International Journal of Spatial Data Infrastructures Research, 2007, Vol. 2, 74-89

Editorial:

Standards Based Approaches for Cross-Domain Data Integration*

Rob Atkinson[1], Keiran Millard[2], David Arctur[3]

[1]Social Change Online (UK) & CSIRO Land and Water Division, rob@socialchange.net
[2]HR Wallingford, Howbery Park, Wallingford, Oxfordshire OX10 8BA, UK, k.millard@hrwallingford.co.uk
[3]OGC Interoperability Institute Inc., 6805 Shoal Creek Blvd, Austin, TX 78757 USA, darctur@ogcii.org

Abstract

The term "geohazards" provides a label for a common way of looking at the relationships between the state of a phenomenon, its geographical context and the impacts it may have. This way of thinking applies equally to hazards such as floods, landslides, severe weather, biological agents etc. Each of these domains must be modeled separately according to the way it behaves, but there are common problems, and in the case of geohazards, need for common views of the potential impacts and linkages. A common approach allows us to integrate data, or simply be the enabler by allowing us to share tools and methodologies.

Agreement on the commonality means "standards" - and mechanisms and governance of these standards. This paper proposes an outline of the set of standards required to achieve cross-domain data integration, and the governance arrangements required to achieve this. In particular, it proposes a potential mechanism for INSPIRE and other Spatial Data Infrastructures to achieve cross-domain harmonisation of data standard specifications through a simple generic geographic contextualisation framework that removes the need for complex cross-domain interdependencies in data models.

1 INTRODUCTION

Some problems require data from multiple sources to be synthesized. These data sources may belong to quite different domains of use, with no prior requirement

to create a common data model across all the domains relevant to a new application. It has proven difficult to achieve cross-domain interoperability, but there is an even more pressing need, to allow each domain to create some degree of intra-operability – a standardization of semantics within the domain. It turns out, however, that an understanding of cross-domain harmonisation approaches can provide useful patterns to make any domain model easier to develop and more powerful in practice.

Every exercise to develop a common data model to facilitate exchange of a data product runs into a similar set of issues. Differences in aspects such as the overall approach, level of technology specificity, levels of detail, the way of formalizing the data model, availability of related components etc means there is, to date, virtually no cross-domain interoperability achieved between such data models.

Each domain is also faced with a significant barrier arising from the lack of a common practice, tools and reusable data model components. Efforts such as INSPIRE have made an effort to improve this situation by publishing some common base elements that can be specialized (the Generic Conceptual Model[1]) and a methodology for determining the scope of a domain model[2]

Some international data modeling exercises have already been applying a similar methodology and achieving valuable consensus, such as the GeoScience Markup Language initiative[3]

ISO has published standards for the semantic building blocks of data models, and reusable components for geographic aspects[4] This set of building blocks provides a starting point for commonality in the modeling process. The intention of INSPIRE is that domain experts will be able to use these building blocks to create a coherent set of data product specifications (including data models) that can be harmonized and made interoperable.

---

[1] INSPIRE Drafting Team "Data Specifications", Deliverable D2.5: Generic Conceptual Model, 2007, URL: http://inspire.jrc.it/reports/ImplementingRules/inspireDataspecD2_5v2.0.pdf

[2] INSPIRE Drafting Team "Data Specifications", Deliverable D2.6: Methodology for the development of data specifications, 2007, URL: http://inspire.jrc.it/reports/ImplementingRules/inspireDataspecD2_6v2.0.pdf

[3] https://www.seegrid.csiro.au/twiki/bin/view/CGIModel/GeoSciML

[4] EN ISO/TS 19103:2005, Geographic Information – Conceptual Schema Language

EN ISO 19107:2005, Geographic Information – Spatial Schema

EN ISO 19108:2005, Geographic Information – Temporal Schema

EN ISO 19109:2005, Geographic Information – Rules for Application Schemas

EN ISO 19110:2006, Geographic Information – Methodology for feature cataloguing

ISO 19131:2007, Geographic Information – Data Product Specification

The challenges that remain are however very significant, complex and subtle. An analogy that reflects the complexity of this is the difference between having a selection of possible materials for a house delivered to a building site, or having an architect-designed, engineer reviewed set of plans for the house. This analogy is particularly apt, as it is the set of tried-and-trusted building standards, coupled with the experience of the builders in appropriate techniques that ensures a house can be built efficiently that meets the new owners expectations. The building design will actually be a synthesis of proven patterns, such as doorways, stairs, electrical circuits, room layouts etc.

