tide gauge observes the last service date of the gauge.

Feature types may also have a range of associations with other feature types. Many different association types are possible, and the set of association types is extendable. The association types explicitly included in this research include:

- **inheritance/parent-child/specialisation/subTypeOf:** the feature type is a direct child (subtype) of another feature type and inherits all of its operations, attributes and associations. For example, a land parcel feature type may have a child freehold land parcel feature type. This type of relationship does not allow multiple inheritance, and the subtype feature type inherits all associations, attributes and operations of the supertype.
- **implements:** the feature type is a realisation of a more generic feature type with characteristics that apply to multiple implementing feature types. In this case, the implementing feature types inherit all of the properties (attributes, operations and associations) of the implemented feature type, and multiple inheritance (implementation) is possible. The implementing feature type inherits all associations, attributes and operations of the implemented feature type. For example, a generic feature type for the representation of spatial geometries may be implemented by a land parcel feature type. The same land parcel feature type may also implement a generic feature type for the representation of temporality.
- **equivalentTo:** the feature type is a synonym of the related feature type. For example, road and street may be considered equivalent (although this depends on the information community).

- relatedTo: the feature type shares some unspecified relationship with the related feature type.

### **4.3 *The Registry Interface***

The previous two Sections have described the content of the semantic registry at a conceptual level. Another important aspect of the semantic registry is the interface that is made available to users to access and navigate around the registry. At a conceptual level, this interface allows access to all of the components of a feature type directly by accessing the feature type (including operations, attributes, associations and links between operations and attributes), so the user need not navigate to other structures or locations to gain a full picture of the semantics of a given feature type. Furthermore, it allows associations to be navigated, with particular importance given to the sub-type and implementation associations that define the inheritance structure of the FTC (see Section 4.2). Section 5 provides a specific example of the interface that was implemented to test this method.

## **5. An Open Standards Based Registry Implementation**

To test and validate the proposed model, the semantic registry (including FTC) was implemented using the OASIS/ebXML Registry Information Model (ebRIM) (OASIS/ebXML Registry Technical Committee 2002) extended with the ISO 19110 (International Standards Organisation 2005) model to include the objects necessary for the FTC. An extension package was created for the OGC ebRIM Profile for the Web Catalogue Service (CSW) (also known informally as the Web Registry Service – WRS) to encapsulate this extended information model and the modifications to the interface necessary to allow the model to be accessed and manipulated. This Section describes the implemented registry in more detail.

### **5.1 *The Registry Information Model***

The Registry Information Model used to test the approach described in this research was based on ebRIM. This information model provides a flexible structure consisting of an extrinsic object class that may be used to represent various other classes as specialisations. This approach was taken in preference to the creation of a specialised set of intrinsic classes because the latter would have required an extension to the ebRIM model and thus customisation of existing ebRIM software packages and interfaces.

The ebRIM model was extended using the objects represented in ISO 19110 for an FTC, described in Section 4. Figure III is a UML diagram of the extended information model for the ebRIM FTC based on ISO 19110.

The core of the information model is the ebFC\_FeatureType class, which implements the Feature Type element of the FTC and acts as a container for the properties of the Feature Type. It is a specialisation of ebRIM's ExtrinsicObject class, and as such, inherits its attributes. Three of these attributes (description, lid and name) reflect the attributes that the Feature Type has in ISO 19110. The Feature Type in ISO 19110 also has an additional attribute that is modelled in ebRIM as a Slot. ebRIM's Slot class is a generic structure to add attribute information to any class in an ebRIM registry with a defined data type.

The ebFC\_FeatureType class is also a composition of the abstract ebFC\_PropertyType, a generalisation of the different types of properties: attribute, operation and association. As with ebFC\_FeatureType, it has inherited attributes and an additional slot to handle the ISO 19110 attributes not inherited.

ebFC\_FeatureAttribute and ebFC\_FeatureOperation represent attributes and operations of a Feature Type respectively, with relevant inherited attributes and slots. ebFC\_FeatureAttribute also uses a number of different classification schemes to handle attributes included in ISO 19110. ebRIM has a generic structure for handling classifications, allowing classification schemes to be created and extended to accommodate the requirements of a particular registry. In this case, classification schemes were created for attributes for which there was an enumerated set of possible values (for example, measurement units).

The information model stores a number of different associations. All associations except one (discussed below) are represented as specialisations of the ebRIM Association class. This class contains sourceObject, targetObject and associationType components. Additional attributes are accommodated by the creation of an Association class (for example, ebFC\_InheritanceRelation).

AssociationType is itself an extendable classification scheme. In this research, a limited number of the predefined ebRIM association types were used (SubTypeOf, Implements, RelatedTo, Uses, Extends, EquivalentTo, AffiliatedWith). The SubTypeOf self-referencing association on ebFC\_FeatureType is used to represent the hierarchy of feature types, and has attached to it the InheritanceRelation class that stores additional information about the inheritance. The Implements self-referencing association on ebFC\_FeatureType is used to represent the implements relationships discussed in Section 4.2, allowing multiple inheritance from generic feature types. Some additional association types were also created to handle the ISO 19110 associations, including those between operations and attributes (TriggeredBy, Affects and Observes) and the PropertyOf association. These were included by extending the ebRIM canonical association types.

