Spatially-Aware Information Retrieval on the Internet

SPIRIT is funded by EU IST Programme
Contract Number: IST-2001-35047

Design of a Geographical Ontology

Deliverable number:	D5 3101
Deliverable type:	R
Deliverable nature:	PU
Contributing WP:	WP 3
Contractual date of delivery:	Jan 1, 2003
Actual date of delivery:	Jan 31, 2003

Authors: Gaihua Fu, Alia Abdelmoty and Christopher Jones

Keywords: gazetteers, thesauri, spatial relations, spatial reasoning, mark-up languages, geographical information retrieval

Abstract: An ontology has a key role to play in the SPIRIT project with regard to providing support for query disambiguation, query term expansion, relevance ranking and web resource annotation. This document explains these functions, discusses the user requirements which influence the design of the ontology, with regard to different types of query and fundamental spatial concepts, before presenting a base model for a geographical ontology which will provide a foundation for subsequent implementation as well as experimentation with alternative ontology models. The report also reviews various ontology languages available for expressing ontologies and proposes that DAML+OIL is a suitable language for specifying the SPIRIT ontology. The use of this language for encoding the proposed base model is demonstrated.

Contents

Design of a Geographical Ontology

Draft 1.4

1. INTRODUCTION

A key aim in the SPIRIT project is the development of a search engine that displays some intelligence in the interpretation of geographical terminology. The assumption is that people may wish to find information about *something* that relates to *somewhere*. The most common way to refer to a location is to use place names, which may be qualified by spatial relationships (such as *in* or *near*). In order to assist in recognising place names and spatial relationships when they are employed in a search engine query it is proposed to employ an ontology which encodes geographical terminology and the semantic relationships between geographical terms. The idea is that the geographical ontology will enable the search engine to detect that the query refers to a geographic location and to perform a search which will result in the retrieval and relevance ranking of web resources that refer both exactly and approximately to the specified location [9]. This will entail retrieval of resources that refer to alternative versions of a specified place name as well as to places that are spatially associated with it through relations such as those of containment and adjacency. It is also proposed that an ontology should be used to assist in a process of metadata extraction whereby the geographical context of resources is determined for the purpose of search engine indexing as well as providing the potential to annotate a resource to improve its future geographical visibility.

1.1 Requirements for a SPIRIT ontology

The purpose of this document is to present a design of a geographical ontology that can be implemented in the SPIRIT project. In creating an ontology of geographic place the question arises as to what properties of place should be represented. The answer to this question is dictated here by the requirements of a spatially-aware search engine. The original SPIRIT research proposal and the SPIRIT User Requirements Specification [35] indicate that there are requirements to:

1. Recognise the presence of a place name or spatial relationship in a query or a document;
2. Find web resources that contain alternative versions of a user specified name;
3. Find web resources that refer to places that are inside or nearby to a specified location;
4. Distinguish between different types of place;
5. Perform efficient indexing of web resources to find quickly resources relating to particular regions of space;
6. Perform relevance ranking with regard to geographic space as well as to non-geographic factors.

Clearly the ontology should encode names of places and the terms that describe spatial relationships. It should be noted that places occur at multiple levels of detail ranging from continents and oceans down for example to small villages, streets, buildings, monuments and streams. The locations associated with many place names are relatively exact referring to particular cities or streets, but in natural language we often refer to places that may be more vague in their definition, such as the south of France, the Midlands or the Rocky Mountains, but which it should be possible to recognise in a spatially-aware search engine. Some places are known, or have been known, by several different names. Encoding alternative names will enable recognition of historical variants as well as modern alternatives of a place name. Ideally alternative language versions should be maintained.

When a user refers to a particular place they may be interested in exact matches or they may be interested in retrieval of resources that refer to contained or neighbouring places. The retrieval of places that are inside or nearby a named place might be achieved by explicit maintenance of qualitative spatial relations such as *inside* or *touches*, but it may also be achieved through the use of geometric data based on coordinates on the Earth's surface. The geometric representation of a place is referred to as a *footprint*. In addition to a place footprint, in this document we introduce the idea of a *query footprint* which refers to the extent in coordinates of the region the query relates to, and a *document footprint* which relates to the region or regions of space that a document refers to.

When searching geographically it may be envisaged that a user may refer to different place types such as *city* or *river*. Thus the ontology should be able to distinguish between the categories of places and to take account of the fact that many places may have multiple categories relating to social, administrative and topographic factors. It may also be useful to maintain information about category specific properties such as population or elevation. This information could then be used to assist in distinguishing between places more precisely and in the implementation of relevance ranking procedures.

For a spatially-aware search engine to be useful it must respond very quickly. Thus when a location is specified then there should be an efficient mechanism to find references to documents that relate exactly or approximately to the location. Query term expansion, to generate equivalent or related names, for submission to a textual search engine may be appropriate in some circumstances. However, for a country or a city that contains many other named places it may not be efficient to submit term-based queries with thousands of generated names. An alternative may be to index resources with respect to geometric coordinate-based space using conventional spatial indexing methods. Such an index would make use of footprint data and would be in addition to a text index.

If a search engine is to perform both exact and approximate matches to the user's query then there will be a need for geographical relevance ranking. This might be simply with regard to distance, in which case again there is a need for a geometric footprint. It might also be with regard to geographic characteristics of the place which would be encoded by spatial relationships such as of containment, adjacency and overlap with associated places.

1.2 Gazetteers, geographical thesauri and ontologies

Ontologies specify sets of concepts within a particular domain. Concepts may be named and associated with each other via sub-type/super-type relationships resulting in hierarchical structures. The term ontology has itself been used in several different ways and encompasses a range of degrees of formalisation. Sowa for example distinguishes between formal ontologies, that define concepts with axioms and logic, and terminological ontologies that may use hierarchical structures but with limited formality. Formal ontologies provide the potential to reason automatically with the concepts and terminology of a domain. The idea of encoding terminology within hierarchical relationships is found in thesauri, which have been around for a long time in the context of information retrieval. A thesaurus may be regarded as a structured vocabulary. and as such is a terminological ontology. Terms in the vocabulary are associated with each other via broader term (BT) and narrower term (NT) hierarchical relationships. A distinction can be made between different types of BT/NT relationships, including sub-type and part-of, but these distinctions are not always specified in particular thesauri and hence may restrict their use in automatic reasoning. A term in a thesaurus may be associated with one or more alternative or non-preferred terms via equivalence relations (USE and USE-FOR) and terms may be associated with other terms via the related term (RT) relationship which implies that there is some link in meaning. There are many different possible types of RT relationship.

The most often cited geographical thesaurus is the Getty Information Institute's Thesaurus of Geographic Names (TGN) [16] which is a specialisation of the general thesaurus model. For each place name the TGN maintains a unique id, a set of place types taken from the Art and Architecture Thesaurus (AAT) [15], alternative versions of the name, its containing administrative region, a footprint in the form of a point in latitude and longitude, and notes on the sources of information. Gazetteers also constitute geographic vocabularies but some of them are very limited with regard to the semantic richness. Typically a gazetteer may encode just a single name, a single place type, a point-based footprint and a parent administrative area. As such they constitute fairly crude terminological ontologies. A recent development in the realm of gazetteers is the Alexandria Digital Library (ADL) gazetteer content standard [3] which supports many types of place name metadata that may be represented either as core or optional information. This provides for a relatively rich description of place, but unlike a thesaural model there is no requirement to encode hierarchical relationships. However the facility to do so is provided, through the "Related Feature" data item, and if it is employed then there is little to distinguish the ADL from a thesaural model of place names. The ADL is associated with a Feature Type Thesaurus for specification of place types.

Recently the Open GIS Consortium has been developing a Web Gazetteer Service standard [24] for distributed access to gazetteers. Its gazetteer model is based on location instances as defined in ISO DIS 19112 [23], which are related to each other via "thesaural" hierarchical relationships to parent and child location instances.

In adopting the term geographical ontology it is intended in the SPIRIT project that we should move towards more formal ontology models with a view to exploiting the automatic reasoning that it will facilitate. This move toward formalisation is reflected later in this report in the use of the ontology language DAML/OIL [7] to encode the specified ontology design. The language, and its successor OWL [37], is associated with various editing and reasoning tools that will then become exploitable in this project.

It might be regarded as desirable for the SPIRIT ontology design to conform where possible to existing standards in gazetteers and geographical thesauri, but by the nature of the research project it may also be desirable to experiment with alternative geographical ontology designs in order to find the best design to meet the needs the SPIRIT search engine. It is of interest for example to determine the most appropriate set of spatial relationships that might be encoded between places and it is at present an open question as to the appropriate balance between the use of pre-computed spatial relationships between places and the on-line computation of relationships using the geometric footprint. With regard to some of the prominent existing geographical ontologies, the TGN design is limited by the use of only a point form footprint and the restriction to only hierarchical relations between places. The OGC Web Gazetteer Service model is also limited by the use of only hierarchical thesaural relationships between locations. The ADL may hold the potential for forming the basis of a more versatile geographical ontology, provided appropriate relationships between places are defined and used in addition to those specified in the published documentation. In all cases there is considerable scope to experiment with computational geometric and spatial reasoning techniques to exploit the stored place name information for purposes of effective information retrieval.

1.3 Domain ontologies

The focus of work on ontologies in SPIRIT concerns the modelling of geographic place. An aspect of this process is the modelling of place types. There is also an interest in modelling the terminology of one or more application areas or domains that we may use for evaluation of the SPIRIT prototype. Here we are referring to the *something* aspect of a SPIRIT query. It is expected that the modelling of place types and domain-specific terminology can be accomplished using conventional thesaural methods, i.e. without the need to introduce specialised types of relationships and category attributes. Thus equivalent terms or synonyms are represented via USE and USE-FOR relations. Hierarchical relations whether generic (is-a) or meronymic (part-of) are represented with Broader Term (BT) and Narrower Term (NT) relations, though it is appropriate to distinguish between these hierarchical types.

1.4 Objectives of ontology design in SPIRIT

Some of the main anticipated requirements of ontologies in the SPIRIT project were outlined above. The main objectives of the ontology design process are to

1. Summarise the requirements for global geographical search which the ontology should support;
2. Describe the expected role of ontologies in the SPIRIT project;
3. Review existing related research and resources for geographical ontologies;

4. Discuss the criteria that should be used in deciding upon an appropriate SPIRIT ontology design;
5. Present a design for a SPIRIT ontology that provides the basis for initial implementation and subsequent experimentation and evaluation of design alternatives;
6. Review ontology languages that may be appropriate for representing the SPIRIT ontology;
7. Highlight important implementation issues.

1.5 Report structure

The remainder of the report is organised as follows. Section 2 described the roles of ontologies in SPIRIT elaborating somewhat upon some of the introductory material of Section 1. A review of related research and resources in the area of geographical and spatial ontology is provided in Section 3. Section 4 starts with a discussion of design criteria that have been considered in producing the conceptual design for a SPIRIT ontology which is presented in Section 5. This section concludes with a summary of some of the functions that will be provided in association with the SPIRIT ontology. The issue of encoding our ontologies is addressed in Section 6 with a review of existing ontology languages and their suitability for SPIRIT. Section 7 considers a number of issues concerning the implementation of the SPIRIT ontologies. The report concludes in Section 8 with a summary of the main outcomes of the report and an indication of future work.

2. ROLES OF ONTOLOGIES IN SPIRIT

In the previous section we introduced the motivation for using ontologies in the SPIRIT project and provided an indication of the expected uses. Here we summarise the main roles that we envisage for ontologies. There are five main areas of application of ontologies in SPIRIT which are 1) user's query interpretation 2) system query formulation 3) metadata extraction; 4) relevance ranking; 5) spatial indexing.

2.1 User query interpretation

When a place name is employed in a user query, a geographical ontology will serve several purposes. It will facilitate disambiguation of the place name in the event of there being more than one place with the given name. It will also enable graphical feedback of the location which SPIRIT computes for the given location, i.e. the *query footprint* as defined by the place name and any associated spatial relations, by plotting the location on an interactive map. The query footprint could be created as a function of the query place name footprints and any associated spatial qualifiers, with regard for example to distance or direction relative to the specified place. The user could then be given the option of accepting or revising the interpretation of the extent of the location. The ontology will also be able to generate alternative names, including historical variants, which the user could accept or reject according to their interests.

Domain-specific ontologies could be used to expand non-geographical terms to include synonyms. In the event of the subject (i.e. the *something* element) of a query being itself a place type then the place type ontology could also be used to generate similar terms for purposes of query expansion.

2.2 System query formulation

The ontology could be used to generate alternative names and spatially associated names (according to spatial relationships such as inside, near or north of), which could in principle be included in a query expression to a text-based query processor. Alternatively, or as well, the ontology could be used to generate a geometric query footprint, as indicated above, which was used to access a spatially indexed database of web document metadata. Thus all documents whose own footprint intersected the query footprint could be retrieved prior to being filtered according to the textual query terms. Equally it could be that text-indexed search preceded spatial filtering (again based on the query footprint).

2.3 Metadata extraction

Ontologies could be used to identify the presence of place names, spatial qualifiers and domain-specific terminology in a free text document. If the geographical terminology was regarded as characterising the geographical context of the document, then the footprints of the respective places could be used to generate a document footprint or set of footprints that were associated with the document. This footprint metadata could be stored in the SPIRIT search engine database, or as metadata that could be attached to the original document using an annotation tool. The metadata might also include the textual place names extracted from the document in combination with the concept terms (or subjects) that were associated with them.

2.4 Relevance ranking

A geographical ontology will provide the potential for geographical relevance ranking that might be combined with non-geographical ranking. The footprints associated with documents could be used to measure the distance between the document and the query footprint in geographic coordinate space. In the case of queries that used a directional qualifier the document footprint could be used to assess geometric similarity with the interpretation of the user's query footprint, according to whether it was near its core or its periphery. It would also be possible to use other aspects of the structure of geographic space for purposes of ranking. Thus for example the similarity of the query footprint and the document footprint might be regarded as a function of the parent (containing or overlapping) regions that they had in common, and those that were non-common [9].

2.5 Spatial indexing

An important part of the SPIRIT project will be the evaluation of the use of spatial indexing of documents in addition to conventional term indexing. In order to index a document spatially it will be necessary to generate a document footprint or footprints as indicated above in the context of metadata extraction. The geometry of the footprint will then determine the location of the document in the spatial index. Using an R-Tree index for example a minimum bounding rectangle (MBR) would be placed around the document footprint (assuming it was itself more complex than a rectangle) and this MBR would determine the location of the node in the R-tree which referenced the document. In a linear quadtree index, all cells that intersected the document footprint would provide references to it.