This paper explores some of the patterns that will be required to create a set of data models that support cross-domain interoperability. It starts with an exploration of the most fundamental aspect, the role of governance, and then provides some worked examples and a suggestion for ongoing development and utilization of the concepts.

2 RELEVANT MODELLING PRINCIPLES

2.1 Relevance

This paper does not set out to explain in detail all aspects of good modeling practice, or even the application of the INSPIRE data modeling methodology. However, the solutions proposed to several key cross-domain harmonisation or interoperability challenges are grounded in best practices from data modeling frameworks. The applications of these principles are outlined below to provide insight into the underlying re-usability of the patterns identified in these solutions.

2.2 Platform Independence

The first enabler of cross-domain interoperability is to ensure that different domain models exist at the same conceptual level. It will be difficult to achieve interoperability if one domain has a conceptual model and another has a persistence model (where the model is expressed using the structures, data types and naming limitations of a particular technology).

There are at least three reasons that differences in the level of abstraction of models will hinder interoperability:

- Different domains will use different technologies, and this will result in the same concept being expressed in different ways.
- Different patterns exist in conceptual models and implementations where database denormalisation usually occurs to support specific transactions or queries.
- Technology platform bindings are less stable than conceptual models, since they may change with requirements to update technology or tune performance.

The INSPIRE methodology, which is consistent with best practices in Model Driven Architectures, aims to create a Platform Independent Model (PIM) that can be implemented in various different ways, completely or partially. This approach provides for the least complex and most useful common semantics to describe a particular domain. It follows, therefore, that harmonisation of a small number of conceptual models will also provide a simpler option for cross-domain interoperability than mapping many related, but different, partial implementations based on particular implementation platforms.

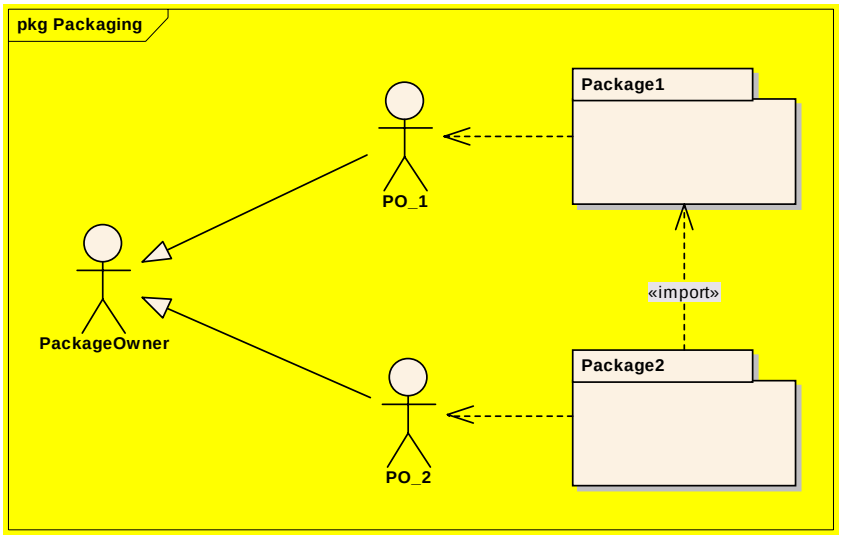
2.3 Package Modularity

Any domain model may become very complex as the level of detail being modeled increases. For example an administrative boundary may be bounded by a coastline, whose definition depends on interpretation of complex tidal phenomena.