There is one exception to the modelling of associations as specialisations of the ebRIM Association class. The information model presented here includes a more complex model for associations between feature types. This model allows for associations between more than two feature types, and also allows an association to be broken down into a set of roles. For example, an association between a bridge and a road running under it may be broken down into 'goesOver' and 'goesUnder' roles.

The information model also shows an association between ebFC\_FeatureOperation and the ebRIM ServiceBinding class. This association represents the link between the conceptual and implementation aspects of a feature type discussed in Section 3.2 and only exists for an operation if there are web services that implement it. The ServiceBinding class contains information about how to actually execute the web services and allows a user to use the registry to find and execute web services directly.

The UML model shown in Figure III includes only those attributes in each class that are required by ISO 19110. However, by the rules of inheritance, all attributes are inherited by parent classes, so most classes have additional inherited attributes that they may also choose to use. Also, the ebRIM model for Slots is unconstrained and is inherited by all classes as shown in the Figure III, allowing users to create a new slot for a new attribute of interest to them that does not conform to the model shown here. This is considered a drawback of the ebRIM model for the purposes of interoperability, since user-defined slots are not included in the model and will thus not be understood by other systems or users.

The registry was implemented and demonstrated using some simple feature types from the marine domain, and specifically explored the use of different packages of feature types within the FTC and the ability to represent and inherit operations, attributes and relationships between feature types from different packages.

The division of an FTC into packages is used for governance purposes, allowing different contributors to the SDI to manage different parts of the FTC.

This particular example provides an additional challenge in that it represents objects from a field view rather than an object view. The object view sees the world as a collection of objects. In this case, feature types in the marine domain might include things like oceans, beaches, estuaries and buoys. The field view sees the world as a series of values or observations at particular locations (Galton 2004). The feature type model naturally fits well with the object view of the world, because feature types correspond directly to objects (or more correctly, categories of objects). However, the feature type model may also be used to represent the field view, by considering a series of observations to be a feature type. This approach reflects that described by Grenon *et al.* (2004) in which a field view is represented as an enduring entity that is defined in terms of the geographic region with which it spatially coincides.

The example feature types used in this research included:

- AbstractFeature from the Geography Markup Language (GML) package, which represents an abstract geographic feature as a high level node in the FTC that others specialise.
- PointSeriesFeature from the Climate Science Modelling Language (CSML) package, which is a specialisation of gml:AbstractFeature and represents a generic feature type containing a series of point features (for example, a regular grid of measurements).
- TidalWaterLevelPointSeries from the Marine Overlays on Topography for Annex II Valuation and Exploitation (MOTIIVE) package, which is a specialisation of csml:PointSeriesFeature and comprises a series of point features representing tidal water level values.

Each of these feature types inherits the operations, attributes and relationships from its parent feature type, but adds properties that are specific to its own semantics. The highest level feature type, AbstractFeature, is a representation of a real world (though abstract) object, but the other feature types represent grids of measurements (the field view). Each of these feature types is a member of a different package, and each is governed by a different organisation.

## **5.2 The Registry Interface**

In addition to the information model, the WRS registry standard also includes an interface. It is this interface that ensures that the registry is interoperable, because it tells people and systems that want to access the registry what format to use and what response to expect. The WRS specification implements the registry as a web service, and the interface defines the operations that may be executed against the web service, and the expected parameters and responses for those operations. The operations cater for the gathering of information about the registry, discovery of content of the registry and publication of content to the registry.

The operations that are defined under the existing WRS specification are sufficient to allow the semantic registry to be populated and for content to be discovered, mainly because the semantic registry information model is implemented using specializations of the eBRIM information model, so no additional structures are required. However, the WRS specification allows stored queries to be defined, and as part of this work, a set of stored queries were created to allow the FTC content to be easily accessed and navigated. The stored queries were defined as follows:

- return feature type records with a specified identifier;
- return the attributes of a feature type with a specified identifier;
- return the operations of a feature type with a specified identifier;
- return the service bindings for a feature type operation with a specified identifier;

- return the attributes that are linked to a feature type operation with a specified identifier and the details of the association between the attribute and the operation;
- return the associations of a feature type with a specified identifier and their association roles, the details of the associations and the participating feature types;
- return the feature types that are subtypes of a feature type with a specified identifier (including recursive subtypes);
- return the feature types that are supertypes of a feature type with a specified identifier (including recursive supertypes);
- return the feature types that implement a feature type with a specified identifier (including recursive implementations) and
- return the feature types that are implemented by a feature type with a specified identifier (including those that are recursively implemented by the feature type).

These stored queries may then be combined to create other stored queries and return relevant subsets and combinations of information from the FTC. They exist along queries that are defined as part of the existing standards to allow users to access content from other (non-semantic) parts of the registry.

The WRS queries are formulated in XML and return XML content, allowing users to conveniently access content and interrogate or transform the responses for display or further processing.