3. RELATED RESEARCH AND RESOURCES

In this section, we review related geographic data resources in terms of the geographic information they encode. This will begin with the introduction of traditional gazetteers and then move to the modern digital gazetteer development. A summary will be given in the final part of this section to evaluate these resources in the light of the SPIRIT project.

3.1 Gazetteers

Many types of information reference specific places on the Earth's surface. A prevalent form of geo-referencing is through place or feature names. This form of geospatial reference is usually supported through the use of gazetteers. A gazetteer is a list of geographic names, together with their geographic location and other descriptive information. The essential components for a gazetteer entry are: a geographic name, a geographic location represented by coordinates, and a type designation.

Traditionally, gazetteers are produced in printed formats. Thus many world and national atlases have a gazetteer section that can be used to look up a geographic name and find the page(s) and grid reference(s) where the corresponding feature is shown. The British Ordnance Survey (OS) has published a gazetteer which lists UK national place names in alphabetical order. The information encoded in the OS gazetteer includes feature name, the County that a feature belongs to, National Grid reference, Latitude, Longitude, Feature code, Landranger map number on which the feature appears [31].

Though traditional gazetteers are useful geographic resources for SPIRIT ontology development, the information they encode and formats they are in make it difficult for them to support sophisticated applications. Various digital gazetteers have been developed that they can be embedded in information systems to support new application needs. Compared with the traditional gazetteers, the digital gazetteers are often not only richer in the geographic data they encode, but also provide various services to handle the encoded geo-data. In the UK for example, many local government authorities are adopting the British Standard BS7666 which defines gazetteers for land and property, streets, addresses and rights of way. Some digital gazetteers support the representation of geo-data in a standard form to facilitate information exchange and sharing. An internet communication protocol for distributed access is another service that may be provided by digital gazetteers, as exemplified by the ADL gazetteer service protocol and the OGC Web Gazetteer Service.

Since different gazetteers are developed for different purposes, there is a remarkable diversity in the geographic contents they encode for the same geographic space. In the remaining part of this section, we will introduce some of the most frequently referenced digital gazetteers. In order to demonstrate how these resources can contribute to SPIRIT, for each particular gazetteer, our discussion will focus on the following aspects:

- For each geographic feature, what properties are encoded to describe it. The properties will include non-spatial properties and spatial properties. For spatial properties, we are concerned with
    - how the geometric aspects of a geo-feature are described
    - what spatial relationships are supported among geographical features
- What geographical feature types are supported and what relationships between geographic feature types are supported
- Which geographical space is covered
- How many geographical features are encoded

3.2 Gazetteer Development in ADL Project

The Alexandria Digital Library (ADL) [4] has been engaged in gazetteer development since the beginning of 1994. Its implementation team combined the two major U.S. federal government gazetteers GNPS (non-U.S. names) and GNIS (U.S. names) into one gazetteer containing nearly 6 million geographic features. Notable outcomes of the ADL project were a Gazetteer Content Standard [3] and a Feature Type Thesaurus [2].

The Gazetteer Content Standard supports rich descriptions of geographic features that go beyond those of most traditional gazetteers. Major sections of the Standard (with brief descriptions) are as follows

- **featureID**, unique identifier for the feature
- **featureName**, the names for the feature
- **classSection**: the primary type of the feature
- **codeSectionType**: the code associated with the feature
- **spatialLocation**: the map location of the feature
- **streetAddressSection**: the street address of the feature
- **relatedFeature**: relationships (including spatial) of the feature with other features
- **description:** a short narrative description of the feature
- **featureData**: data about the feature, such as population or elevation
- **featureLink**: website which provides information on the features
- **supplementalNote**: note explaining unusual circumstance of the feature
- **entryMetadata**: documents about entry and modification dates

It is important to note that each of the above sections may further be divided into subsections for encoding more detailed information. For example, **streetAddressSection** is divided into **streetAddress**, **city**, **stateProvince**, **postalCode** and **country**. For space reasons in this report, we will not discuss this in detail and the complete Gazetteer Content Standard can be found in Appendix A.

Besides a content standard for gazetteer objects, ADL also developed a feature type thesaurus to be used with the ADL Gazetteer. The thesaurus contains a set of terms for categories of geographic feature types and supports the following set of relationships between feature types.

USE USE
USED FOR UF
USE With US+
USED FOR With UF+
Broader Term BT
Narrower Term NT
Related Term RT

For example, to describe the geographical feature type *city*, we can use the following Feature Type Thesaurus entry:

Feature Type Name: *city*
UF: *municipalities*, *towns*, *urban areas*
BT: *populated places*
NT: *capitals*
RT: *Metropolitan Statistical Areas*

The full list of geographical feature types supported in ADL can be found in [2].

3.3 Getty TGN

TGN (Thesaurus of Geographic Names) [16] was developed by the Getty Research Institute and is a structured geographical thesaurus containing around 1 million geographic feature names and other information about places. The TGN includes all continents, nations and major populated places of the modern political world, as well as historical places.

The focus of each TGN record is a place, represented by an unique numeric ID. Linked to the record for the place are names for the place, hierarchical positions of the place in the physical and administrative world, a single coordinate point representing the geometric location of the place, narrative notes of the place, sources for the data, and "place types," which are terms describing the role of the place (e.g., "inhabited place" and "state capital"). Names for any place can include the vernacular, English, other languages, historical names, natural order, and inverted order. Among these names, one is flagged as the preferred name, or "descriptor." Sources of the various data items are also encoded. Figure 1 shows an example record produced by TGN (with some necessary editing for reasons of space).

Record ID*:* [7000457]

Coordinates*:* Lat: 43 47 N Long: 011 15 E
Note*:* Was a Roman military center at the head of navigation on the Arno river

Hierarchical Position:
...Europe.........(continent)
.... Italia........(nation)
.... Toscana......(region)
.... Firenze.......(province)

Names:
Firenze (C,V, preferred)
Florence (C,O, prefEnglish)
Florencia (C,O)

Place Types*:*
inhabited place (C), city (C), regional capital (C), provincial capital (C), commune (administrative) (C), river settlement (C), tourist center (C)

Sources and Contributors*:*
Fiorenza.... *Companion Guide: Florence (1979), 14 [VP]*
Firenze.... *Columbia Lippincott Gazetteer (1961) [BHA]; Webster's Geographical Dictionary (1984) [GRLPA]; Times Atlas of the World (1992), 66 [VP]; Companion Guide: Florence (1979), 62 ff [FDA]*

Figure 1.TGN Record

From Figure 1, we can see that the TGN supports encoding of the following information for a geographical place:

- **ID:** an unique numeric ID of the geographical feature.
- **Place Names:** the name of a geographical feature. Names for any place can include the vernacular, English, other languages, historical names, and one is flagged as the preferred name.
- **Display Dates:** when the name was used (e.g. medieval, Roman.).
- **Name Sources**: the institution that contributed the name.
- **Place Types**: can be either physical or political feature types chosen from AAT (Art & Architecture Thesaurus developed by Getty [15]).
- **Geographic Coordinates:** map location of the feature represented by a single point.
- **Descriptive Note:** additional information about the place.
- **The Hierarchies:** displaying the places within their broader contexts.

The emphasis in the TGN is on places important for art and architecture. It supports physical features and administrative feature types.

3.4 ISO/TC 211

Cooperating with OGC (Open GIS Consortium) [28], ISO/TC 211 [24] is a standards-developing activity and its aim is to develop a family of international standards that will support the understanding and usage of geographic information. These standards specify, for geographic information, methods, tools and services for acquiring, processing, analyzing, accessing, presenting and transferring such data in digital/electronic form between different users, systems and locations.

The cooperation between OGC and ISO/TC 211 has so far led to several standards under development. ISO19112 [23] Spatial Referencing by Geographic Identifiers is one draft standard which defines the conceptual schema and guidelines for describing indirect spatial (non-coordinate) reference systems. Conceptual schemata describe models of data structures and provide the basis for further standards development. The base Feature Type SI_LocationInstance (As shown in Appendix B) proposed in ISO/TC 211 19112 supports encoding the following geographic data for each geographic feature:

- description
- name
- geographicIdentifier
- boundedby
- temporalExtent
- alternativeGeographicIdentifier
- position
- geographicExtent
- administrator
- parent
- child
- locationType
- sourceFeature

The information it encodes largely overlaps with the ADL. For example, it supports encoding the feature name, type, geometric location and hierarchical (parent/child) relationships that could be equivalent to spatial part-of relations. We will not explain them here any further, the reader is referred to the OGC document [23] relating to the Web Gazetteer Service for details.

3.5 Summary

With more and more gazetteers developed for different purposes, we are aware of the possibility of reusing these geographic resources in SPIRIT. The potential of this reuse of the available gazetteers in SPIRIT is great, but unfortunately there are the difficulties standing in its way. Most notably, is the fact that different gazetteers may be developed for different applications, and thus may not satisfy SPIRIT for the following reasons.

First, most gazetteers do not provide enough support for explicitly encoding spatial relationships, which is essential for SPIRIT. For example, most traditional gazetteers, e.g., gazetteers produced by OS [31] and NIMA [27] do not encode spatial relationships at all. The ADL gazetteer design differs somewhat from the SPIRIT idea of a geographical ontology in that it does not require explicit encoding of spatial relationships such as of overlap and adjacency, though it does allow for the possibility of encoding relationships such as containment. TGN does always encode hierarchical relationships of spatial containment (e.g. of towns within counties or provinces), but it does not encode other complex spatial relationships such as adjacency or overlap.

Secondly, most existing gazetteers support encoding geographic feature types which a particular geographic feature belongs to, e.g. *London* is of the type of *city*, but usually do not encode various relationships existing between different geographic feature types. An exception is the ADL gazetteer, in which several types of relationship are defined between geographic features. Encoding such information is important for SPIRIT since it aims initially to support European applications, which implies that in SPIRIT we need to overcome the term compatibility problems caused by political or cultural differences. For example, the Netherlands uses the term *province* to refer to a second administrative division, however, a similar type of subdivision may be referred as *county, region, state* and *canton* respectively in U.K., France, Germany and Switzerland.

Thirdly, existing gazetteers support encoding geographical features of different types in the same manner. That is, all geographic features are described by the same properties, and significant characteristics of a particular type are largely ignored. For example, *population* may be a property that should be encoded for features of type *inhabited-region*, while *street-address* may just be useful for features of the type *Building*.

Finally, existing gazetteers support encoding geographic features which are well defined, for example, *London, Paris*, but provide little support to encode imprecise geographic features, e.g. *central England*.

In the remaining part of this report, we address the above problems by proposing an ontology design for SPIRIT. As with some other geographical ontology designs, in the SPIRIT ontology not only do we consider the kind of geographic data we will admit for a geographic feature, but also consider the spatial relationships existing between different geographic features. However, in designing such an ontology, we are not aiming at encoding all the geographic information associated with a feature. Rather we bear in mind that the proposed ontology is intended to aid the specific task of intelligent information retrieval on the Internet, and thus we hope that it can fulfil this role as presented in Sections 1 and 2.

4. REQUIREMENTS ANALYSIS

Several factors must be taken into account while designing a geographical ontology for SPIRIT, as indicated in section 1 and as set out in the SPIRIT User Requirements Specification [35]. In this section we focus on the role of the ontology in the processes of search query interpretation and expansion. A typology of possible queries that may be issued at the SPIRIT interface is identified. It should be noted that these queries meet requirements relating to the user scenarios documented in [35]. This is followed by a discussion on design issues related to the various elements of a query, including the concept of Place and different types of possible spatial relationships between geographic concepts that may be processed with the SPIRIT engine. The remaining sections address the specific issues of coordinate systems, the role of time, the level of geometric and semantic detail of the ontology, and finally the question of the balance between storage of explicit and implicit spatial information.

4.1 Basic Definitions

The main distinguishing factor of SPIRIT is its search for information about a Place. Hence, queries to SPIRIT are characterised by their need to identify, either precisely or vaguely, a Place, which may be an extended region in space.

Consider the query “Find Universities near Cardiff”. First, the geographic content of the query needs to be extracted from the query expression, namely, “near Cardiff”. This expression is then interpreted and expanded using the ontology, and a new interpreted geographic content satisfying the above constraint is derived, e.g. Caerleon is recognised as one of the possible places near Cardiff. A set of expanded terms is then generated and used by the engine to find all the matching documents.

Hence, any query to SPIRIT will always be associated with a geographical context. The search engine needs to match the geographic content of the query with that of the available resources and the most relevant resources would then be returned. Possible definitions of the concepts of geographical content and geometric footprint associated with queries and documents are given below.

A reference to a geographic Place could be by its name and/or by its location. A location is either absolute or relative [25]. The type of the Place is also an important identifier in most cases which facilitates the disambiguation of Places with similar names.

A query to SPIRIT may contain references to more than one Place or Place type. Similarly, different types of references to Places may be found in web documents. In both contexts, these references needs to be interpreted, related and accumulated and the geographic content of a query and a document need to be identified. Hence, a Place reference can be either absolute or relative and defined as follows.

- Absolute Place Reference

An absolute Place reference is defined by a Place name, a Place type and an exact location as follows:

Pl-Ref-Abs = <Pl-name, Pl-type, Location>

where Location is an exact position of the Place in space. A point may be used here to define a location, but other representations are possible, e.g. a line, a minimum bounding rectangle, etc. The term Place footprint or Pl-FP will be used to denote location. Hence the above definition can be rewritten as follows:

Pl-Abs = <Pl-name, Pl-type, Pl-FP>

The following are examples of absolute Place references:

<Cardiff, City, {<xmin,ymin>, <xmax,ymax>}>
<Eiffel Tower, Monument, {<x,y>}>

- Relative Place Reference:

A Place may be defined through its spatial relationship(s) to other Places. Hence, a relative Place reference could be defined as follows:

Pl-Rlv = <Spatial Relation, Pl-name, Pl-type, Pl-FP>

Note that, in this case, a Pl-FP would normally be computed using the spatial relationship in the expression.

The following are examples of relative Place references:

<In, Zurich, City, {<x,y>}>
<Close, HauptBahnhof, Railway Station, {<x.y>}>

Hence, in general a reference to a Place may be either of the above definitions, i.e.

Pl-Ref = Pl-Abs | Pl-Rlv

A query to SPIRIT will contain one or more references to Pl-Ref. The same is true for web resources to be searched by SPIRIT. The process of query interpretation would result in the identification of the geographic content of the query as defined by the Pl-Ref(s), and similarly the process of (semantic and spatial) metadata extraction in web documents would result in the identification of the geographic content of the document as defined by its contained Pl-Ref(s). These are defined as follows.

- **Geographic Content of a Query**

The geographic content of a query, denoted, Query_GC is defined as a set of Place reference expressions as follows.