The first principle of package modularity is to ensure that any definitions (components of the model) that are to be imported from an external domain are encapsulated in a separate package (Figure 1).

If the related domain has a reusable domain model, the package can be directly imported, otherwise the definitions should still be separately packaged to reflect the governance of the definitions and support future model harmonisation by replacing the initial implementation (Package1 in the figure) with the canonical one published by the PackageOwner.

Figure 1: Packaging imported concepts



In general it is also necessary to separate a domain into the set of definitions that are common across the domain and specialized models of object behaviour.

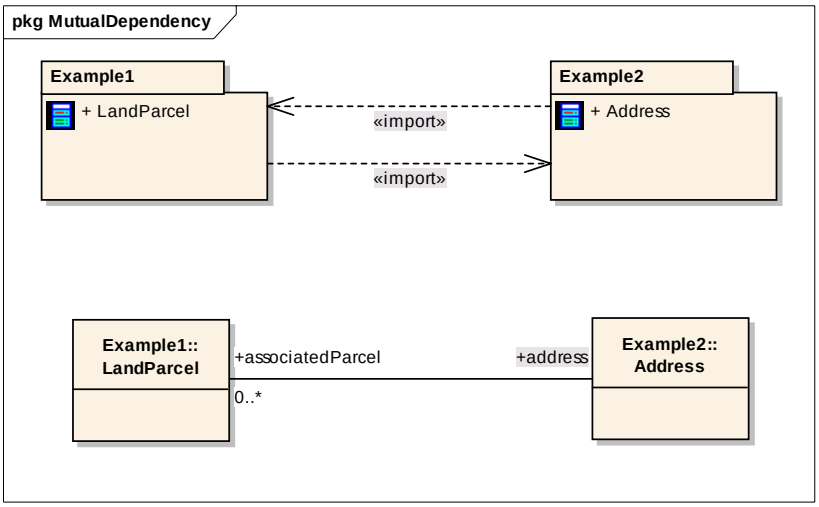
Often, however, it is necessary to capture the definitions through describing inter-feature relationships, for example the fact, that a building is located on a specific land parcel or that a road junction joins two road segments. Hence we find that the core packages of a domain describe fundamental behaviours of the domain, such as the ability to traverse from one feature to a related feature.

In this case it is often desirable to create a modular reusable package that describes common relationship patterns (e.g. Observations and Measurements [13]) and then allow different sub-domains to create client packages under their own governance arrangements. The Identity-carrier pattern described in this paper clearly separates these concerns into a meta-model (identity carrier model), domain semantics (realization of an identity carrier for the domain entity) and implementation (addition of attributes and operations for a particular use).

2.4 Avoiding Mutual Interdependence

Mutual interdependence between model packages causes significant problems with both the implementation and the governance of each package. For example, Figure 2 shows mutual interdependencies between packages created by a bi-directional association between a LandParcel and an Address.

Figure 2: Mutual dependency between packages



Neither package can be safely modified without a process that includes some form of participation from the owners of both packages to ensure that integrity is maintained. This pattern causes even more difficulty at the implementation level – it would create XML schemas with circular dependencies, that could only be loaded together, or in a database context a problem identifying which object needs to be created before the related object can be inserted.

One solution is to push such relationships into the same package. Within a database the equivalent is to enforce long transactions to maintain integrity. This approach doesn't work when no organization exists with remit over all aspects of both models. The ability to maintain cross-references is typical of the cross-domain harmonisation or interoperability problem. The solutions proposed below are heavily influenced by the pragmatic need to avoid the governance complexity of mutual interdependence.

2.5 Semantic / Structural Duality

It is often proposed to use ontology based semantics to construct cross-domain models. Ontologies formalize a set of definitions and relationships. It should be noted however that the modeling of an application schema for a domain within the UML framework used by ISO, and INSPIRE conforms to a similar underlying meta-model. UML is simply a set of relationships between objects, with a set of constraints on the nature of those relationships.