### **5.3 *Interacting with the Registry***

Figure IV provides an example user interface to the type of information that may be represented in the FTC for the TidalWaterLevelPointSeries feature type. Descriptions may also be included, but have not been shown in this example. This interface is intended to illustrate FTC content for research purposes, not to act as a mechanism for user interaction with the FTC. In a production information system, an additional level of abstraction would be required to shield users from technical detail and present a more user friendly interface. For example, a user might be presented with network showing the feature types and their relationships, and be able to select a feature type for more information, including textual descriptions of the operations that may be executed as web services.

The attributes table identifies those attributes that are inherited from other feature types through a subTypeOf association (for example, the location attribute is inherited from PointSeriesFeature), and attributes that are applied to the feature type by virtue of a implements association (for example, deploymentDate from LifeCycleType). Attributes for which neither of these columns is populated are attributes of the feature type itself. The right-most column shows the links between the operations and the attributes and the type of link (affects, triggeredBy, observes).

The operations table can be understood in a similar way, with the addition of the linked web services. These are not part of the FTC itself, but are also stored within the registry together with invocation information. In this example, the user can click on the web service in order to invoke it, but may need to provide other input to specify parameters for the web service. These parameters reflect the arguments that are passed into a web services request via URL or XML expression (for example, parameters about the styles to display for features on a map, or selection of the specific data sets provided by a web service), and are stored in other parts of the information model (not shown here) that incorporate information about the implementation aspects of the feature types (as opposed to the conceptual aspects).

Finally, the related feature types section displays in textual form the semantic associations between this feature type and other feature types. The related feature types are hyperlinked to allow discovery of more detailed information.

## **6. Evaluation and Discussion**

The proposed model for a semantic registry and FTC including feature types that encapsulates operations as well as attributes and relationships was successfully implemented using open standards from OASIS, ISO and OGC, supporting the validity of the approach. Navigation and discovery through the different types of associations enabled easy access to related and inherited operations from associated feature types. Specifically, all of the conceptual information relating to a feature type could be returned using a single query from an encapsulated storage structure, in contrast to the need to browse separate web service and static object ontologies that would be required by other approaches. Furthermore, the implementation illustrated the ability of the method to conveniently divide governance responsibilities among different contributors to an SDI, another limitation of previous semantic registry designs.

This Section discusses some conclusions that were drawn as a result of the implementation, and identifies potential future enhancements to the approach.

### **6.1 Inheritance**

Inheritance of operations is not problematic from a conceptual point of view, but inheritance of linked web services presented some issues during implementation. The principle of inheritance implies that if a feature type inherits an operation from another feature type, then it also inherits the web services that implement the operation. However, in the case of data being exposed by a particular service instance (as, for example, in an OGC Web Feature Service (Vretanos 2005)), the inheritance of a more general web service may make available to child feature types data instances that are not appropriate. For example, a `getPointSeries` web service may return a range of data content from a wide range of themes, but only some of that content is appropriate for access under the child `WaterLevelPointSeries` feature type that inherits it. Thus for an inherited web service to be an implementation of an operation of the inheriting feature type, it is necessary to include instance-specific service information about the range of data available to determine which services (and which subsets of those services) can actually implement an operation of a child feature type effectively. Transformation or processing services that do not adhere to such restrictions are likely to offer greater opportunities in this regard. For example this problem does not restrict the OGC Web Processing Service (Shut *et al.* 2005).

This issue of inheritance of operations is accommodated conceptually by the ability of the FTC model to allow the semantics of operations to be represented at different levels of granularity. The basic structure of the FTC is an inheritance hierarchy, so very general operations may be represented at the level of a general feature type, while more specific operations may be represented further down the hierarchy. For example, an operation to merge two geographic features into a single feature may be represented at the level of `gml:AbstractFeature`, while a very specific operation like `ReturnTidalLevel` is a much lower level operation. This ability to represent an operation at any level in the FTC caters for the case in which a web service applies to a broad range of data sources, so it would be linked to a feature type further up the hierarchy than a web service that applied to only one feature type. This corresponds to the principle in activity theory in which there is a hierarchical description of the levels of 'things' that represent activities, actions and operations. Operations are more detailed than actions, and actions are more detailed than activities (Timpf 2002). The

term operation used in this paper covers the full range of these terms from activity theory, and also caters for the range of different options available in OWL-S to represent composite and atomic processes.

If this concept of varying granularity is extended, it allows the possibility for the FTC to represent very general, high level operations that are equivalent to the types of long term behaviour included in Grenon *et al.*'s (2004) SPAN ontology (for example, life cycles). Grenon *et al.*'s work clearly separates the SNAP (snapshot) and SPAN (lifecycle) ontologies from each other. The two ontologies represent different ways of viewing the world – by focussing on the object or by focussing on the lifecycle. This distinction is less clear in the FTC. As developed thus far, operations would fall within the SNAP ontology because they are seen from the perspective of the objects, but a full and complete FTC would include operations at a range of granularities, and lifecycles may then be considered from the object view, representing the combination of the two approaches. Worboys (2005) identifies the need for hybrid views of objects and processes to fully represent spatio-temporal modelling requirements and Stock (2006) develops a typology for increasingly advanced combinations of space, time and attribute information. However, Worboys (2005) also points out the conceptual problems arising from the notion of objects as elements that are subject to change over time. Such change can result in the modification of attributes, location and sometimes the very fabric of the object, so it is sometimes difficult to determine when an object continues to exist or becomes something else. Further work is required to explore the conceptual implications of the inclusion of processes at a high level of granularity in the FTC.