Query_GC = {Pl-Ref}

- **Geographic Content of a Web Resource**

The geographic content of a web resource, denoted, Doc_GC is defined as a set of Place reference expressions as follows.

Doc_GC = {Pl-Ref}

- **Query Footprint**

The geometric footprint of a query could be defined as a function of the footprints of the associated Pl-Ref(s) and is defined as follows.

Query_FP = {FP(Query_GC)}

- **Resource Footprint**

Similarly, the geometric footprint of a document could be defined as a function of the footprints of the associated Pl-Ref(s) and is defined as follows.

Doc_FP = {FP(Doc_GC)}

4.1 A Typology of Possible Queries in SPIRIT

In this section, the possible types of queries that SPIRIT is expected to handle are identified.
In what follows, a set of atomic query expressions is first identified which can then be used to build more complex queries and scenarios.

A basic query expression in SPIRIT will consist of a reference to:

- A Place Name, or,
- An aspatial Entity with a Relationship to a Place Name (an aspatial Entity is a non-geographic entity), or,
- An aspatial Entity with a Spatial Relationship to a Place Name, or,
- A Place Name with a Spatial Relationship to a Place Name, or,
- A Place Type with a Spatial Relationship to a Place Name, or,
- A Place Type with a Spatial Relationship to a Place Type.

A Place Name is an actual name of geographic object, e.g. Cardiff. Aspatial entities are general non-geographic objects, which may correspond to a physical or an abstract theme, subject or activity, e.g. a person, a publisher, a holiday, etc. A Relationship is an instance of an aspatial, semantic, relationship which may exist between concepts in a conceptual data model, in particular, the is-related-to relationship. A Spatial Relationship is an instance of a relationship between any types of objects in space, e.g. inside, contains and near-to. A Place Type corresponds to a class of Place Names, e.g. City, Town, River and Restaurant.

In what follows, Pl-name is used to denote a Place Name, SRel is used to denote a Spatial Relation, Pl-type is used to denote a Place Type, and AS-entity is used to denote an aspatial entity and AS-Rel is used to denote a non-spatial (semantic) relation.

The set of basic types of queries to SPIRIT can be recognised as follows.

1. Find <Pl-name>:
   Cardiff

2. Find <Pl-name SRel Pl-name>:
   Queen's Buildings IN Cardiff
   Barry NEAR Cardiff

3. Find <AS-entity AS-Rel Pl-name>:
   Books on-the-subject-of (About) Edinburgh
   People with-memories-of Cardiff

4. Find <AS-entity SRel Pl-name>:
   Scottish Dance groups based IN or NEAR Edinburgh

5. Find <Pl-type SRel Pl-name>:
   Restaurants IN Cardiff.
   Hotels near Paris.
   Cities Garonne south France.
   Big Cities in Japan.
   Renting a House NEAR Cardiff.

6. Find <AS-entity SRel Pl-type>:
   Database conferences NEAR Sunny Beaches.

7. Find <AS-entity AS-Rel Pl-type>:
   Presidents of countries
   Books on the subject of rivers

8. Find <Pl-type SRel Pl-type>
   Hotels NEAR Airports.
   Airports NEAR Big Cities.

The above are atomic query expressions that may be used to generate more complex query expressions using binary logic operators and spatial operators. Hence, in processing the complex queries, atomic expressions are first extracted that correspond to one of the forms above. The following are examples of such queries and are not intended as an exhaustive set of possibilities. In what follows, Op is used to denote a logical operator, e.g. AND, OR, NOT.

- Find <(Pl-name Op Pl-name) SRel Pl-name>
  Atomic expressions:
  Find <Pl-name SRel Pl-name> OP Find <Pl-name SRel Pl-name>
  Example:
  Queen's Buildings AND Castles IN Cardiff

- Find <Pl-type SRel Pl-type SRel Pl-name>
  Atomic expressions:
  Find < Pl-type SRel Pl-name> AND < Pl-type SRel Pl-name>
  Example:
  Hotels NEAR Airports IN Cardiff
  (This query is interpreted as Hotels NEAR Airports AND IN Cardiff)
  Hotels IN Munich within a short walk from Main Station.
  Accommodation along Bicycle Paths and Rivers

- Find <Pl-type SRel Pl-name SRel Pl-name>
  Atomic expressions:
  Find < Pl-type SRel Pl-name> AND < Pl-name SRel Pl-name>
  Example:
  Hotels NEAR Castle Street IN Cardiff
  (This query is interpreted as Hotels NEAR Castle Street AND IN Cardiff)

- Find <Pl-type SRel Pl-name OR SRel Pl-name>
  Atomic expressions:
  Find < Pl-type SRel Pl-type> OR < Pl-type SRel Pl-name>
  Example:
  Hotels NEAR Cardiff Airport OR NEAR Cardiff town-centre.

- Find <Pl-name SRel Pl-name SRel Pl-name>
  Atomic expressions:
  Find < Pl-name SRel Pl-name> AND < Pl-name SRel Pl-name>
  Example:
  Queen's Buildings ON Newport Road IN Cardiff

- Find <Pl-name SRel Pl-name OR SRel Pl-name>
  Atomic expressions:
  Find < Pl-name SRel Pl-name> OR < Pl-name SRel Pl-name>
  Example:
  Tesco store IN OR NEAR Cardiff

The typology of queries suggested above is general and it does not impose any restrictions on the concept of Place or on the type of spatial relationships to be used in SPIRIT. All the query examples from the scenarios of use reported in the User Requirements Specification document [35] are supported. Some queries from the latter report have been used above as examples of query types

4.2 Design Considerations Regarding the Primary Query Elements

From the above, it can be seen that the main query constructs are: a Place Name, an aspatial Entity, a Place Type, a Relation and a Spatial Relation. In this section, an investigation of the issues related to the above constructs is presented.

There are various ways for specifying the geographic extent of a query, including an exact name, in the form of place name, a location which may be expressed interactively on a map, or a postal zone, in the form of a post-code or a zip-code. The extent may also be derived through the relation of the required object with another object or objects in the geographic space. In the later case, an interpretation of the spatial relation(s) associating the objects is necessary.

Place Name:
A place name is used to reference a particular geographic object. There may exist different names and variations of names for the same geographic object, e.g. Treforest and Trefforest. The ontology is expected to store as many as possible Place names and known alternatives, including historic names. Ideally, Place names in different languages should also be stored. The interpretation of a Place name in the interface is done by matching the name against the possible names in the ontology.

As noted earlier, Places may be referred to that may have no formal definition, such as the south of France, the Midlands or the Rocky Mountains. There are two ways to define such imprecise regions. The Places, and their associated locations, may be pre-recognised and stored explicitly in the ontology, or an interactive dialogue with the user needs to be carried out at the interface to clarify the location and/or extent of those objects. Indeed, both scenarios may be used together to confirm the correspondence between the stored and intended definitions. In both cases, the aim is to identify a precise location of such regions. If the location offered by the system is not acceptable, then users should be allowed to define the location of the regions interactively as discussed next.

Place Location:
The most natural means of specifying a location at the interface is through an interactive map that allows standard graphical manipulation, such as pointing and clicking, bounding boxes (or circles) and zooming. Users should be able to specify an exact point or region in space about which information is required.

Due to constraints related to data availability, it may not be possible to offer users maps at sufficient resolutions to enable precise choice of locations. In this case, approximate (or vague) selections are necessary. Another reason is when exact location is not known or indeed required by the users; in which case, a clarification of the required location(s) is necessary at the interface.

In the case of point selections, the search engine could match the selected point to an exact (or approximately nearest) footprint of a geographic object in the ontology. The resulting matched set would then be returned to the user for confirmation of the intended selection. Constraints may be used to guide the selection process, e.g. by asking the user to specify the object type or types required. For example, users could indicate that they are only looking for Rivers or Towns in the particular location. This filtering process would serve to constrain and enhance the search.

In the case of region selections, the search engine could match the specified region to a set of exact (or approximately nearest) footprints of objects in the ontology. In a similar fashion as above, the interface may help constrain the search by object types.

From the above, it is clear that the ontology must associate a geometric footprint with all geographic objects stored. The footprint may be approximate, e.g. a representative (centre) point or a bounding box, or more detailed, e.g. approximate shapes, or exact, i.e. a faithful representation of the object geometry This decision has direct storage implications and the benefits and limitations of the choice need to be carefully studied. Also, more than one type of geometry may be associated with the same object. For example, a region may be associated with both a representative point and a polygon, which may itself be an approximation of the actual shape.

Place Address:
The use of an address is a common form of reference to the location of a geographic object. A street name is considered to be a type of Place name as defined above. A postcode or zip-code is normally a part of an address used commonly to group sets of individual addresses or places. However, most people may recognise only a handful of exact postcodes and especially, their home or work ones. It may be desirable in some cases to issue queries to SPIRIT using particular postcodes. The codes may also serve as a constraint on query location during query interpretation and expansion.

If postcodes are to be utilised, then the interface must facilitate the input of such codes and the ontology must provide means of modelling the codes, possibly by associating them to individual places or to groups of places. A study of the effectiveness and overheads of handling postcodes needs to be carried out.

Place Type:
A Place type represents classes of geographic objects described by Place names above such as city or river. Different classifications of Place types may be used to serve different contexts of use, e.g. topographic and administrative. Hierarchical classifications of Place types are often used with sub-class and super-class relationships, e.g. a motorway is-a road. Place types may also be spatially related, most commonly by containment relationships, modelled through part-of hierarchies.

However, other types of relationships may also be possible, e.g. intersection between road objects.

Note the difference in the semantics of Part-of relationships, where spatial part-of denotes physical containment, whereas semantic part-of does not, e.g. a faculty (including a set of people) may be part-of a university but with no specific spatial relationship, while a departmental building may have a part-of relationship of physical containment within a university campus.

It may also be useful to maintain information on specific properties of Place Types to facilitate their qualification, e.g. properties to indicate size such as length, area, population, elevation, or semantic qualifier, e.g. to indicate an associated activity such as a town being a holiday resort and a city being a financial centre.

Hence, the ontology should model the concept of Place type, along with possible relationships between Types, including spatial relationships. Individual places will be considered as instances of Place Types and consequently inherit their associated properties and relationships.

Aspatial Entity:
As shown in the previous section, the use of non-geographic concepts is allowed in the expression of queries to SPIRIT. The concepts will be treated as general terms and follow standard means of interpretation as in general search engines. An ontology of domain specific concepts will need to be built (or adopted) for use in SPIRIT.

Spatial Relations:
As described above, a spatial relation is a primary means of specifying the geographic content of a query and consequently its query footprint. Users of SPIRIT must be allowed to express different types of spatial relations at the interface. A direct way of expressing a spatial relation is by its name, e.g. inside or near. However, in many cases, the specification of a relationship name is difficult, vague and imprecise. A visual approach to the expression of spatial relations at the interface may therefore be desirable. One approach is where the interface may guide the user to the choice of a particular type of spatial relation by providing a set of possible icons of relationships.

The set provided must be context sensitive and dependent on the objects or object types required in the query. Users may engage in more advanced interaction sessions with the system through a sketching interface allowing the expression of possibly more precise relations in specific contexts. A hybrid interface with a mix of textual and visual approaches is desirable.

A textual interface may either allow the free expression of spatial relations terms or constrain the choice of terms by using drop-down lists of possible terms. In the first case, an interpretation of the relation term is necessary to map it to one of the possible stored terms in the ontology. Feedback could then be sought from the user to confirm the appropriateness of the interpretation. For example, the user may enter the term “within” at the interface. The system will need to match the term and provide an equivalent term (or set of terms), e.g. inside, for the user who then confirms the choice.

The process is necessary and may need to be associated with the visual, non-textual, feedback to the user, e.g. by providing one or more possible icons to choose from. The problem stems from the variety of terminologies available for the different types of spatial relations, and the variety of possibilities of mental models of such relationships. It is therefore essential that the interface facilitates this filtering process of the relationships.

It is desirable that an ontology of spatial relations be defined in the system to allow for the interpretation of terms given at the interface. The ontology of relations should cater for the different types of spatial relations possible, namely, topological, proximal, directional and size in both quantitative and qualitative expressions.

A number of explicit types of spatial relationships between geographic objects may be stored in the ontology facilitating the interpretation and expansion of query expressions by direct matching and reasoning over spatial relations. This will have a direct storage and performance implication and, therefore, a study of the possible scenarios of stored spatial relations is necessary.

Quantitative expressions of spatial relations are also necessary, where users are allowed to express precisely the nature of the relation, e.g. 30 degs. North, or 15 m. away, etc. In this case, interpretation of the query will involve the use of the geometric footprints of the objects after a similar process of clarification and constraining of the spatial terms used. Again, the interface will play a major role in facilitating the expression of such queries, by guiding the user with interactive interface components (menus, drop-down lists, check boxes, etc.).

Note that the problem is further complicated when complex spatial relations need to be expressed, e.g. roads crossing north of a region, or within 20 km. drive south of a region, etc.
SPIRIT may choose to handle these situations, either by dividing up the expressions, and allowing users to enter one term at a time, or by providing a more sophisticated form of visual interaction. Note also the problems associated with building complex scenarios of spatial relations that are difficult to comprehend or trace. A trail of query history (possibly visual) may assist in resolving the latter issue.

4.3 Scenarios of Possible Use of Spatial Relationships in the Geo-Ontology

In the above section, the case was made for the possible storage of explicit types of spatial relationships between Places as well as between Place Types. In this section, some possible scenarios of models are presented, primarily to illustrate the variety of possible relationships and to indicate merits and overheads of use. It is to be noted, however, that a more elaborate study of the cost of storing such relationships in the ontology needs to be carried out, before finally deciding on the model to use in SPIRIT.

Four types of spatial relations can be recognised in geographic space, namely, topological, proximal, directional and size relationships. Topological relationships represent the degree of connectivity between geographic objects, e.g. whether they are overlapping or touching. Proximal relationships describe distance relations when objects are not spatially connected, e.g. whether objects are close or far. Proximal relationships are usually associated with directional or orientation relationships, e.g. near north. Different types of directional relationships are recognised, depending on the frame of reference used. *Extrinsic orientation* is used to describe direction when a fixed external frame of reference is used, for example, cardinal directions of east, west, etc. Intrinsic orientation is used when objects involved carry their own frame of reference determined by some inherent property in that object, for example, front of the house. Typical values on this reference frame are, front, back, left and right.

The above types of relations can be quantitative. For example, relationships can express the degree of proximity, e.g. x is within 50 metres of y, or, x is within 10 minutes driving from y and the degree of connectivity, e.g. x is crossing y in two points, x share 10 km. of its boundary with y, or, x has a 40 % overlap with y and so on. Similarly, orientation relationships can be quantified, e.g. x is 30 degrees north-east of y and x is within 10 km. of the front of y. Quantifying the size relationships can be absolute, e.g. x is 10m. longer than y or, relative, e.g. x is twice as large as y.