It is the authors' opinion that coherent cross-domain model harmonisation methodologies can be expressed in UML or ontological frameworks, and be transportable between the two environments.

It should be noted, however, that the UML packaging facilities and predefined meta-model provides a significant efficiency advantage as the model management environment. We expect that the patterns used for semantic reasoning and ontology mapping techniques may have value for the design of such models however. A plausible scenario might be to import a set of definitions from an ontology environment into a model and then specialize behavioural patterns to turn domain semantics into data product behaviour specifications. The behavioural patterns would also be available as an ontology to support reasoning about cross-domain integration possibilities.

3 THE PROFILE PATTERN

A common requirement is for an object in a domain to behave like a common object, but be specialized by restriction on the content it may have.

One place this pattern occurs frequently is in ISO19115 metadata patterns. The canonical schema for ISO19115 is defined by ISO19139, but this declares an object MD_Metadata in an ISO owned namespace, and each element is an optional element (i.e. its minimum cardinality is zero). In other words, ISO19115 provides a menu of possible attributes, but no implementation guidance for use of a particular subset of these within a domain.

Each domain that wishes to implement ISO19115 metadata must define a profile that specifies which attributes will be mandatory, and the value domains where common semantics of content are required. Another key requirement is to define a narrower choice of data types for each element – for example requiring that a date be machine readable format, as opposed to, for instance, a geological time period (e.g. "Jurassic").

The issue that arises is that it is difficult to create such a "structural" profile using the ISO modeling framework or the schema encoding rules to develop an XML schema. The resulting schema should be created in a namespace owned by the profile specifier, who has no logical right to publish schemas within the ISO namespaces. The end result is a schema that will not validate as a valid ISO19139 implementation, though all it really intends to be is a restriction on the contents of a valid implementation.

Detailed exploration of the issues and potential solutions to this problem are beyond the scope of this paper, however lack of a standardised solution hampers cross-domain interoperability prospects.

The profile pattern will appear in the other solution patterns below, and its relevance established by such examples. The examples below use a profiling mechanism that is a natural fit for the UML world, which is to override the types of properties of supertypes.

4 THE IDENTITY-CARRIER PATTERN

“Identity-carrier” is the name given by the authors to a pattern that has proved critical in harmonizing the different levels of abstraction in a domain models. This is especially important when reconciling common semantics of multiple implementation-oriented data models within a common conceptual model.

The basic concept is to promote to a super-class a very simple object that contains nothing beyond the common agreement that an object exists and that a particular governance authority will be recognized to designate identity of these objects. Figure 3 shows an example where different implementations of a feature, in this case a Landslide, are implemented with different geometries (to support different spatial operations), yet it is agreed that the feature is the same in each case. Figure 4 shows the role of the identity-carrier as a bridge between abstract behaviours (that can be supported by reusable software) and specific implementations. In this case, multiple related data products are available from a single sampling site defined according to the semantics of the Observations and Measurements pattern (OGC 2007a and 2007b). The identity-carrier makes it explicit that instances each data product shares common identifiers of the sampling site, and therefore these products can be interoperable when deployed through services or other packaging.

Figure 3: Identity Carrier interface between conceptual and implementation models