The gap between the conceptually appropriate inheritance of operations and the implementation difficulties of inheritance of web services in some cases may be accommodated by the use of inheritance rules attached to the 'Implements' association between the operation and the web service. Such rules would define the restrictions that apply to inheritance of a particular web service.

## **6.2 *Independent Operations and Associations***

ISO 19110 models attributes, operations and associations as properties of a feature type. While this model is valid for some operations (for example, an operation to return water level belongs to the water body that it measures) and associations (for example, a SubTypeOf association belongs to the child feature type), there are operations and associations that potentially belong to multiple features, or are generic. For example, an operation to determine a river's catchment area is a property of both the river and the topography over which the catchment exists, and cannot be determined without either of those features. Similarly, any symmetric association belongs to both participating features (for example, an adjacency relationship belongs equally to both adjacent features). This suggests that the FTC model must be expanded to allow for both independent and dependent operations and associations, and future work will address this. This is also a recognised issue in ISO 19110 and is being addressed by the committee responsible for that standard.

## **6.3 *Semantics of Operations and Process Flow***

There has been some discussion in this paper about the differences between the OWL-S method for representing web service semantics and the FTC method presented in this research. The OWL-S method currently allows the representation of semantics in the form of inputs, outputs, preconditions and results using logical languages (for example, SWRL) and referencing domain ontologies, and also allows representation of process flow. The FTC as implemented does not represent the semantics of behaviour of operations or process flow. However, methods to augment the current approach to include such semantics using linguistic primitives (Wierzbicka 1972) are currently being explored by the author.

#### **6.4 Service Chaining, Reasoning and Axioms**

Further research is also required to explore methods for using the semantic registry and FTC for enhanced discovery and automatic service chaining. These issues have not been addressed in detail thus far, and are likely to require further formalisation of the FTC as implemented in this work. Such formalisation will allow the development of reasoning tools to take advantage of the encapsulation of operations within feature types and the links between operations and attributes and then to web services. It is expected that the richness of the model will provide a number of opportunities in this area to develop additional capabilities.

Further work will also enhance the FTC model to include axioms. The model currently provides the capability to represent constraints, but these constraints are not formalised, so are of limited benefit in terms of expressing the semantics of a feature type or the relationships between feature types.

#### **6.5 A Feature Type Catalogue Ontology?**

The enhancement of the FTC in the ways described above may be conveniently done by developing a new ontology that more closely reflects the FTC than existing ontologies that are used for this purpose (for example OWL-S). Specifically, such an ontology would adopt the model that separates conceptual from implementation aspects, encapsulates behavioural and static aspects for the one object and includes the axioms and other extensions discussed above. This would allow existing reasoning tools to be used over the contents of the FTC and allow a more formal expression of the meaning of the domain concepts to enable orchestration, but would also provide the benefits of separation of implementation aspects that have been identified in this research.

### **7. Conclusions**

This research has presented a method for the representation of semantics of concepts in a registry that includes attributes, relationships between concepts and the operations of those concepts, and links the operations to the web services that implement them. The method is described in detail, and an example implementation using open standards from OASIS, ISO and OGC was used to illustrate the model.

The research describes a model for a Feature Type Catalogue that includes semantic information about concepts, and also shows how this can be implemented in a registry. The model differs from previous approaches in that it represents operations as part of the semantics of the feature type, or concept, and also separates conceptual from implementation aspects of the concepts, but stores both in a standards-compliant registry.

The proposed model provides a number of benefits over methods that have previously been used to represent behavioural semantics in SDIs (mostly focussing on OWL-S), including:

- a common formalism for the semantic relationships between instances of Feature Types and their properties (including attributes, operations and associations), enabling use of common tools to manage and reason across related implemented FTCs in different domains or jurisdictions (with separately governed SDIs);
- encapsulation of the semantic aspects of a feature type in a single structure;



- a hierarchical structure that reflects the nature of many real-world objects in the geographic domain and allows feature types to inherit a full range of elements from ancestor feature types, including inheritance of attributes, inheritance of relationships with other feature types and inheritance of behaviour;
- convenient navigation, allowing users and systems to easily traverse linkages to examine related feature types, thus providing similar benefits as tools like WordNet (Fellbaum 1998) and also allowing the same capabilities as heavy weight ontologies for discovery and search using terms that are related to a specified query term (as, for example, in Klien *et al.* 2004);
- the ability to represent behavioural semantics without requiring an implementation to be available;
- the ability to represent operations, associations and attributes at the level of granularity that best suits their scope;
- links between the conceptual to the implementation aspects of the feature type allow actual implementations of a selected operation to be easily identified and allowing users and systems to browse and interrogate concepts without having to deal with the machinery of invocation and execution;
- links between operations and related attributes providing additional semantic information that aids in interpretation of concepts as well as assisting in query building by identifying attributes of relevance for the operation;
- a mechanism for management of common semantics across multiple domains of use by the publication of a common FTC (for example, to contain representations of geometry included in standards and potentially adopted by various information communities), and domain-specific FTCs that contain specialisations of the common semantics.