Different granularities of relationships can be defined. The types of involved objects usually constrain the type of relationship they may be involved in, in particular in the case of topological relationships, e.g. point objects may be inside region objects but may not cross them.

A notable problem with spatial relationships is that of terminology, where many terms can be used in natural language to denote the same mental model of a spatial relationships. The problem is also complicated with different treatments across languages and cultures. Representation formalisms have been proposed in the literature for different types of relations and for the hybrid use of relationships [1, 5, 13, 20].

Scenarios of use of spatial relationships are first presented, followed by a proposal for a possible ontology of spatial relations.

4.3.1 The Spaghetti Scenario

Here, no spatial relations between Places are represented explicitly in the model. All implicit relations need to be computed on the fly using geometric computation algorithms and spatial indexing facilities.

No storage implications are envisaged. The resulting ontology is equivalent to an unstructured geographic database, with all the known limitations. Note also, that the quality of relationships derived is directly related to the accuracy of representation of the geometric footprints associated with objects.

4.3.2 The Topological Structure Scenario

Here, a pre-processing stage is used to extract some topological relationships between Places using a topologically structured geometric base as commonly applied in most GIS. Different granularities of relations may be derived between Place objects as follows.

1) **Disjoint and Overlap:** This is the most general case, where Places may either be connected or disjoint. The concept of *overlap* is general in this case and encompasses any form of connectivity. The connectivity relationship can be inferred, relatively directly, from the underlying topological structure. A flag can be associated with the object to identify which other objects it is connected to. The degree of connectivity may also be stored explicitly, e.g. which boundary points the objects share. Simple connectivity queries can be directly processed in this case.

2) **Disjoint, Touch, Inside and Overlap:** Here, explicit storage of the type of connectivity between ``direct neighbours'' of Place objects in the ontology is used. A relatively expensive pre-processing stage is required to determine the exact nature of the connectivity. The three types of connectivity relationships need to be established between every object and its direct neighbours, in the first instance. It may be possible to cluster the *touch* and *overlap* relationships in a general *overlap* relationship and still achieve the same effect.

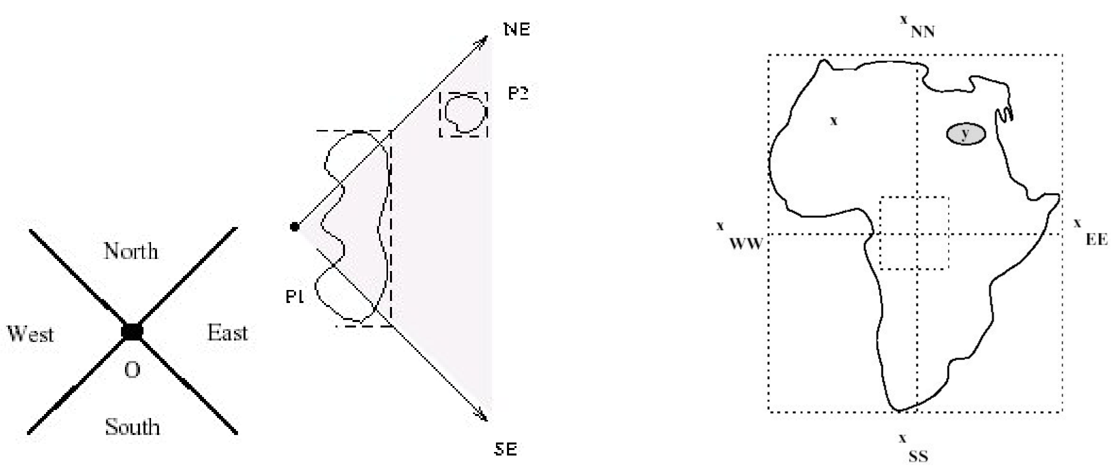
3) **Directed connectivity for linear objects:** The directed connectivity of the underlying topological structure may be propagated to the object level where ``real'' direction information can be explicitly stored with the objects, e.g. to indicate one-way roads and upstream direction of rivers.

4.3.3 The Orientation Scenario

Two forms of extrinsic orientation can be used in SPIRIT. First, there is the external extrinsic orientation, where relationships between distinct, mostly disjoint (not connected) objects are sought, e.g. countries north of France. The other type of extrinsic orientation is the internal or containment orientation relation, where relation between objects and their contained objects are sought, e.g. cities in the north of France, or Cardiff north, etc. The orientation reference frame needs to be determined apriori for the interpretation of such relationships. Many approaches exist for the definition of the orientation frames of reference [1, 10, 22, 39]. Some examples are shown in Figure 2.

As shown in Figure 2, the orientation frames of reference divides the area around the object, in the case of external orientation, or the object itself, in the case of the internal orientation, into orientation regions which could be used to determine the orientation relationship. For example, the frame of reference in Figure 2 (a) divides up the space around the object into four cardinal direction regions (East, West, North and South). The figure shows examples of possible definitions of reference frames which takes account of the object's shape and size. In Figure 2 (b) the reference frame divides the object itself into cardinal direction regions, which may be used to relate its contained objects.

Hence, it may be possible that such orientation regions (external and/or internal) be stored explicitly with the objects in the ontology to facilitate the interpretation of these types of relationships. Also, the internal (or containment) relationships may be stored explicitly in the ontology between direct parent objects, e.g. countries in continents; Egypt is in the north of Africa, and cities and countries; Cardiff is in the south of Wales, etc.



(a) (b)

Figure 2: Examples of orientation reference frames. (a) external, (b) internal.

4.3.4 The Proximity Scenario

Although it is practically impossible to store proximity relationships between all objects in the ontology, it may be feasible to define them explicitly in the model. Proximity is a fuzzy relationship. It is dependent on the type of objects, their size, as well as the context they are used in. The interface to SPIRIT could support the clarification of the semantics of such relationships, as mentioned before, by involving the user in the specification of their intended meaning.

It is however desirable for the system to assume "default" definitions to resort to. It may be possible to define default proximity zones between similar Place types depending on their size. For example, if the (approximate) radius of the region is x, then its near zone could be a buffer of a radius 1.5x, as shown in Figure 3.

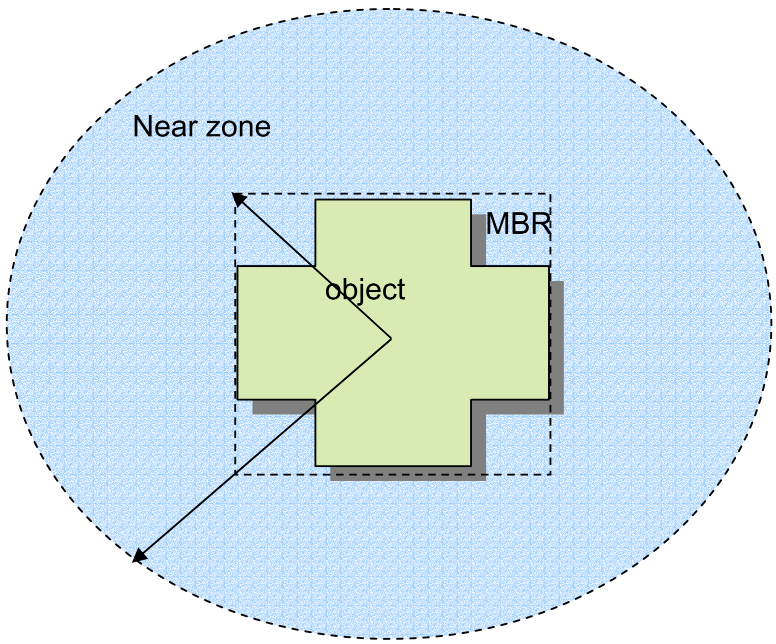


Figure 3: Possible definition of proximity zones using the type and size information of the objects.

Spatial reasoning rules can facilitate the derivation of simple proximity facts between related objects. Also, an ontology of spatial relations can be used for interpretation, e.g. touch implies close and inside implies close.

4.3.5 The Size Scenario

As in a GIS, actual size information, e.g. length and area, may be stored explicitly with the objects in the ontology. These properties are straightforward to compute in the pre-processing stage and are valuable in reasoning tasks. Other semantic size properties may also be stored explicitly, e.g. qualifying cities as big or small using population information.

4.4 An Ontology of Spatial Relations

As noted above, many terms can be used to denote one spatial relation. Also, some spatial relationships can subsume others. Hence, an ontology of spatial relations needs to be defined and associated with the model to facilitate the interpretation of queries as well as resources in the system. Table 1 shows a collection of sets of synonymous spatial terms and their equivalent "preferred" terms that may be used in the model.

Spatial Relation	Synonym
Beside	(alongside, next-to)
Near	(close, next-to)
Overlap	(intersect, cross)
Inside	(in, contained-in, within)
Disjoint	(outside, not-connected)
Touch	(adjacent, on the boundary of, next, side by side, close, abutting, adjoining, bordering, contiguous, neighbouring) [wordnet]

Table 1: Synonymous spatial terms.

In Figure 4, the subsumption relationships between spatial terms are shown. The hierarchy could be traversed in the query interpretation phase and exact or near terms extracted for query expansion.

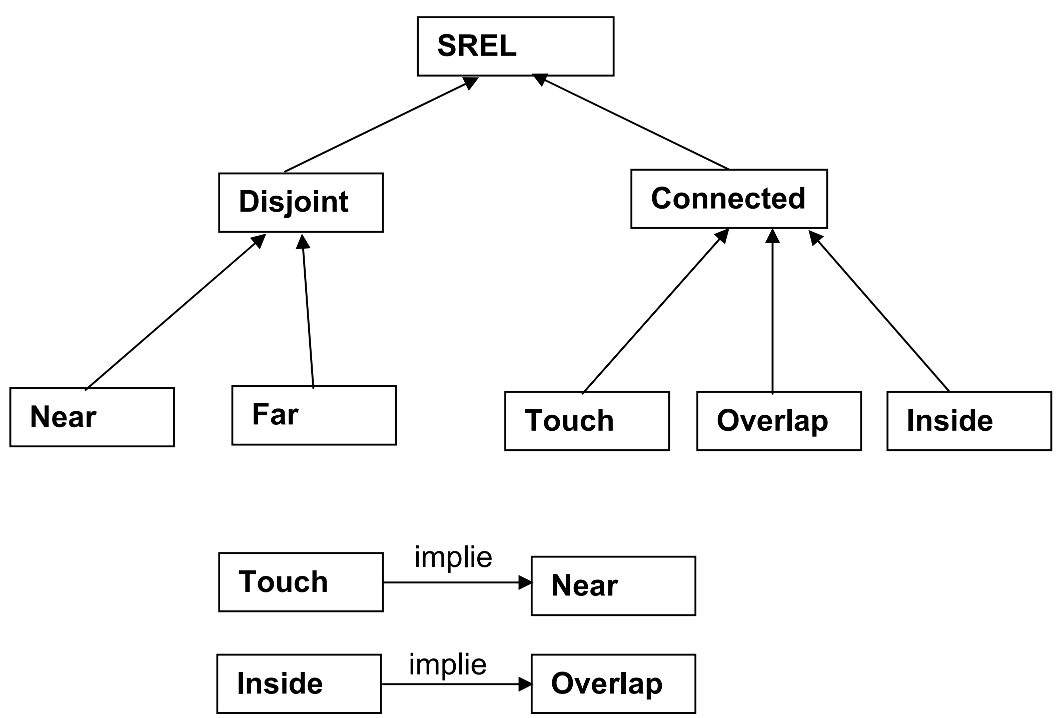


Figure 4: Subsumption relationships between spatial terms.

4.5 The Role of Spatial Reasoning

Spatial reasoning over the ontology could be used to derive implicit relationships using the explicitly stored ones. Qualitative spatial reasoning is an active research area which aims at complementing traditional computational geometry techniques by providing means of automating the derivation of implicit information in the geographic database.

Spatial reasoning exploits the properties of spatial relations, such as transitivity and symmetry in the composition process. For example, the transitivity property of the containment relationships, as well as the directional and size relations could be used to automatically traverse the ontology tree to deduce "new" spatial relationships using rules such as the following.

Inside(a,b) $\wedge$ Inside(b,c) $\Rightarrow$ Inside(a,c)

Contains(a,b) $\wedge$ Inside(b,c) $\Rightarrow$ overlaps(a,c)

East(a,b) $\wedge$ East(b,c) $\Rightarrow$ East(a,c)

Size(a) > Size(b) $\wedge$ Size(b) > Size(c) $\Rightarrow$ Size(a) > Size(c)

Distance(a,b) > Distance(b,c) $\wedge$ Distance(b,c) > Distance(c,d) $\Rightarrow$ Distance(a,b) > Distance(c,d)

The above are examples of definite composition. Qualitative spatial reasoning can also propagate indefinite relations. A study of the use of spatial reasoning and its role in enriching the ontology in SPIRIT needs to be carried out.

4.6 Coordinate systems

In view of the objective of a global geographical ontology it would appear desirable to employ a single geometric coordinate system that is global in coverage. The obvious choice is therefore the use of latitude and longitude ("geographical" as opposed to "grid") coordinates. In practice latitude and longitude are not unique as they are based on a specific geodetic datum, which denotes the dimensions of a spheroid that is used to approximate the shape of the Earth. Assuming that the SPIRIT ontology employs geographical coordinates on a specified datum, then all geometric calculations such as distance, orientation and area could be performed directly on the surface of the spheroid.

An alternative approach would be to store coordinates on the various local grid systems (e.g. the UK National Grid) used by the national mapping agencies or other data providers. This might be more efficient relative to spherical (geographical) coordinates for calculations that were confined to the geographic zone of the respective grid system, but would cause problems whenever inter-zone calculations were required (these could be done via intermediate coordinate transformations). In conclusion the simplest approach to adopt in the first place appears to be to use geographical coordinates on a specified datum. Alternative approaches could then be considered at the implementation stage.

4.7 Time

A characteristic of all geographical places is that they are embedded not just in space but also in time. Settlements and other topographic features have some time of origin (though it may not always be known) and in some cases dissolution. The names of many places have changed over time and geopolitical and natural environmental boundaries are subject to appearance, disappearance or re-location over time. Full support for spatio-temporal information is highly desirable in a geographical ontology for purposes of information retrieval, but it is also demanding and may be beyond the resources available for this project. On the assumption that some of the data resources for the SPIRIT ontology may have some temporal data relating for example to the date of establishment or duration of a place name it seems appropriate to support the storage of such data with a view to developing procedures for their exploitation if time permits in the course of implementation. It should be noted that the introduction of support for time would extend the typology of possible queries presented in section 4.1.