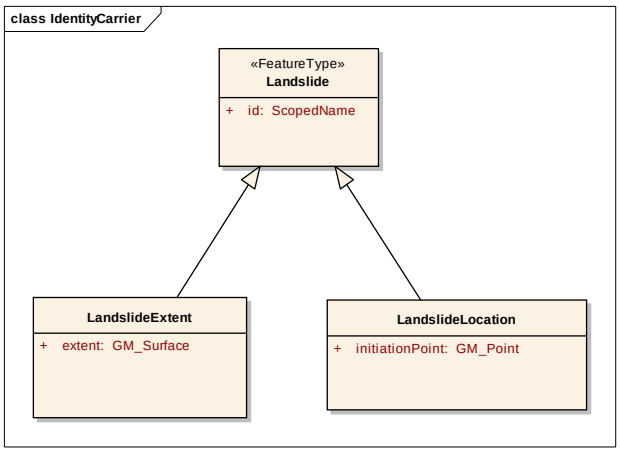
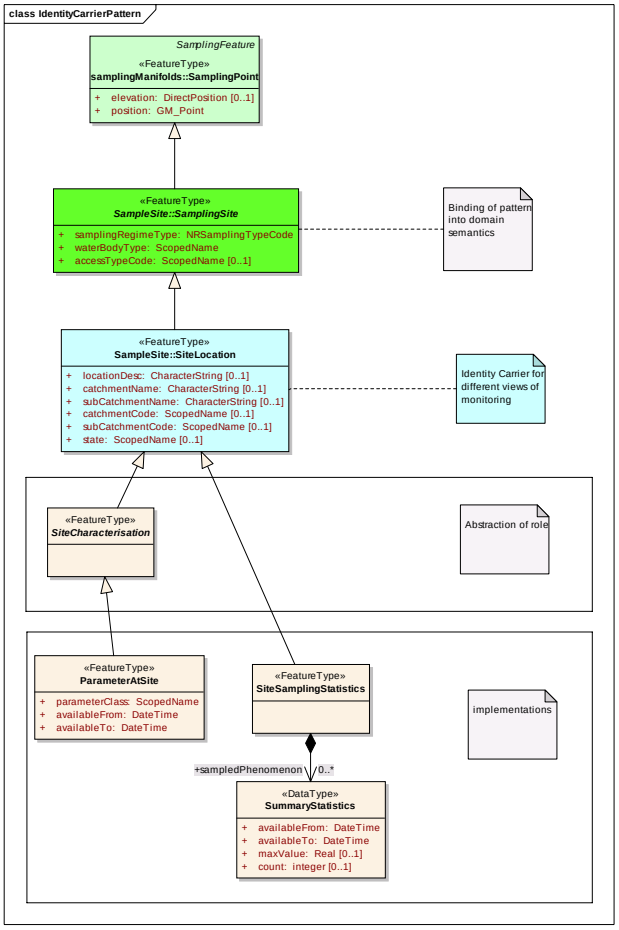


Figure 4: Role of identity-carrier in model abstraction



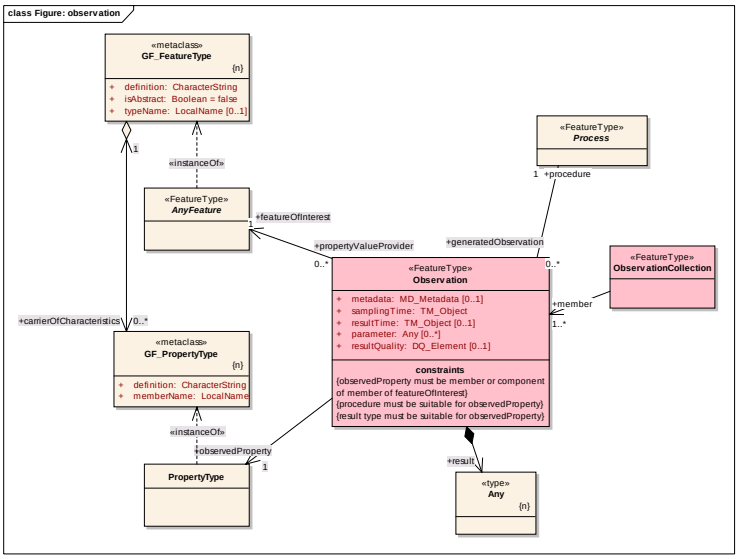
5 THE OBSERVATIONS AND MEASUREMENTS METADATA PATTERN

Cross-domain interoperability is highly complex, and the capture of all the metadata that may be required to understand, validate, discover or process data captured about a domain is extremely complex. If each domain attempts to solve such problems it will be a significant burden for the domain modelers. More significantly, however, is that each users of that domain model (and data encoded with it) will need to interpret a complex, unique model.