Further work is required to extend the model described in this paper, including the formalisation and extension of representation and the automation of discovery and use of web services. Future combination of the work described in this paper with the semantic richness and reasoning behind ontologies is proposed by the creation of a new ontology that uses the FTC model instead of the currently popular models for the use of ontologies in SDI registries.

### **Acknowledgements**

The authors would like to thank the anonymous referees for the constructive and interesting comments that resulted in significant improvements to this paper before final submission.

### **References**

- BERNARD, L. EINSPANIER, U., HAUBROCK, S., HÜBNER, S., KLIEN, E., KUHN, W., LESSING, R., LUTZ, M. and VISSER, U., 2004, Ontology-based Discovery and Retrieval of Geographic Information in Spatial Data Infrastructures. In *Geotechnologien Science Report No 4*.
- BRODARIC, B., 2007, Geo-pragmatics for the Geospatial Semantic Web. *Transactions in GIS*, **11**, 453-477.
- CAI, G., 2007, Contextualization of Geospatial Database Semantics for Human-GIS Interaction. *Geoinformatica* **11**, 217-237.
- CAMARA, G., MONTEIRO, A.M.V., PAIVA, J. and SOUZA, R.C.M., 2000, Action driven ontologies of the geographical space: beyond the field-object debate. In *Proceedings of the 1<sup>st</sup> International Conference on GI Science*, 28-31 October 2000, Savannah, Georgia, USA.

COLEMAN, D. AND MCLAUGHLIN, J., 1994, Building a Global Spatial Data Infrastructure: Usage Paradigms and Market Influences. *Geomatica* **48**, 225-236.

DE LA BEAUJARDIERE, J., 2006,. *Web Map Service Implementation Specification*. Open Geospatial Consortium Specification 06-042.

DAHLGREN, K., 1988, *Naïve Semantics for Natural Language Understanding* (Boston: Kluwer Academic Publishers).

DOGAC, A., KABAK, Y., LALECI, G.B., MATTOCKS, C., NAJMI, F. and POLLOCK, J., 2005, Enhancing ebXML Registries to make them OWL Aware. *Distributed and Parallel Databases*, **18**, 9-36.

DOGAC, A., CINGIL, I., LALECI, G.B. and KABAK, Y., 2002, Improving the Functionality of UDDI Registries through Web Service Semantics. In 3<sup>rd</sup> VLDB Workshop on Technologies for Web Services (TES-02), 23-24 August 2002, Hong Kong, China.

ELENIUS, D., DENKER, G., MARTIN, D., GILHAM, F., KHOURI, J., SADAATI, S. and SENANAYAKE, R., 2005, The OWL-S Editor – A Development Tool for Semantic Web Services. In *The Semantic Web: Research and Applications - Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 -June 1 2005*, Lecture Notes in Computer Science Vol 3532, Gómez-Pérez, Asuncion; Euzenat, Jerome (Eds.) (Berlin: Springer), pp.78-92.

FELLBAUM, C., 1998, *WordNet: An Electronic Lexical Database* (Massachusetts: MIT Press).

GALTON, A.C., 2004, Fields and Objects in Space, Time and Space-Time. *Spatial Cognition and Computation*. **4**, 39-68.

GOMEZ-PEREZ, A., FENANDEZ-LOPEZ, M. and CORCHO, O., 2004, *Ontological Engineering*. (London: Springer-Verlag).

GRENON, P. and SMITH, B., 2004, SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition and Computation*, **4**, 69-104.

INTERNATIONAL STANDARDS ORGANISATION, 2002, *ISO 19109: Geographic information – Rules for application schema*. (Geneva: International Standards Organisation).

INTERNATIONAL STANDARDS ORGANISATION, 2003, *ISO 19115: Geographic information – Metadata*. (Geneva: International Standards Organisation).

INTERNATIONAL STANDARDS ORGANISATION, 2005, *ISO 19110: Geographic information – Methodology for feature cataloguing*. (Geneva: International Standards Organisation).

JACKENDOFF, R. (1983). *Semantics and Cognition*. (Cambridge, Massachusetts: The MIT Press).

KLIEN, E., LUTZ, M., EINSPANIER, U., and HÜBNER, S., 2004, An Architecture for Ontology-Based Discovery and Retrieval of Geographic Information. In *Proceedings of the 7th Conference on Geographic*

*Information Science (AGILE 2004)*, 29 April – 1 May, Heraklion, Greece, Toppen, F. and Painho, M. (Eds.), pp. 179-188.

KLIEN, E., LUTZ, M., and KUHN, W., 2006, Ontology-Based Discovery of Geographic Information Services - An Application in Disaster Management. *Computers, Environment and Urban Systems* **30**, 102-123.