4.8 Language

The requirements analysis stage of SPIRIT highlighted the importance of multi-lingual support for geographical information retrieval. Just as with time, this is undoubtedly highly desirable, but its full support in the SPIRIT prototype would constitute a major overhead, due to the lack of appropriate existing multi-lingual ontology construction resources. Support for encoding alternative language versions of names is however relatively simple to provide in the ontology design as well as at the interface, and would leave the option for demonstration of some limited capacity for recognition of multi-lingual terms in the SPIRIT prototype.

4.9 Geometric Generalisation

It is well known that geographic data may be represented at multiple levels of generalisation. One aspect of generalisation concerns the level of detail with which a *specific object* is represented. Thus the areal extent of a settlement could be represented for example by a polygonal boundary with detailed sinuosity, representing a large proportion of the humanly perceptible detail. Alternatively it could be represented by a coarsely simplified polygon, a bounding rectangle or simply a representative point or centroid. These types of generalisation are examples of geometric generalisation. For reasons of data availability and usefulness, it would be impractical and also unnecessary to encode all geographic data in the SPIRIT ontology at the highest levels of geometric detail. However, in order for the geographical ontology to fulfil its roles in SPIRIT, it is desirable that it can encode geographic data, especially the geometric data, with *sufficient geometric detail*. For example, encoding the footprint with a single coordinate point might be adequate for a feature which is of type *village*, of relatively small areal extent, but might not be sufficient for a feature which is of type *country,* especially when the query expansion, relevant ranking, spatial index are considered.

4.10 Semantic Generalisation

Another aspect of generalisation concerns the semantic level of detail. Most significant topographic features can be allocated to levels of detail within semantic hierarchies or classification systems. Thus high level classes might include countries, cities, primary roads, major rivers and mountains ranges. Lower levels include counties, towns, secondary roads, tributaries and individual mountains. More detailed semantic levels could include individual buildings, side streets, minor streams and small hills. In general it can be assumed that people are interested in information at different levels of semantic detail. In order to be really useful and beneficial, the SPIRIT ontology should therefore be able to encode geographic data at multiple levels of semantic generalisation.

4.11 Explicit vs implicit maintenance of spatial data

It has been noted above that there are several types of spatial information, ranging from coordinate-based geometry, in the form of points, lines, areas and volumes, to the spatial relationships categorised as topology, proximity, orientation and size. The question arises as to what is an appropriate balance between explicit storage of spatial information and the use of online procedures to derive or deduce information from what is stored. In general the spatial relationships can be derived from the coordinate geometry data. Correct determination of topological relationships can however require detailed and accurate coordinate data and can be a computationally expensive task. Computation of the other spatial relationships can in contrast be performed with less detailed geometry. Reduction in detail of the geometry may result in a degradation of the quality of the resulting relationships but because of their numerical nature it may be argued that they can tolerate some degradation while remaining useful. This differs from the determination of topological relationships which, because of their categorical (nominal) nature, are typically either true of false.

Because of the high storage costs of detailed geometry and the associated computational costs, there is an argument for explicit storage of topological relationships between neighbouring objects. As indicated above, topological relationships between non-neighbouring objects can often be deduced reliably with spatial reasoning rules. From a computational point of view there might be a case for explicit storage of proximal, orientation and size relationships, at least between neighbouring objects. Clearly this would result in a significant storage overhead. It is also the case however that logical deduction of these relationships (apart from size) between non-neighbouring objects cannot be performed reliably, due to the often imprecise nature of the relations. The cost of explicit storage of all possible such relationships would be combinatorially explosive.

There is clearly merit in performing a detailed analysis of the relative costs of different storage strategies but this has not yet been done. Following the above considerations it appears reasonable therefore to decide initially to store geometry at variable levels of detail in addition to storing topological relationships between neighbouring spatial objects. The balance between online computation and explicit storage of other spatial relations and of more detailed geometry can then be examined at the implementation stage.

5. ONTOLOGY DESIGN FOR SPIRIT

As indicated above, in designing a geographical ontology, we are concerned with a conceptualisation of a place on the Earth. In this section, we propose a geographical ontology to be used by SPIRIT for supporting intelligent information retrieval on the Internet, we also define a set of operations to be used in accessing and query processing of geographic data stored in the ontology.

5.1 A Conceptual Design for SPIRIT Ontology

It is proposed here that the geographic ontology to be used in SPIRIT is composed of three components as shown in Figure 5. The first component is the geographical feature type ontology which encodes the various feature types which each geographic feature instance falls into. As illustrated in Figure 6, each feature type is assigned a feature type name, for example, *river* or *city*, and a resource from which the feature type comes, and one possible such resource is ADL as we discussed in Section 3.2. Feature types can relate to each other via several relationships as follows

USE – specifying which feature type is used instead of this one
UF – specifying which feature types this feature type is used for
BT -- broader feature types of this one
NT -- narrower feature types of this one
RT -- related feature types of this one

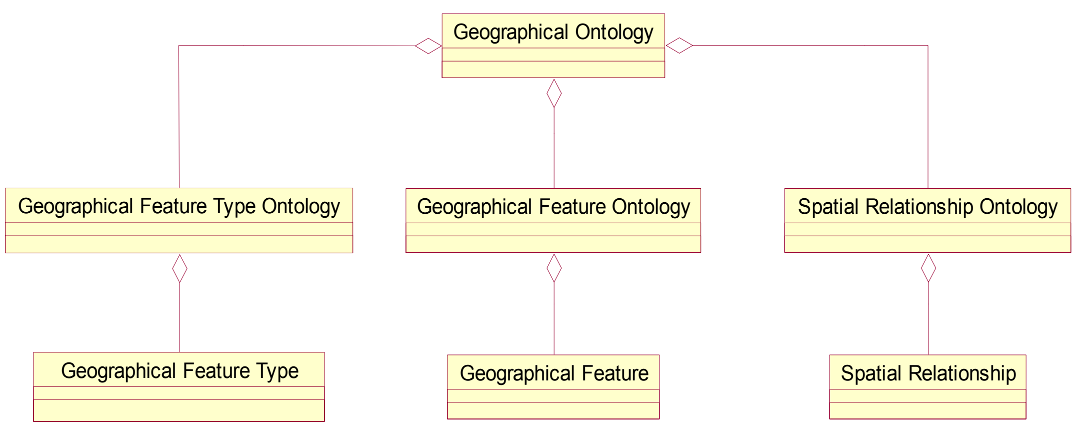


Figure 5. Geographical Ontology in SPIRIT

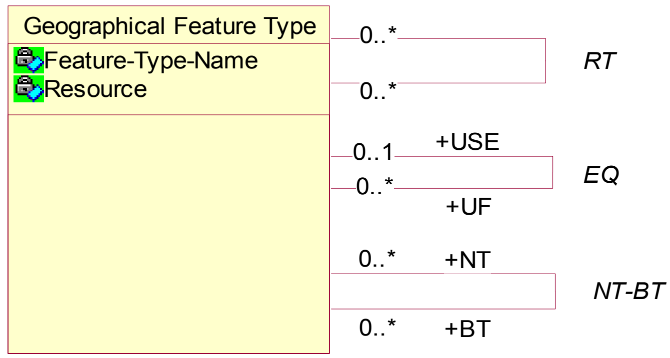


Figure 6. Geographical Feature Type

The second component of the geographic ontology is the geographical feature ontology which encodes the concrete feature instances in a given geographical space. For performance and storage reasons in SPIRIT, we consider very detailed descriptions of geographic features are not necessary for many types of geographical search. However, unlike existing gazetteers which encode all geographic feature in the same manner, in SPIRIT we expect that our ontology can accommodate the properties that are specific to particular types of geographic features. So a practical approach may be to define a base schema, in which only the properties which are common to all geographic features are maintained. New feature types can then be defined by extending the base schema with the specific properties of their own.

The base schema for the geographic feature ontology is illustrated in Figure 7. That is, for each geographic feature, it encodes

1. one and only one **Feature-ID**, which uniquely identifies a geographical feature

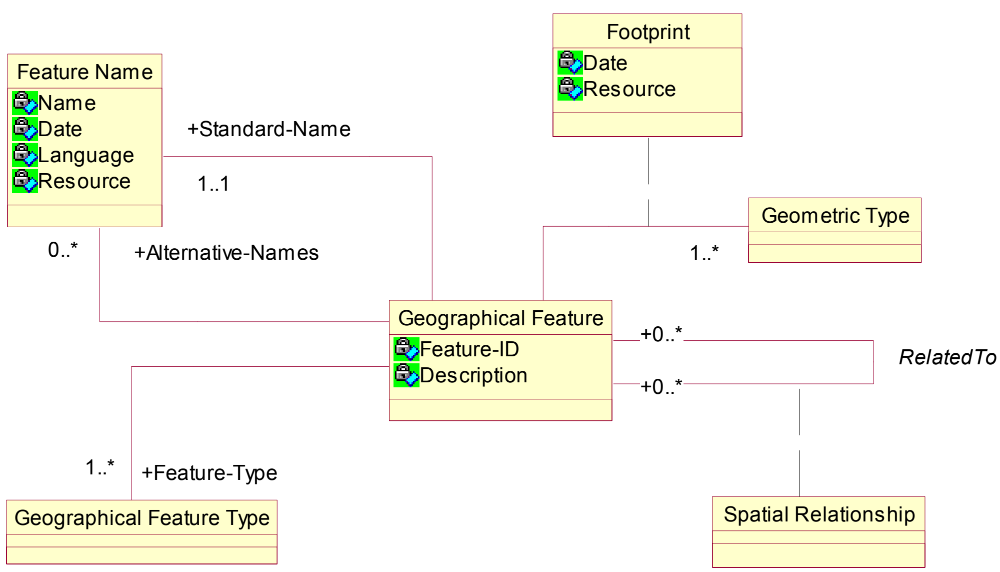


Figure 7. Base Schema of Geographical Feature Ontology

2. one and only one **Standard-Name**, which specifies a name by which a geographical feature is best known. The temporal aspect, the **date** when the name is used, is accommodated in our ontology to provide a context for the referred name. We also associate **Standard-Name** with a **language** in which the name is specified. This is necessary, because although at the moment, we are concentrating on English language specification of the place names, we expect that multiple languages will be supported in SPIRIT in the future. Finally, this section also specifies the **resource** which contributes the information.
3. zero or more **Alternative-Names**, which specify the variant names of a geographical feature. As with the Standard-Name, each variant name is also associated with a date when the name is used, a language in which it is specified and a resource which contributes to the information.
4. one or more **Feature-Types** as defined in geographical feature type ontology**.**
5. one or more spatial **Footprint.** Currently we allow each footprint to be one of following geometric types as shown in Figure 8:
    - a **geometric point** (composed of one and only one coordinate point), which represents the central location of a geographic feature. This geometric type is used when the extent of a geographic feature is not significant.
    - a **polyline** (composed of two or more coordinate points), which represents the centre-line of a geographic object. This geometric type is usually used for a geographic feature when its length is significant, such as a *river*.
    - a **polygon** (composed of two or more coordinate points), which represents the boundary of a geographic feature (two points represent a bounding box). This geometric type is usually used for representing two-dimensional geographic features such as a *forest.*

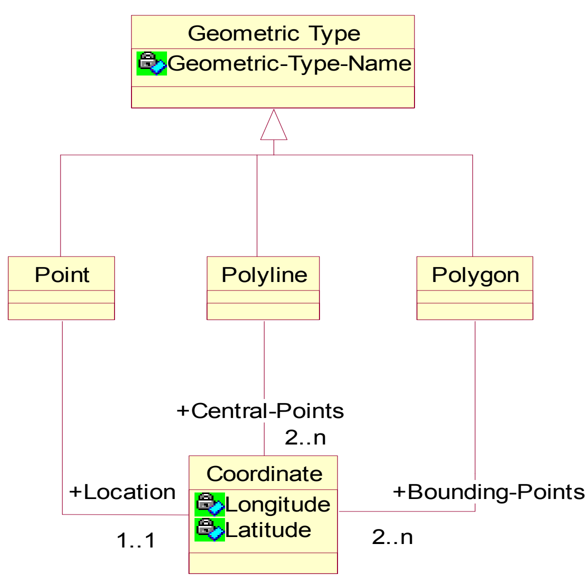


Figure 8. Geometric Feature Types

6. **Description**, a short narrative description of the geographical feature.
7. zero or more **Spatial Relationships**, representing how a geographical feature is related to other geographical features. The possible spatial relationships between various geographic features is defined in the third component of the geographical ontology – spatial relationship ontology as illustrated in Figure 9.

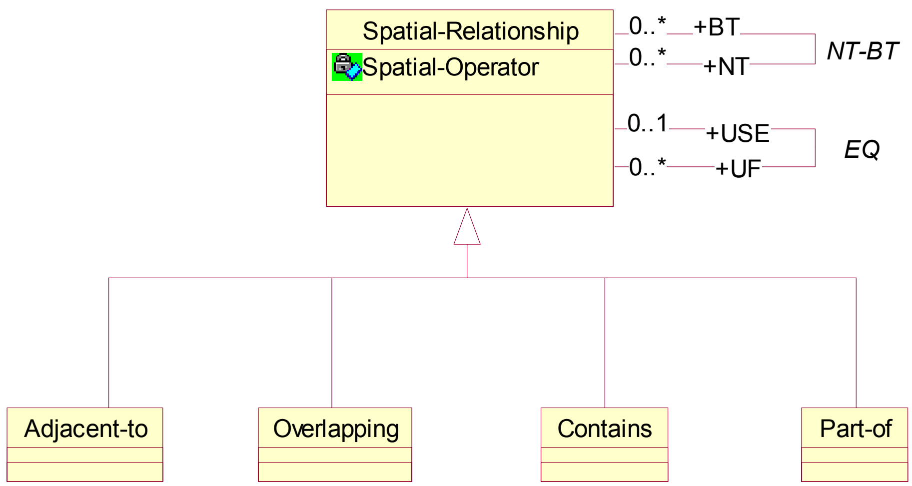


Figure 9. Spatial Relationship

Currently, the type of spatial relationships we choose to support in SPIRIT are as follows:

- **part-of** relationship, encoding which geographical features this geographical feature belongs to.
- **contains** relationship, this is the converse of ***part-of*** and is used to encode which geographical features are inside this geographical feature.
- **adjacent-to** relationship, encoding which geographical features share a boundary with this geographical feature.
- **overlapping** relationship, encoding which geographical features overlap with this geographical feature .