The Observations and Measurements (OGC, 2007a) pattern provides a building block to create consistent approaches to modeling one of the most complex problems faced by any domain. Experience shows that it is broadly applicable, but further guidance as to application is probably required to achieve ease-of-use and consistent application across domains.

Figure **5** shows the Observation class at the core of the O&M model (OGC, 2007a). Note that it has properties which are associations with other FeatureTypes. For example, the property **featureOfInterest** references an abstract object type AnyFeature, and **result** can be any data type. Clearly practical implementations will require a specific implementation of relevance to the domain, for example an observation of landslide cause would have the featureOfInterest type bound to a landslide feature, and the result would be bound to a specific classification or rating of the likely cause. Software in general is unable to implement unconstrained data types. Even soft-typing frameworks still require you to be able to recognize the objects you find, and this is the same agreement in a different part of the model.

Figure 5: Observation model



Application of O&M seems to revolve around the **profile pattern**, where the general observation pattern is constrained to apply specific result types to specific features of interest. In most cases it will also be necessary to adopt standardised specializations of each attribute and associated object for use within an implementation environment.

At the XML implementation level, for instance, the link between an Observation and a featureOfInterest may be implemented as a reference:

```
<om:featureOfInterest xlink:href="urn:some:identifier"/>
or as an inline object:
<om:featureOfInterest>
  <mydomain:SomeFeature gml:id="X">
        <gml:name codespace="urn:some_identifier_scheme">X</gml:name>
…
</mydomain:SomeFeature>
</om:featureOfInterest>
```

In either case, the level of detail or the implementation form of the related feature is not obviously required, and if the featureOfInterest was highly specialized then multiple Observation profiles would be required for each form. It is proposed that the **identity-carrier pattern** is applied for such profiles, so that the minimum implementation burden is to unambiguously identify *which* feature is referenced, not how a characterization of that feature is held in a particular system.

6 THE CONTEXTUAL SETTING PATTERN

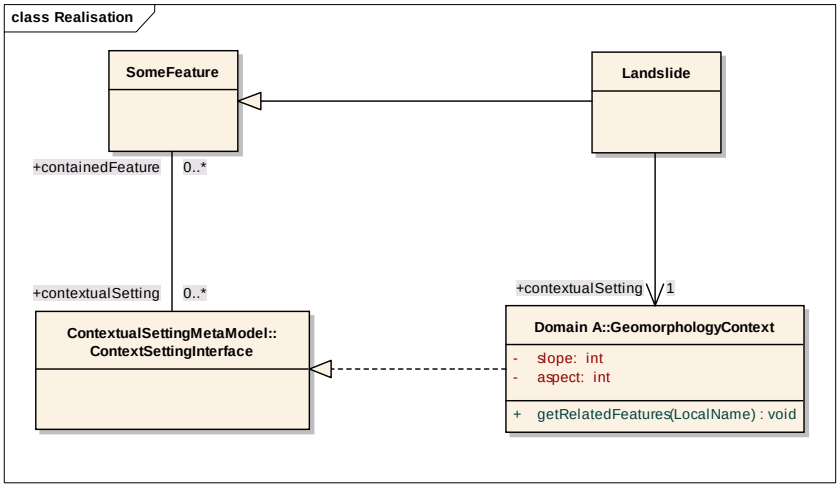
It is necessary to create an extensible mechanism to describe a feature using concepts from a related domain, whose semantics are outside the governance scope of the domain being modeled. For example, a model for landslide hazards should not be redefining basic geology, geomorphology or risk concepts.

The contextual setting pattern creates an abstraction for objects in any related domain, so that current domain model is not dependent on the details of the referenced domain. This may be necessary because:

- The related domain has not published a compatible data model;
- There is insufficient control or transparency of the modification process of the related domain model;
- There is insufficient knowledge of how to model the related domain;
- Different implementations of the current domain model may use different related domains, with no commonality justifying inclusion of any specific domain into the current model;
- Additional related information may be added to any implemented system at any time.