KUHN, W., 1994, Defining Semantics for Spatial Data Transfers. In *Advances in GIS Research: Proceedings of the Sixth International Symposium on Spatial Data Handling Volume 1*, 5-9 September, Edinburgh, Scotland, Thomas C. Waugh and Richard G. Healey (Eds), pp. 973-987.

KUHN, W., 2001, Ontologies in Support of Activities in Geographic Space. *International Journal of Geographic Information Science*. **15**, 613-631.

LUTZ, M. and KLIEN, E., 2006, Ontology-based retrieval of geographic information. *International Journal of Geographical Information Science*. **20**, 233-260.

LUTZ, M. and KOLAS, D., 2007, Rule-based Discovery in Spatial Data Infrastructures, *Transactions in GIS*, Special Issue on the Geospatial Semantic Web, forthcoming.

MARTELL, R., 2005, *OGC Catalogue Services – ebRIM (ISO/TS 15000-3) profile of CSW*. Open Geospatial Consortium Specification 05-025r3.

MINSKY, M., 1975, A Framework for Representing Knowledge. In *The Psychology of Computer Vision*, P.H. Winston (Ed.) (New York: McGraw-Hill), pp. 211-277 .

MCGUINNESS, D., 2003, Ontologies Come of Age. In *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster (Eds), (Massachusetts: MIT Press).

OASIS/EBXML REGISTRY TECHNICAL COMMITTEE, 2002, *OASIS/ebXML Registry Information Model v2.0*. Approved OASIS Standard. <http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebRim.pdf>

OBJECT MANAGEMENT GROUP, 2007, *Unified Modeling Language: Superstructure Version 2.1.1*. <http://www.uml.org/#UML2.0>

RUGG, R and EGENHOFER, M., 1997, Formalizing Behavior of Geographic Feature Types. *Geographical Systems*. **4**, 159-179.

RUMBAUGH, J., 1995. OMT: The Object Model. *Journal of Object Oriented Programming*, **8**, 21-27.

SHUT, P and WHITESIDE, A., 2005, *Web Processing Service Implementation Specification*. Open Geospatial Consortium Specification 05-007r4

SMITH, B., 1988, Objects and their Environments: From Aristotle to Ecological Ontology. In *The Life and Motion of Socioeconomic Units*, Andrew Frank (Ed.), (London: Taylor and Francis).

SMITH, B., and MARK, D.M., 1998, Ontology and geographic kinds. In Proceedings, International Symposium on Spatial Data Handling, 12-15 July, Vancouver, Canada.

STOCK, K., 2006, Spatio-Temporal Data Management Using Object Lifecycles: A Case Study of the Australian Capital Territory Spatial Data Management System. *Journal of Spatial Sciences*, **51**, 43-58

TIMPF, S., 2002, Ontologies of Wayfinding: A Traveler's Perspective. *Networks and Spatial Economics*, **2**, 9-33

VOGELE, T. and SPITTEL, R., 2004, Enhancing Spatial Data Infrastructures with Semantic Web Technologies In *Proceedings of the 7th Conference on Geographic Information Science (AGILE 2004)*, 29 April – 1 May, Heraklion, Greece, Toppen, F. and Painho, M. (Eds.).

VRETANOS, P., 2005, *Web Feature Service Implementation Specification*. Open Geospatial Consortium Specification 04-094.

WIERZBICKA, A., 1972, *Semantic Primitives*. (Frankfurt am Main: Athenaum).

WORBOYS, M., 1994, Object-Oriented Approaches to Geo-Referenced Information. *International Journal of Geographical Information Science*, **8**, 385-399

WORBOYS, M., 2005, Event-oriented Approaches to Geographic Phenomena. *International Journal of Geographical Information Systems*, **19**, 1-28.

YUE, P., DI, L., YANG, W., YU, G. and ZHAO, P., 2007, Semantics-based automatic composition of geospatial Web service chains. *Computers and Geosciences*, **33**, 649-665.