To support spatial reasoning and intelligent interpretation of spatial query terms, we allow that each spatial relationship defined above can be related to other spatial relationships, either defined in Figure 9 or not. The relationship *UF* is used to encode other spatial operators which denote same spatial relationships as defined above, for example, *adjacent-to* may have an *UF* relationship with *bordering, and overlapping* may have an *UF* relationship with *intersect. USE* is inverse to *UF.* In addition to *UF/USE*, we also support *NT/BT* (narrow/border) relationships between spatial relationships, for example, *adjacent-to* may have a *BT* relationship with *near* if we can deduce *near* from *adjacent-to*.

The geographical ontology proposed above represents an effective way for encoding geographic features for SPIRIT. First, it encodes both meta-level (geographical feature types and spatial operators) and data-level (geographical features) information, and two levels are closely related to each others through the feature type that an individual feature instance belongs to and spatial relationships that an individual feature instance may have. Consequently, it supports intelligent interpretation of place name, place type and spatial operators as discussed in Section 4.1.

Second, in proposing the base schema for the geographic feature ontology (the second component of SPIRIT ontology), we effectively declares a set of properties which are necessary for most geographic features. For example, it supports encoding both the standard name and variant names of a place. This can help a search engine retrieve documents which refer to a place either in standard way or in alternative contexts. The base model also supports alternative representations of geometric properties of geographic features respectively as geometric point, polyline and polygon. This makes it possible for us to encode the geographic data in both coarse-grain and fine-grain manner. Fine-grain encoding is important especially when the geometric *preciseness* of geographic ontologies is significant. Coarse-grain encoding is necessary since in SPIRIT it is unrealistic for us to encode detailed geometric data for every feature. Alternative encoding of footprints thus gives us the flexibility to encode some selected places in a fairly detailed manner and others places coarsely for experimental purposes. But please note that in proposing the base schema for the geographic feature ontology, we are trying to provide an open framework for describing geographic features, and we do not impose a restriction that all features are described only using the properties defined for the base schema. We allow new feature types to be constructed by extending the base schema to accommodate some specific properties. For example, in the base schema we restrict the footprint to be one of the following types: geometric point, polyline or polygon. However, we allow other complex geometric types to be encoded in the SPIRIT ontology by extending the base geometric feature types.

Finally, the set of spatial relationships supported in the proposed ontology is based on the consideration of their usefulness and storage expense. For example, using *part-of* relationship, we are able to generate the hierarchy position of a particular place, which may be useful in clarifying spatial query terms supplied by users. *Adjacent-to* will be useful to interpret various spatial operators, such as *near, bordering, neighbouring* etc. To reduce the storage expense for SPIRIT ontology, we restrict both *part-of* and *contains* relationships only encoding geographic features which are directly related to the concerned feature, rather than the whole hierarchies. For example, given the geographic feature *UK*, the *contains* relationship just encodes *England, Scotland, Wales* and *Northern Ireland*, rather than all the counties, cities, towns inside in *UK*. These regions would then refer to their contained counties and so on.

5.2 Ontology Access Operations

Based on the ontology proposed in Section 4.1, in this subsection, we specify a set of operations to be used in the SPIRIT system for accessing, querying and processing of geographic data stored in the SPIRIT ontology. It is important to note that the operations we specify here are identified as the base operations that are most likely to be requested by the SPIRIT system, and other operations might be built on the top of the base operations to perform more complex tasks.

5.2.1 getFeature(L1, L2) Operation

The getFeature operation supports the retrieval of geographic features which are explicitly stored in the SPIRIT ontology. The operation has two optional parameters. The first parameter L1 restricts this operation by exposing constraints on the properties that describe the features. For example, getFeature(<Feature-Type.name='city'>) will retrieve all features of which the feature type is *city*, and getFeature(<Standard-Name.Name='New York' and Feature-Type.name="state">) will retrieve all features of which the standard feature name is *New York* and feature type is *state.*

The second parameter L2 restricts which aspect of a geographic feature is to be retrieved by explicitly listing the desired property names of the features. For example, getFeature(<Feature-Type.name='city'>, <Footprint>) will displays the footprints of features of which the feature type is of *city,* and getFeature(<Standard-Name.Name='Cardiff'>, <Identifier, Feature-Type>) displays the Identifiers and the Feature-Types of the feature of which the standard name is *Cardiff*.

5.2.2 getFeatureType(L1) Operation

The getFeatureType operation retrieves geographic feature types which are supported in the SPIRIT ontology. The optional L1 is used to restrict the classes of the geographic feature types we are interested in. For example, getFeatureType(<Feature-Type.NT='city'>) displays the feature types of which the narrow term is *city, and* getFeatureType(<Feature-Type.USE='city'>) displays the feature types of which *city* is used as the preferred term.

5.2.3 getHierarchy (L1, L2, L3) Operation

The getHierarchy operation supports the retrieval of hierarchies of the specified feature. This is achieved by transitive traversal of the *part-of* or *contains* relationships of the concerned feature to derive the high or low level features which the specified feature belongs to or contains. The mandatory parameter L1 is used to indicate whether the high level or the low level hierarchy is to be computed. The mandatory parameter L2 specifies the interested feature, and this is done by explicitly supplying the Feature-ID of the concerned feature. The optional parameter L3 is used to specify the required hierarchical level. When not supplied, level 1 will be assumed by default. For example, getHierarchy (<high>, <Feature-ID=01079>, <level=3>) gets the hierarchies up to the third level for the feature whose identifier is 01079.

5.2.4 getConnection(L1, L2) Operation

The getConnection operation supports retrieval of the adjacent features of the specified feature. This is achieved by transitive traversal of the *adjacent* relationship of the concerned feature to derive the features that the specified feature is adjacent or indirectly connected to. The mandatory parameter L1 is used to specify the interested feature, and this is done by explicitly supplying the Feature-ID of the concerned feature. The optional parameter L2 is used to specify the required adjacency step. For example, getConnection(<Feature-ID=01079>, <level=3>) gets the features which are adjacent to the specified feature (whose identifier is 01079) through three steps of the *adjacent* relationship.

5.2.5 getOverlap(L1) Operation

The getOverlap operation supports the retrieval of the features that overlap the specified feature. This operation is quite complex and will involve other spatial relationships in addition to the *overlap* relationship. For example, an overlapped place may be part-of another place which is therefore also overlapped. The mandatory parameter L1 is used to specify the interested feature, and this is done by explicitly supplying the Feature-ID of the feature. For example, getOverlap(<Feature-ID=01079>) retrieves the features which overlap the feature whose identifier is 01079.

5.3 Summary of ontology support for SPIRIT functionality

The ontology will play an essential role in the context of the SPIRIT project, but the functionality associated with it will be directly associated with the three areas of the user interface, metadata extraction and relevance ranking. We now summarise briefly the nature of the support that the proposed ontology will provide to these activities.

5.3.1 Support for the user interface

It is envisaged that the user interface will allow user interaction by means of text and map graphics. We may assume that among the options for text input there will be a structured dialogue that allows the user to enter a subject of interest, a place with which it is associated and the nature of any spatial relationship between these two. When the user names a place, the ontology will allow the place to be identified by searching for all instances of the place name in the ontology. If there are multiple occurrences of the name then the user dialogue will be able to establish which one is of interest to the user. If the specified place is not found in the ontology then the user could be prompted to consider modifying their query. In a structured dialogue the spatial relations offered to the user would be taken from a predefined set maintained by the ontology. In the case of free text input of a spatial relationship (if this were an option), then the ontology could be used to associate the user’s spatial relationship with one of the set of supported relationships for which automatic geometric data processing was implemented. The ontology would also be used to attempt to recognise the subject of a query. Within specific application domains for which ontology support was provided, this would allow for disambiguation of subject terminology. If the subject of a query was found to be a place type stored in the ontology, then it would also allow for the possibility of immediate retrieval and perhaps representation on a map of places of the specified type that were stored in the ontology.

Because each place in the geographical ontology is associated with a geometric footprint, it will be possible to employ the ontology in the user interface to create a map of the user's specified place of interest. The ontology-based map would reflect the system's interpretation of the user's area of interest. If the user wished to modify the interpreted extent, particularly if there was some vagueness in the specification of the place of interest (perhaps due to the use of an imprecise spatial relationship), then the user could modify it interactively. As an alternative to initial textual input, a general small-scale map could be generated from the ontology before allowing the user to home in onto an area of interest by interacting with the map. When resources were retrieved, the footprint of those resources would support map display of the associated location or locations. It may be noted that the content of the geographical ontology will provide not just the geometric location of a place on a map, but also one or more place types that could be used to choose appropriate map symbols, as well as the place names that could be used for annotation

The ontology will play a key role in constructing a query for submission to the search engine on the basis of the user interaction. In this respect the ontology will allow for the possibility of generating additional terms for the subject of a query; generating additional place names, such as alternative versions of a name and the names of contained and nearby places; and creating a spatial window, based on query place footprints, for the purpose of issuing a query to the spatial index of the search index.

5.3.2 Support for metadata extraction

When performing processes of metadata extraction it can be expected that heavy demands will be placed on the ontology for purposes of automatic recognition of the presence of place names, spatial relationships, and perhaps domain-specific terminology, within the text of a document or dataset. Having identified a place name in a document the ontology would then be used to retrieve the associated geometric footprint which could be stored in the metadata. Due to the possibility of the presence of multiple places name references in a document, multiple footprints could be retrieved from the ontology to generate a document footprint which could for example consist of the union of the footprints of the referenced places. This document footprint would subsequently be used for spatial indexing of the document in the search engine database, as well as for purposes of relevance ranking. It may also be the case that the geographical ontology could itself be updated with definitions of places that had been found either directly within geodatasets, or as result of an interpretation of the content of geodatasets.

5.3.3 Support for relevance ranking

The relevance ranking activity in the SPIRIT project will focus particularly on techniques for geographical relevance ranking. This will depend upon information encoded in a document's metadata or on information retrieved directly from the geographical ontology. It is likely that geographical relevance ranking will employ geometric distance measures based comparison of the query footprint and the document footprint. It is also likely that it would create measures of semantic distance that were a function of the relative location of a query place and a document place within the semantic net of the ontology. This could take account of various factors such as the number of links between places in a hierarchy or lattice, the number non-common parent places in a hierarchy, and their respective depths in a hierarchy (Jones et al 2001).

6. ONTOLOGY REPRESENTATION

In order for an ontology to be understood, shared and exchanged among the different users and components of the SPIRIT system, it is necessary that the ontology can be explicitly specified. A step in this direction is to select a language to express the ontology. It is important to note that the same ontology may be expressed in various different languages. The degree of formality of these languages may vary considerably. We can roughly describe these languages as follows:

- highly informal, i.e. ontologies can be expressed loosely in natural language;
- semi-informal, these languages express ontologies in a restricted and structured way, e.g. the text version of "enterprise ontologies" [6];
- semi-formal, these languages express ontologies in an artificial formally defined form, e.g. KIF [14];
- rigorously formal, these languages are equipped with meticulously defined terms, formal semantics, theorem and proofs of such properties as soundness and completeness, e.g. FOL [8].

According to the roles of the ontology in SPIRIT, i.e. clarifying the query terms with respect to the terminology in the ontology, expanding the query terms by reasoning with the ontology, semantic-enriching of web documents by annotating them with spatial information from the ontology, ranking retrieved documents by measuring the semantic distance between the query terms and the terms in web documents, we require that a language that is used in SPIRIT for representing an ontology should satisfy the following requirements:

- It should be compatible with existing Web standards, such as XML, RDF, RDFS, in order to facilitate information exchange with other components of SPIRIT;
- It should have adequate expressive power to represent the ontology proposed in SPIRIT;
- It should be formally specified to facilitate machine understanding and support automated reasoning so that required query expansion services can be performed;
- It is desirable that the selected language has sufficient tool support to aid ontology editing, importing, reasoning etc.

In the remaining part of this section, we will consider how existing ontology languages fit these criteria and recommend one which will be used in SPIRIT.

6.1 Literature Review of the Ontology Languages

Various languages have been used in the literature for specifying ontologies. As shown in Figure 10, some languages are based on XML, such as XOL [36], SHOE [21], OML [26], RDF [38] etc, some are based on Description Logics (DLs) [17], e.g. KIF[14], CycL[11], CLASSIC [34], and some are built based on both of XML and DLs, e.g. OIL [7], DAML+OIL [19], OWL [37]. In the remaining part of this subsection, we will discuss the features of these three groups of languages. This study complements a previous study of mark-up languages undertaken in SPIRIT and reported in [40].

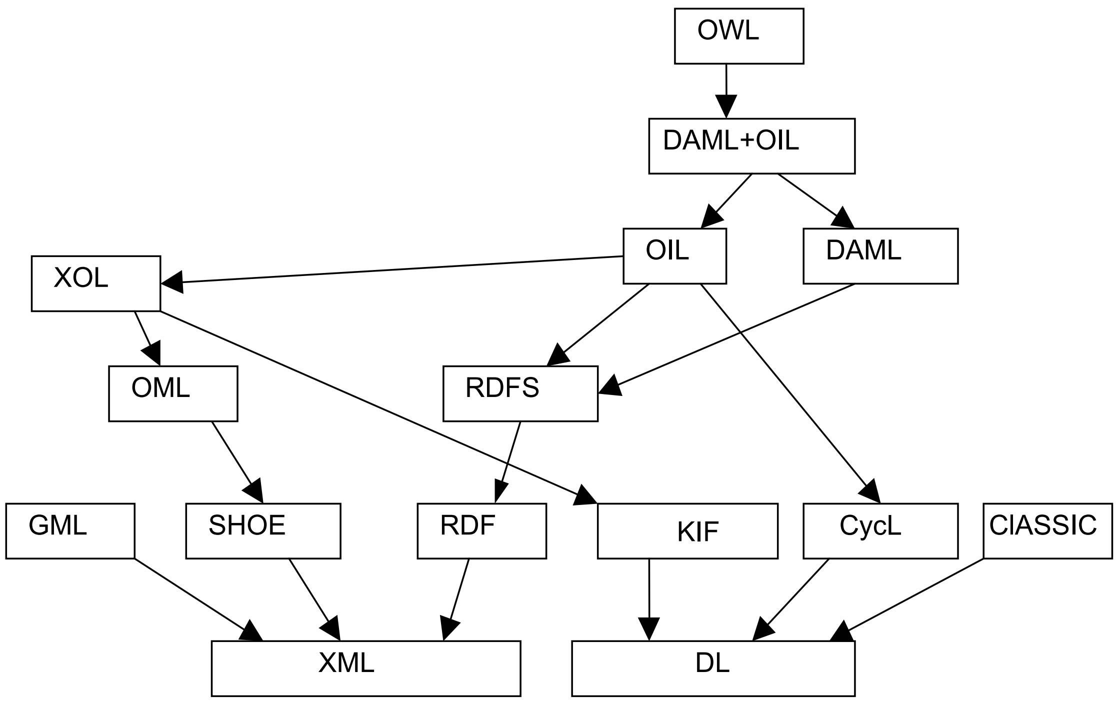


Figure 10. Ontology Language Hierarchy

6.1.1 DL-based Ontology Language

Description logics (DLs) are knowledge representation languages for expressing knowledge about concepts and concept hierarchies. They can be seen as sub-languages of predicate logic. The basic building blocks of DLs are concepts, roles and individuals. Concepts describe the common properties of a collection of individuals and can be considered as unary predicates which are interpreted as sets of objects. Roles are interpreted as binary relations between objects.