Figure 6 shows the basic pattern, with concrete examples of use.

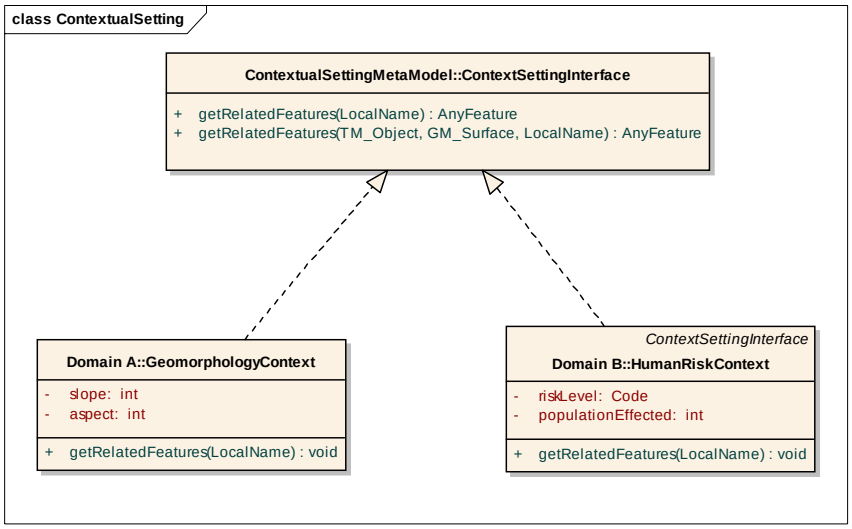
Figure 6: Contextual Setting of a Feature



Note that the relationship between the abstract objects is shown as bi-directional, though this can only be achieved within a common package (at the abstract level of SomeFeature), since it would be unwise to create a dependency from a common reusable package to a specific domain model. In practice, constraints will typically apply, such as the simple case in the example on the right, where a landslide has a single geomorphological context.

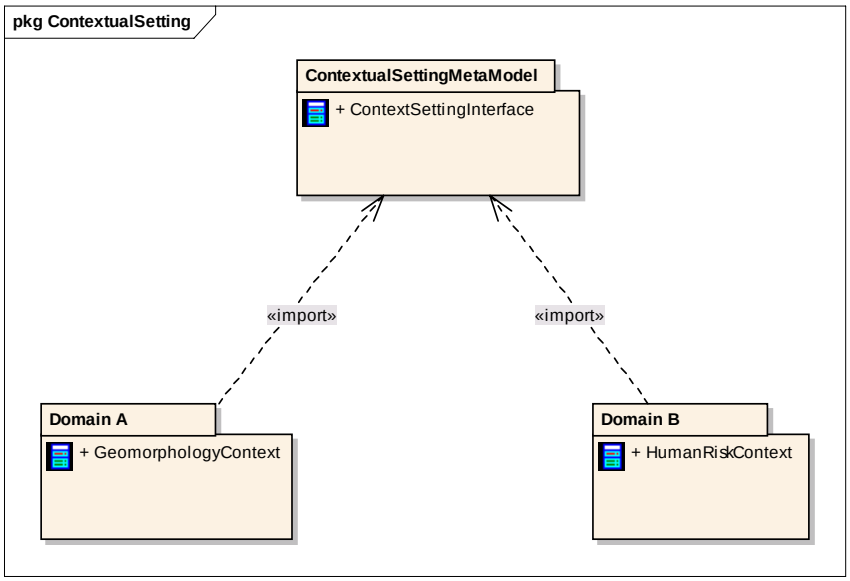
Each specific domain can then simply implement a set of classes that behave like this interface, as in Figure 7. (The exact mechanism of formalizing common behaviour can be expressed as a UML realization, but this needs further work to explore how, for example, XML schema realization of the model would be undertaken in this case).