Figure I: The relationship between OWL-S and domain ontologies

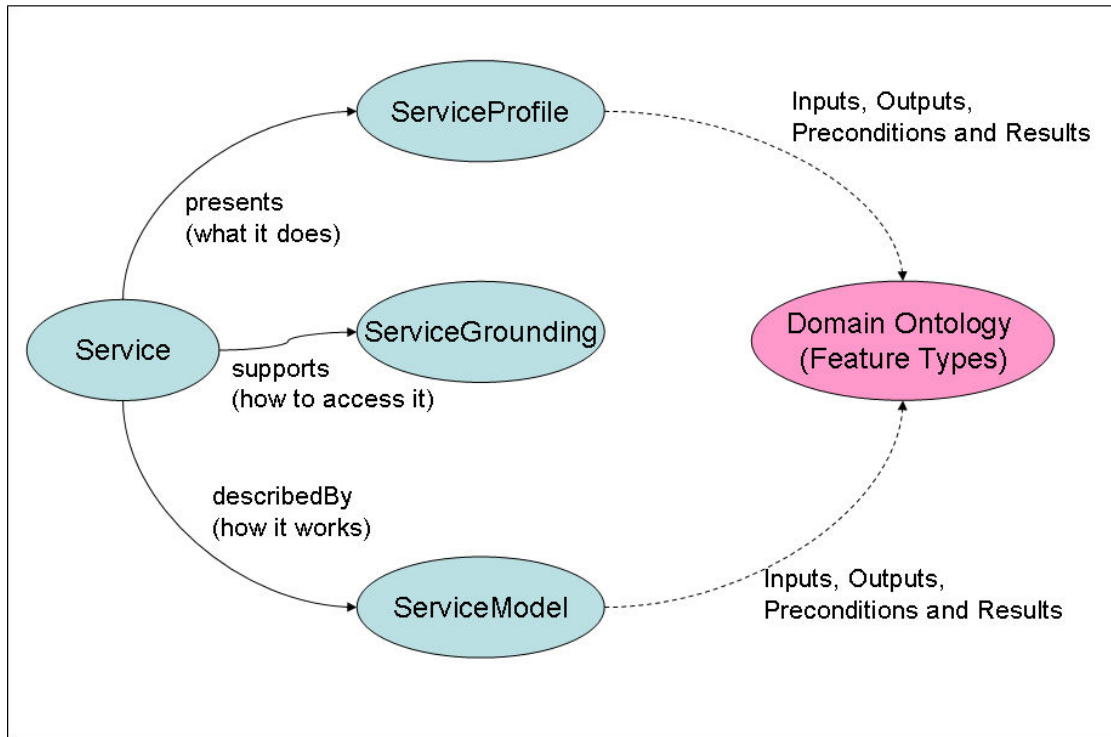
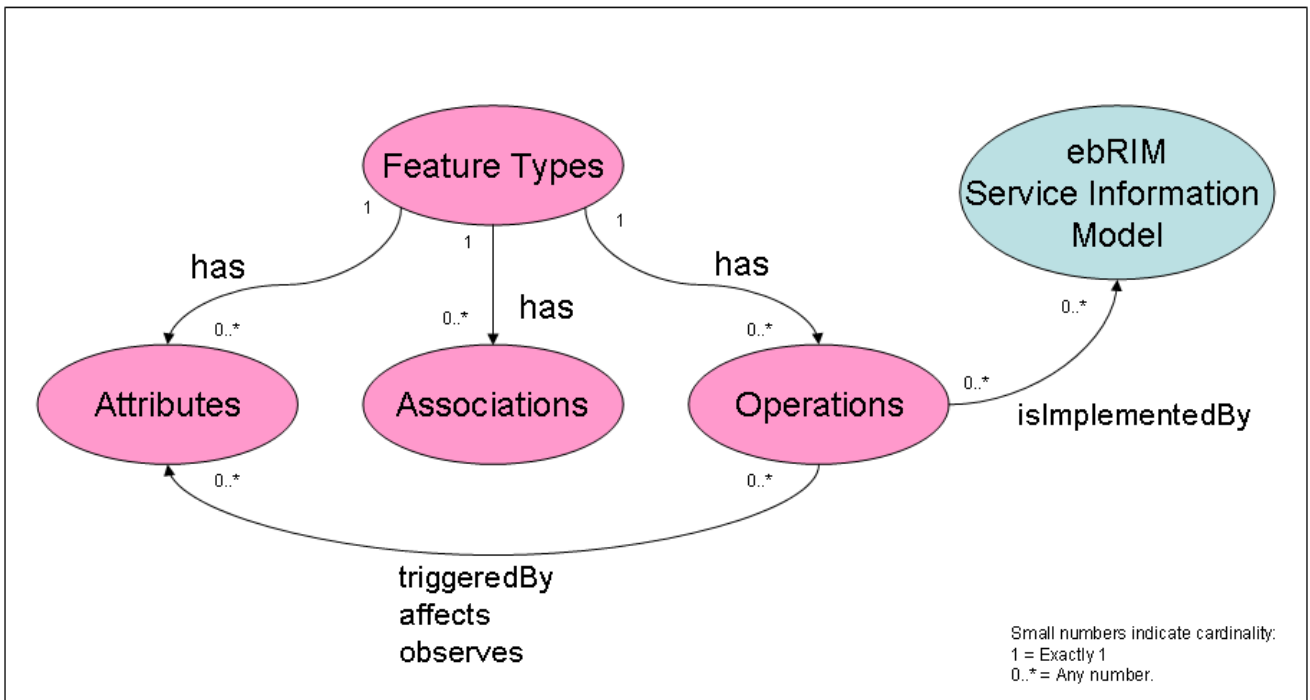