Each DL language defines also a number of language constructs (such as intersection, union, role quantification, etc.) that can be used to define new concepts and roles. For instance, to express "every geographic feature can have one and only one identifier, at least one feature name, and at least one footprint which is one of the following types: point, polyline and polygon", we can use the following DL expression:

```
feature ⊑ (AND  (AT-LEAST, 1, identifier)
                (AT-MOST, 1, identifier)
                (AT-LEAST, 1, name)
                (ATLEAST, 1, footprint)
                (ALL footprint  (ONEOF  point, polyline, polygon)))
```

which is based on a well known DL language CLASSIC [34].

The main advantage of DLs is their high expressiveness coupled with decidable and efficient inference procedures for some reasoning tasks. The principle reasoning tasks of DLs are classification, satisfiability, subsumption and instance checking. Subsumption is used to check if there is an is-a relation between two concepts. Classification is the computation of a concept hierarchy based on the subsumption. Instance checking determines if an individual is an instance of a concept. When DLs are used to represent ontologies, the following reasoning tasks can be recognized:

- Design and maintenance of ontologies
    - check class consistency and compute class hierarchy
- Integration of ontologies
    - assert inter-ontology relationships
    - compute integrated class hierarchy/consistency
- Querying class and instance data w.r.t. ontologies
    - determine if set of individuals are consistent w.r.t. ontologies
    - determine if individuals are instances of ontology classes
    - retrieve individual ontology entries satisfying a query expression

Though most DL languages have well-defined syntax and semantics, and are usually equipped with efficient reasoning, when used to represent the ontology proposed in SPIRIT, they suffer from several limitations. First, they are not compatible with existing web languages, which makes it hard for ontologies represented in them to be shared and exchanged. Secondly, some spatial reasoning tasks in SPIRIT can not simply be reduced to the available reasoning tasks in DLs. Finally, the tools developed for DLs often do not integrate well with existing web tools, which makes it difficult to import, export, access the ontologies specified in them.

6.1.2 XML-based Ontology Languages

In addition to the efforts of employing DLs for ontology representation, some research formulates ontologies using the languages that are based on existing web standard XML. Notable examples include RDF [38], RDFS [18], XOL [36], SHOE [21] etc. They restrict XML by providing a set of primitives to express knowledge in a standardized manner to facilitate machine-understanding. For example, RDF provides three primitives for expressing knowledge: *resources* are used to represent objects of interest, *properties* define specific aspects, characteristics, attributes, or relations used to describe a resource; and *statements* assign a value for a property in a specific resource. RDFS extends RDF by providing a set of additional primitives for defining relationships between resources and properties, e.g. *subClassOf* is introduced in RDFS to express is-a relationships between objects.

A most relevant language of this group is GML (Geographical Markup Language) [30], which is proposed by OGC for specifying geographic information, including both spatial and non-spatial properties of geographical features. GML is based on the OGC abstract specification [29]. The basic idea of GML is to provide an open, vendor-neutral framework for the definition of geosptial application schema. It supports the description of geographic data by providing a set of base types and structures and allowing an application to declare the actual feature types and property types of interest by extending basic types in GML.

For example, the following code defines a geographic feature type *Mountain,* which extends base feature type *AbstractFeatureType* provided by GML, and a specific property *elevation* is defined for it.

```
<complexType name="Mountain">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element name="elevation" type="Real"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

With above geo-feature type defined, we can encode information for *Everest* as follows:

```
<Mountain>
    <gml:description>World's highest mountain </gml:description>
    <gml:name>Everest</gml:name>
    <elevation>8850</elevation>
</Mountain>
```

Unlike DLs, the XML-based languages are relatively compatible with existing Web standards since many of them are designed to facilitate machine-understandable web representation. However, the main drawback of these groups of languages is that they lack reasoning support. Thus when used to represent the ontology in SPIRIT, it is difficult for them to fulfil all the goals of the ontology in SPIRIT.

6.1.3 DL+XML-based Ontology Languages

Another steam of ontology languages are built on top of both XML and DLs, and thus they are compatible with existing Web standards and at the same time retain the formal semantics and reasoning services provided by DLs. Examples of such languages include OIL [7], DAML-ONT [12], DAML+OIL [19], OWL [37] etc.

OIL is a web-based representation and inference layer for ontologies, and it provides a set of modelling primitives which are commonly used in DLs. The main focus of OIL is the class level description of an ontology. For example, the definition for the geographic feature type *Mountain* given in Section 5.1.2 may be specified in OIL as:

```
<rdfs:Class rdf:ID=" Mountain">
    <rdfs:subClassOf rdf:resource gml:AbstractFeatureType"/>
    <oil:hasSlotConstraint>
         <oil:ValueType>
              <oil:hasProperty rdf:resource="#elevation"/>
              <oil:hasClass rdf:resource="#Real" />
 </oil:ValueType>
</oil:hasSlotConstraint>
  </rdfs:Class>
```

The main drawback of OIL is that it specialised on class level reasoning, and does not provide reasoning services at the instance level.

DAML-ONT is developed by DARPA project [12] and it inherits many aspects from OIL, and the capabilities of the two languages are relatively similar. DAML+OIL layers on top of RDFS and combines the efforts from OIL and DAML-ONT. It Inherits many ontological primitives from RDFS, e.g. subclass, range, domain, and adds a much richer set of primitives from OIL and DAML-ONT, e.g. transitivity, cardinality, and it allows assertion of axioms. For example, we can specify in DAML+OIL the axiom “A big city is a city which has a population greater than 5 million” as the follows:

```
<daml:Class rdf:ID="BigCity">
          <rdfs:label> Big City</rdfs:label>
          <rdfs:subClassOf  rdfs:resource=”#City”>
          <rdfs:subClassOf
              <daml:Restriction >
                  <daml:onProperty rdf:resource="#population"/>
                  <daml:hasClass  rdf:resource=” #over5m”/>
              </daml:Restriction>
          </rdfs:subClassOf>
</daml:Class>
```

Derived from DAML+OIL, OWL is released by W3C as a semantic markup language for publishing and sharing ontologies on the World Wide Web. OWL was initiated by W3C in November 2001, and is expected to ultimately replace DAML+OIL. It overcomes various problems with DAML+OIL, for example,

- problems with syntax and semantics, mainly related to relationship with RDF
- some constructs little used and/or hard to understand
- some constructs missing
- some aspects of language not user friendly
- badly chosen "key words", e.g., uniquelyIdentifyingProperty

Compared with other ontology languages, the potential of OWL is great since it is recommended by W3C, combines the efforts from many prevalent ontology languages, and gets support from more than 50 academic and industrial members. However, the disadvantage of OWL is that it is too young to have many tools to support it.

In this research, we suggest that DAML+OIL is a suitable language for expressing the SPIRIT ontology proposed in Section 4. The recommendation is based on the following observations. First, DAML+OIL is expressive enough for representing the ontology proposed in section 4. Second, DAML+OIL supports some basic ontology reasoning services which are important for SPIRIT ontology development. Finally, there are available some tools which are implemented for ontology development in DAML+OIL, which, we believe, can greatly reduce the overhead on SPIRIT ontology development. However, we also envisage the possibility of migrating ontology developed in DAML+OIL to OWL in the later stages of the SPIRIT project as more supporting tools for OWL become available.

6.2 SPIRIT Ontology Specification in DAML+OIL

DAML+OIL is written in RDF, i.e., DAML+OIL markup is a specific kind of RDF markup. Thus, a DAML+OIL ontology begins with an RDF start tag including several namespace declarations:

```
<rdf:RDF
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:gdt = "http://www.cs.cf.ac.uk/scmgf/spirit-dt.xsd#"
  xmlns     ="http://www.cs.cf.ac.uk/scmgf/spirit#"
>
```

The first three declarations make statements about the RDF, RDF Schema and XML Schema datatype namespaces. The fourth declaration says that in this document, elements prefixed with daml: should be understood as referring to things drawn from the namespace called http://www.daml.org/2001/03/daml+oil#. The fifth declaration refers to the location of this ontology document itself.

The remaining part of a DAML+OIL ontology consists of zero or more headers, followed by zero or more class elements, property elements, and instances.

The header asserts that this document is an ontology. For example, the following is a header declaration for a DAML+OIL ontology:

```
<daml:Ontology rdf:about="">
    <daml:versionInfo>$Geographic Ontology,v1.0,2002$</daml:versionInfo>
    <rdfs:comment> A spatial ontology for SPIRIT </rdfs:comment>
    <daml:imports rdf:resource="http://www.daml.org/2001/03/daml+oil">
</daml:Ontology>
```

where the *about* attribute will typically be empty, indicating that the subject of this assertion is *this* document. The *daml:versionInfo* and *rdfs:comment* element generally gives information about this ontology for documentation purposes. The *daml:imports* statement references another DAML+OIL ontology containing definitions that apply to the current DAML+OIL resource, and here we just reference

rdf:resource = "http: //www.daml.org/2001/03/daml+oil"

which means that this particular ontology depends only on the standard DAML+OIL definition.

Now we can specify geographic features in DAML+OIL. In order to specify individual geographic features, it is necessary that we declare some feature types first and properties which describe the features. A feature type can be declared as a DAML+OIL class. For example, the following DAML+OIL statements assert that the *geographic feature* in Figure 7 is a DAML+OIL Class:

```
<daml:Class rdf:ID="Geo-Feature">
        <rdfs:label>Geographic Feature</rdfs:label>
</daml:Class>
```

It doesn't say anything else about *Geographic Feature* other than specifying an identifier *Geo-Feature* for it. We will see below how we can describe attributes for *Geographic Feature.*

Each attribute of a feature can be defined as a kind of DAML+OIL property. For example, in the following we have defined the *Feature-ID* as a property of *geographic feature,* which has *XML Integer* as its value type:

```
<daml:DatatypeProperty  rdf:ID="Feature-ID">
  <rdfs:domain rdf:resource="#Geo-Feature"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#Integer"/>
<daml:DatatypeProperty >
```

Note how DAML+OIL specifies that this property applies to *geographic feature* – it says that the domain of *Feature-ID* is *Geo-Feature* which is the identifier of *geographic feature.*

To relate objects of different classes, we can use the daml:ObjectProperty of DAML+OIL. For example, as shown in Figure 7, geographic feature and Geometric Type are connected to each other via the relationship Footprint. We have specified this kind of relation in DAML+OIL by defining a new property Footprint for geographic feature using daml:ObjectProperty and having Footprint draw it objects from Geometric Type. The following DAML+OIL statements show how this is achieved, where Geometric-Type is the identifier for the DAML+OIL class Geometric Type which is presumably defined somewhere else in the same document:

```
<daml:ObjectProperty rdf:ID="Footprint">
        <rdfs:domain rdf:resource="#Geo-Feature"/>
    <rdfs:range rdf:resource="#Geometric-Type"/>
</daml:ObjectProperty>
```

We can further restrict the properties of a class by supplying constraints on the internal definition of the class. For example, we can place the restriction that each geographic feature can only have one *Feature-ID,* and we have done this in DAML+OIL as following:

```
<daml:Class rdf:ID="Geo-Feature">
        <rdfs:label>Geographic Feature</rdfs:label>
        <rdfs:subClassOf>
        <daml:Restriction cardnality="1">
            <daml:onProperty rdf:resource="#Feature-ID "/>
        </daml:Restriction>
        </rdfs:subClassOf>
<\daml:Class>
```

What happens here is that the *daml:Restriction* defines an anonymous class, namely the class of all things that satisfy the restriction. In this case: the class of all things, each of which has one and only one *Feature-ID*. We then demand that the class *Geo-Feature* is a subClassOf this (anonymous) class. In other words: we demand that every feature must satisfy this restriction.

With all the classes and their properties declared, we can specify individual geographic features in DAML+OIL. For example, the following DAML+OIL statements describe the geographic feature *United Kingdom,* and they specify that this feature has the identifier of *00132*, its standard name is *United Kingdom*, *Great Britain* is its alternative name, its feature type is *country*, it is located at Longitude -4.500, Latitude 54.000 and it is part of a feature whose identifier is *00110.*

```
<Geo-Feature rdf:ID="00132">
        <Feature-ID> 00132 </Feature-ID>
        <Standard-Name>
          <Name>United Kingdom</Name>
          <Date> 01/01/1905 </Date>
          <Language> English </Language>
          <Resource> ADL </Resource>
    </Standard-Name>
        <Alternative-Name>
          <Name>Great Britain</Name>
          <Date> 01/01/1870 </Date>
          <Language> English </Language>
          <resource> ADL </Language>
    </alternative-Name>
```

```
    <Feature-Type>
            <Feature-Type-Name> country </Feature-Type-Name>
            <UF>
              <Feature-Type>
                 <Feature-Type-Name> nation </Feature-Type-Name>
              </Feature-Type>
            </UF>
    </Feature-Type>
    <Footprint>
             <Geometric-Type-Name> point </Geometric-Type-Name>
             <Location>
                   <Latitude> 54.000 </Latitude>
                              <Longitude> -4.500 </Longitude>
                       </Location>
    <RelatedTo>
          <Spatial-Operator> part-of </Spatial-Operator>
          <RelatedFeatureID> 00110 </RelatedFeatureID>

    </RelatedTo>
           <Description> Area was inhabited by ancient Celtic-speaking peoples; invaded by Rome 55
          BC; various parts have been united by England since 11th cen.
          </Description>