Figure 7: Realisation of the contextual setting pattern



As a mechanism the approach is quite simple, as shown in Figure 8. Each domain simple imports the common meta-model so it can declare which features realize this function.

Figure 8: Package dependencies required for contextual setting pattern



This mechanism however highlights the need for a clear governance of the shared package, since it will be required to be available and stable as an international resource. Ultimately, it is suggested that the approaches proposed here will need to become part of the ISO 19000 suite of standards, since those standards are not readily implemented without such methodologies and common implementations.

7 FROM PRINCIPLES TO PRACTICE

The design patterns so far discussed provide a useful set of tools with which to construct complete data models. These patterns were specifically explored because of their relevance to developing multidisciplinary data models; that is, models which could be readily adapted for use across multiple communities of practice. But it is one thing to develop principles for multidisciplinary data modelling, and quite another to harmonise data models for actual use. To accomplish such a feat, it is important to consider certain factors not always taken into account in data model design.

First, there are invariably numerous ways to solve any given problem that is realistically complex, and many of the alternate approaches may be reasonably effective in addressing users' needs. However, the mere fact that two or more related communities develop different models will complicate attempts to correlate and integrate multiple different datasets. It is therefore important for groups of users representing as many related scenarios as practical to collaborate.

Second, in working to harmonise multiple data models, it is important to prototype and test each version of the design as early as possible during project development, and compare results against expectations for each of the communities of users.

Third, data model harmonisation and data sharing generally involve query and access to web-based storage systems. It is not reasonable to expect or require that all data to be shared should physically reside on a single computer or local array of storage disks. Therefore, it is very likely that a distributed, web-service-oriented architecture can and should be considered as part of the overall harmonised data model design. Working out common elements of schemas which are used by separate user communities is difficult and requires patience, discipline, and a readiness to adapt to new conditions previously not communicated.

A number of data model harmonisation projects along these lines have been conducted by various groups of scientists and software engineers. Among the

most notable are GeoSciML (geosciences markup language) and CityGML (used for urban development with full 3D models of builtup environment, surface terrain, and underground structures).

The authors' experience with data model harmonisation has shown that such projects invariably take longer than anyone wishes. It is not uncommon to take two or three years to reach a reasonably stable data model that takes into account most or all of the stakeholders' requirements. The important things are to have and keep one's vision of reaching consensus, and to find a process which enables implementing and testing that vision each step of the way.

8 CONCLUSIONS

Cross-domain interoperability introduces significant challenges that have proven difficult to meet on a systematic basis. A set of modeling approaches grounded in the basic principles of model management have been identified and illustrated with practical examples that have been successfully implemented.

The approaches can be summarized from the perspective of the governance requirements of cross-domain semantic harmonisation:

- Clear governance of reusable aspects of models, through identification of reusable packages;
- The use of profile concept to allow simplification and adoption of a small number of powerful abstract models;
- The use of an identity-carrier to unambiguously reference objects across domains without the complications of domain-specific implementation detail;
- Use of Observations and Measurements (and similar high-level abstractions) to provide extensible metadata for any domain;
- Abstraction of spatio-temporal relevance of related domains into a single interface mechanism, the contextual setting pattern, to allow independent definition of domains and pragmatic integration at run-time.

These approaches form a minimal toolkit for the creation of interoperable domain models. They also provide convenient building blocks for simplifying the process of creating domain models.

At the time of writing, these approaches have been identified as patterns that can be re-used. The need is for formalisation, publication and integration of these patterns into formal methodologies, such as INSPIRE Data Product Specification development. The next step is to undertake more extensive testing and refinement of these patterns within significant cross-domain harmonisation activities.

9 REFERENCES

OGC, 2007a: Observations and Measurements – Part 1 – Observation schema (OGC document 07-022r1)

OGC, 2007b: Observations and Measurements – Part 2 – Sampling Features (OGC document 07-002r3)