Figure II: Feature types and their relationship to operations



### Semantics in registries for SDIs

Figure III: UML diagram of the FTC registry

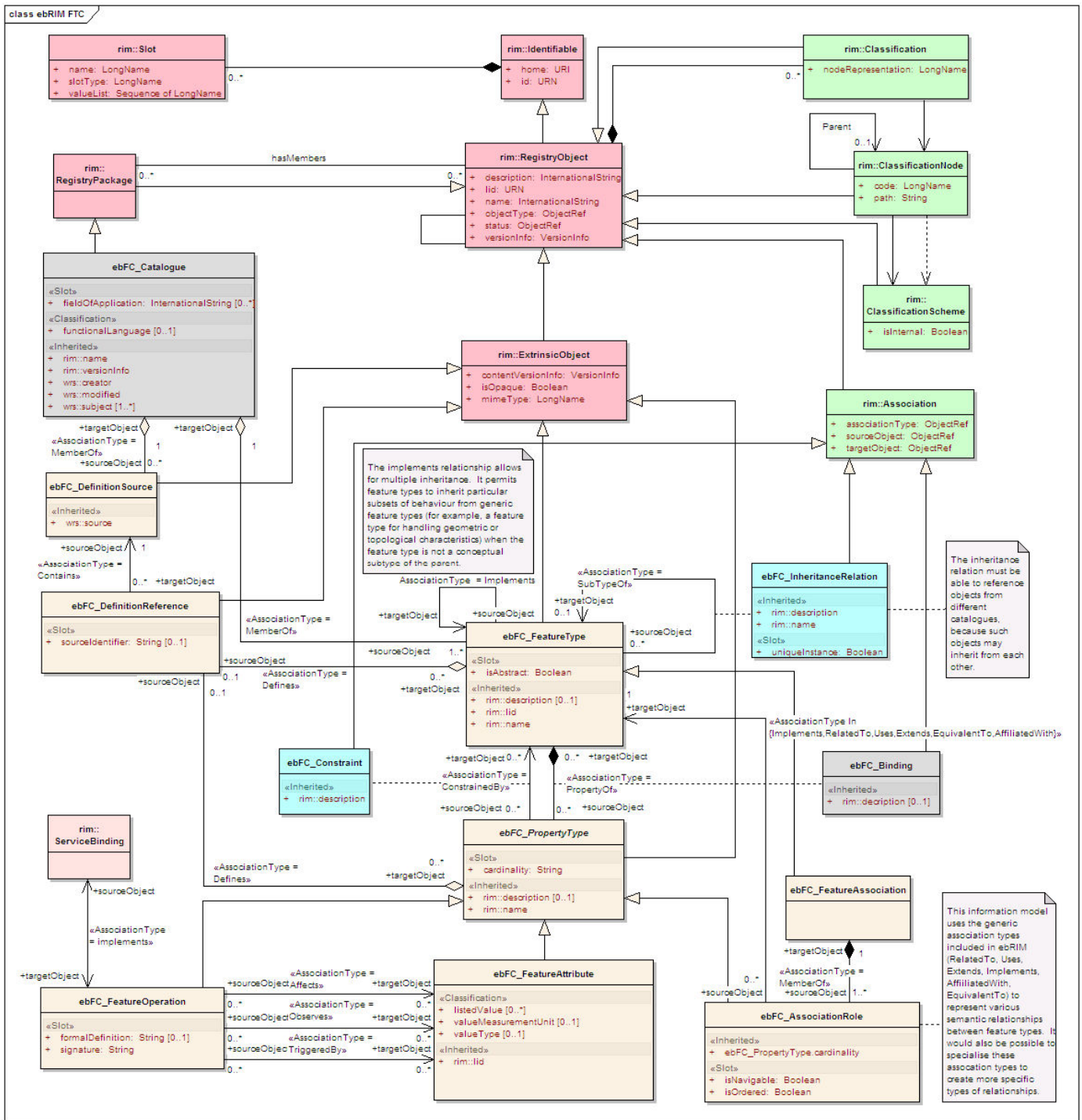


Figure IV: Example FTC content for TidalWaterLevelPointSeries

**Feature Type:** TidalWaterLevelPointSeries  
**Package:** Motiive

**Attributes:**

Attribute Name	Type(Cardinality)	Inherits	Implements	Related Operations
location value	DirectPosition(0..1) Integer	PointSeriesFeature		valueAffectedBy: ConvertTide valueObservedBy: ReturnTidalLevel valueTriggers: SendTsunamiAlert
deploymentDate	DateTime		LifeCycleType	
instrument	String			valueObservedBy: ReturnInstrumentDetails
managingAuthority	String			
monitoringStationName	String			
qualityChecked	Boolean			

**Operations:**

Operation Name	Implements	Implements	Implementing Services	RelatedAttributes
ReturnTidalLevel	PointSeriesFeature		WFS/GetFeature WMS/GetFeatureInfo	Observes: value Observes: value
DisplayTidalLevelsOnMap		VersionType	WCS/GetCoverage	
ReturnInstrumentDetails			XML/GetInstrument	Observes: instrument
PredictFutureLevels			WPS/Predict	

**Related Feature Types:**

Related by 'SubTypeOf' Associations:

Ancestors: [PointSeriesType/PointSeriesFeature](#)  
 Descendants: [UKTidalTimeSeries/NorfolkTidalTideSeries](#)  
 Siblings: [RiverFlow](#)  
[Current](#)

Related by 'Implements' Associations:

Ancestors: [LifeCycleType](#), [VersionType](#)  
 Descendants: None

Related by Other Associations:

RelatedTo: [TidalGridSeries](#)  
 Extends: [ReturnTidalLevel](#)  
 EquivalentTo: [TempsdeMaredeNormandie](#)