<\Geo-Feature>
```


Appendix C lists all the classes and properties specifications of SPIRIT ontology we proposed in Section 4, and the reader is refer to [19] for a detailed introduction to DAML+OIL.

7. ONTOLOGY IMPLEMENTATION

To implement the ontology proposed in this report, we have designed a high level architecture as illustrated in Figure 11.

The modules in the middle part of Figure 11 are the main components for the SPIRIT ontology, which composes of an ontology repository for storing both spatial and domain ontologies, and a reasoning module for other components (as shown in the left part of Figure 11) of the SPIRIT system to access the geographic features stored in ontology repository. The main purpose of the reasoning module is to implement the set of functions we proposed in Section 5.2, and it also supports exporting the retrieved ontology entries into standard format (in our case, DAML+OIL) for exchanging geographic data with other components of SPIRIT.

The modules on the right hand side of Figure 11, i.e. import, create, update and browse modules, are designed to build the ontology, and it is not necessary for them to interact with other components of SPIRIT system.

The Import module is used to generate ontology entries from existing geographic data resources, e.g. TGN, ADL. In addition to the gazetteers, there are available other types of data resource from which we can import some useful geographic data for the SPIRIT ontology. For example, *address point* and *postcode point* are two products from Ordnance Survey which provide detailed geographic data down to street and building levels [32, 33]. The AAT [15] is a useful resource for importing geographic feature types. Data resources which can be used to derive spatial relationships include all kinds of maps.

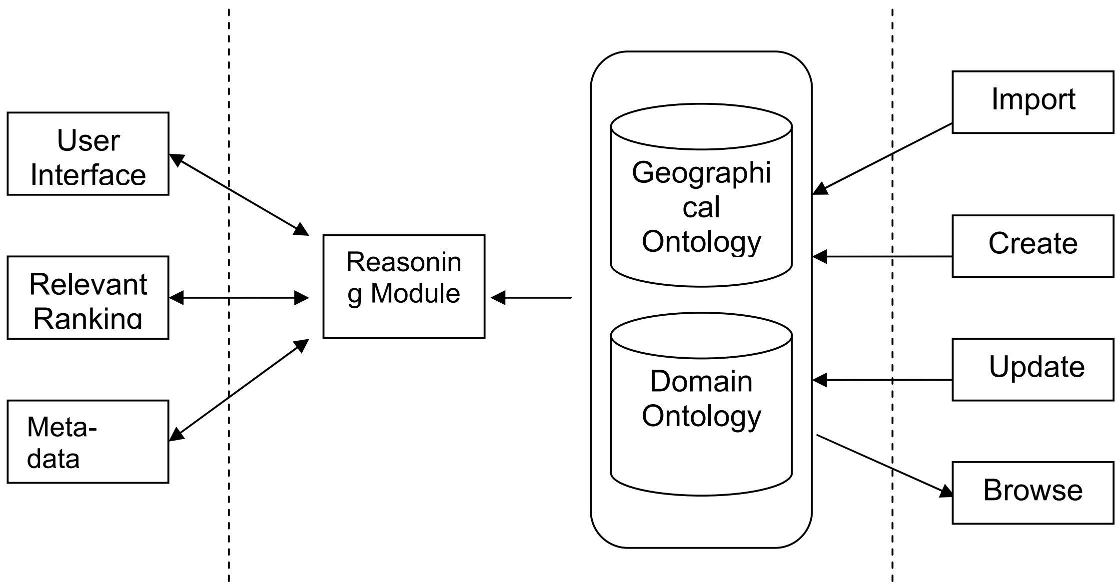


Figure 11. SPIRIT Ontology Implementation Architecture

The Create Module supports generation of ontology entries from scratch. This module is particularly useful for adding individual geographic data into the ontology repository. The Update module is used for modifying existing ontology entries. The Browse module is a graphical user interface that displays geographical information in a user-friendly way.

8. CONCLUSIONS AND FUTURE WORK

In this report, we have discussed the ontology requirements in SPIRIT, and accordingly we have proposed a base model for encoding geographic data to be used in the SPIRIT system for intelligent information retrieval on the Internet. We also reviewed various ontology languages used in the literature for expressing ontologies and proposed that DAML+OIL is a suitable language for specifying the SPIRIT ontology.

There are several issues which will be pursued in the next step of this research, including the following:

- We will investigate how to derive geographic data from various data resources to populate the SPIRIT ontology.
- While we have shown in this report a set of functions for accessing, query and processing of information in ontology, concrete algorithms have to be designed to implement these functions.
- Since SPIRIT is aiming at Global applications, the storage problem might be one of the main factors which affects system performance. The next step of this research will study alternative ontology models and their storage implications in SPIRIT. This will include consideration of the balance between stored and computed spatial relations and the use of distributed ontology architectures. Analysis of the potential for automated deduction of spatial relations will also form part of this work.

APPENDIX A ADL GAZETTEER CONTENT STANDARD

* mandatory (provided parent section exists)
R repeatable

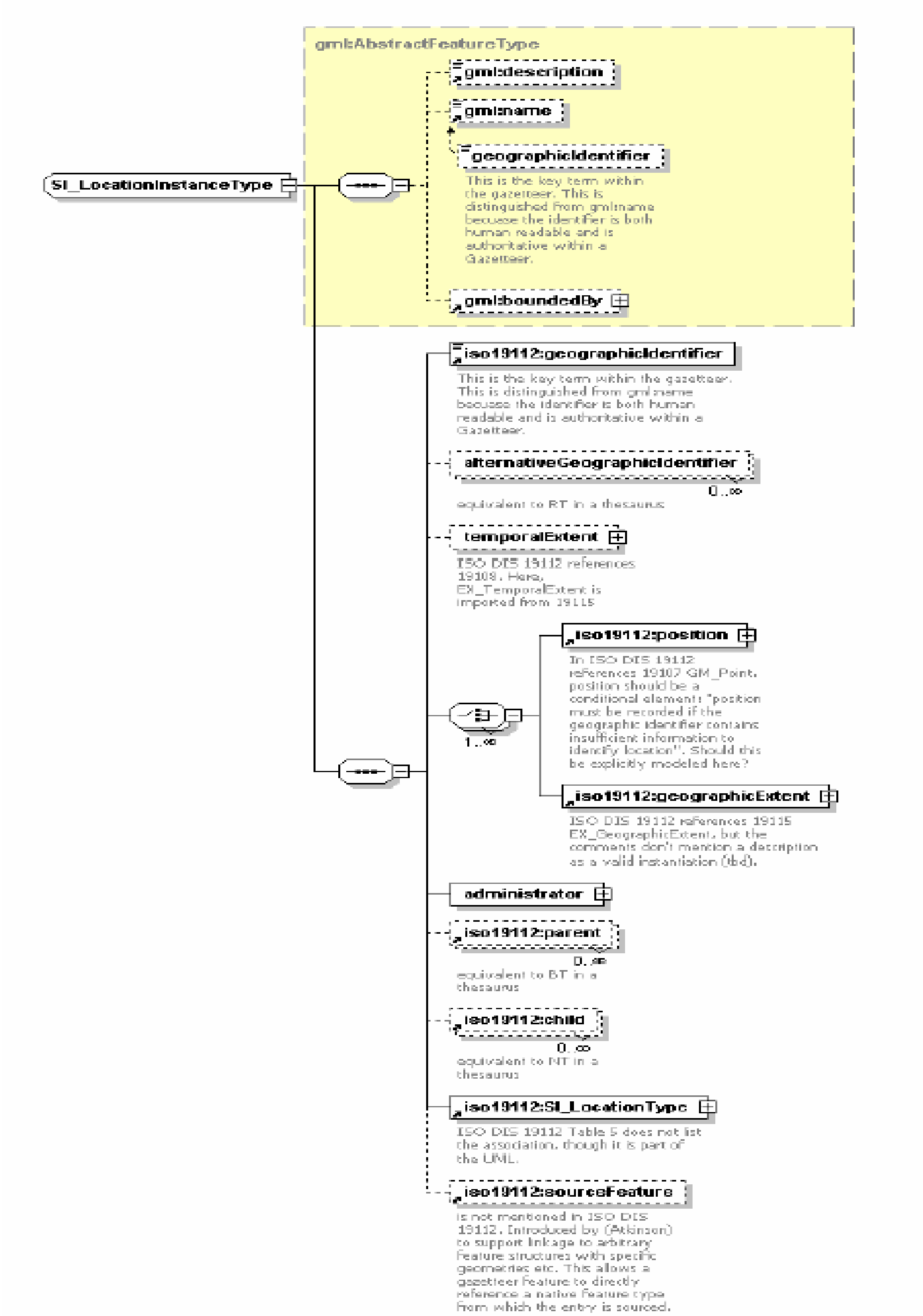
1. Geographic Feature ID *
2. Geographic Name *
    1. Name * [the primary name for feature in a particular gazetteer application]
    2. Name Source
    3. Etymology
    4. Language (default is English)
    5. Pronunciation
    6. Transliteration Scheme Used
    7. Character Set (default is ASCII)
    8. Current / Historical Note * (default is Current) {boolean}
    9. Beginning Date
    10. Ending Date
    11. Time Period Note
    12. Source Mnemonic
    13. Entry Date
3. Variant Geographic Name (R)
    1. Same attributes with Geographic Name
4. Type of Geographic Feature *
    1. Feature Type Schema * [e.g., ADL Feature Type Thesaurus]
    2. Feature Type * (R)
5. Other Classification Terms (R)
    1. Classification Schema
    2. Classification Term (R)
6. Geographic Feature Code (R)
    1. Feature Code Schema *
    2. Feature Code *
    3. Source Mnemonic
    4. Entry Date
7. Spatial Location *
    1. Planetary Body * (default of Earth)
    2. Spatial Representation * (R)
        1. Bounding Box Spatial Geometry Representation *
            1. West Bounding Coordinate *
            2. East Bounding Coordinate *
            3. South Bounding Coordinate *
            4. North Bounding Coordinate *
            5. Current / Historical Note * (default is Current) {boolean}
            6. Beginning Date
            7. Ending Date
            8. Time Period Note

        9. Measurement Date, Beginning Date
        10. Measurement Date, Ending Date
        11. Method of Measurement
        12. Accuracy of Measurement
        13. Source Mnemonic
        14. Entry Date
    2. Detailed Spatial Geometry Representation (R) [set of points; dependent on system capabilities and requirements; can represent set of non-contiguous areas]
        1. Detailed Spatial Geometry Representation * {point, bounding box, linear, complex object}
        2. Number of Points *
        3. Points Order *
        4. (Longitude, Latitude) (R) *
        5. Current / Historical Note * (default is Current) {boolean}
        6. Remaining attributes are same with 7.2.6 to 7.2.14

8. Street Address (Physical Address)
    1. Address *
    2. City *
    3. State or Province *
    4. Postal Code
    5. Country
    6. Source Mnemonic
    7. Entry Date
9. Related Feature (R)
    1. Type of Relationship * {in-state-of, in-province-of, in-county-of, in-country-of, in-region-of, part-of, formerly-known-as}
    2. Geographic Name of Related Feature *
    3. Related ADL Feature ID
    4. Current / Historical Note for Relationship *
    5. Beginning Date of Relationship
    6. Ending Date of Relationship
    7. Time Period Note
    8. Source Mnemonic
    9. Entry Date
10. Description
    1. Short Description *
    2. Source Mnemonic
    3. Entry Date
11. Geographic Feature Data (R)
    1. Type of Geographic Feature Data *
    2. Geographic Feature Data Value*
    3. Geographic Feature Data Value Unit of Measure *
    4. Explanatory Note
    5. Same with 2.9 to 2.13

12. Link to Related Source of Information (R)
    1. Description of Linked Item *
    2. URL *
    3. Source Mnemonic
    4. Entry Date
13. Supplemental Note
    1. Note *
    2. Source Mnemonic
    3. Entry Date
14. Metadata Information *
    1. Entry Note
    2. Entry Date *
    3. Modification Date *
15. Source Information * (R)
    1. Source Mnemonic *
    2. Contributor Organization *
    3. Contributor Web Site
        1. Web Site Title *
        2. URL *
    4. Contact Person
    5. Email
    6. Telephone Number
    7. Contributor Address *
        1. Address *
        2. City *
        3. State or Province *
        4. Postal Code *
        5. Country *
    2. Source Information * (R)
        1. Author Statement
        2. Title *
        3. Edition
        4. Series Name
        5. Series Issue
        6. Publisher *
        7. Publication Date *
        8. Publication Date Note
        9. Pages
        10. Source Identifier
        11. Source_URL

APPENDIX B BASE SCHEME SUPPORTED BY ISO 19112



APPENDIX C SPIRIT ONTOLOGY SPECIFICATION IN DAML+OIL

```
<rdf:RDF
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
  xmlns:daml=”http://www.daml.org/2001/03/daml+oil#”
  xmlns     ="http://www.cs.cf.ac.uk/scmgf/spirit#"
>

<daml:Ontology rdf:about="">
  <daml:versionInfo>$Geographic Ontology,v1.0,03-12-2002$</daml:versionInfo>
  <rdfs:comment> A spatial ontology for SPIRIT </rdfs:comment>
  <daml:imports rdf:resource="http://www.daml.org/2001/03/daml+oil">
</daml:Ontology>

<daml:Class rdf:ID="Geo-Feature">
   <rdfs:label>Geographical Feature</rdfs:label>
   <rdfs:subClassOf>
        <daml:Restriction cardnality=”1”>
             <daml:onProperty rdf:resource="#Feature-ID"/>
        </daml:Restriction>
   </rdfs:subClassOf>
    <rdfs:subClassOf>
        <daml:Restriction cardnality=”1”>
            <daml:onProperty rdf:resource="#Standard-Name"/>
        </daml:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
        <daml:Restriction mincardnality=”1”>
            <daml:onProperty rdf:resource="#Feature-Type"/>
        </daml:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
        <daml:Restriction cardnality=”1”>
            <daml:onProperty rdf:resource="#Footprint"/>
        </daml:Restriction>
   </rdfs:subClassOf>
</daml:Class>

<daml:Class rdf:ID="Feature-Name">
   <rdfs:label>Feature Name</rdfs:label>
   <rdfs:subClassOf>
        <daml:Restriction cardnality=”1”>
            <daml:onProperty rdf:resource="#Name"/>
        </daml:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
        <daml:Restriction cardnality=”1”>
            <daml:onProperty rdf:resource="#Language"/>
        </daml:Restriction>
   </rdfs:subClassOf>
</daml:Class>

<daml:Class rdf:ID="Geo-Feature-Type">
   <rdfs:label>Geographical Feature Type</rdfs:label>
   <rdfs:subClassOf>
        <daml:Restriction cardnality=”1”>
```

```
            <daml:onProperty rdf:resource="#Feature-Type-Name"/>
        </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
        <daml:Restriction maxcardnality=”1”>
            <daml:onProperty rdf:resource="#USE"/>
        </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>

<daml:Class rdf:ID="Geometric-Type">
  <rdfs:label> Geometric Type</rdfs:label>
</daml:Class>


<daml:Class rdf:ID="Point">
    <rdfs:label>Point</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Geometric-Type"/>
         <rdfs:subClassOf>
              <daml:Restriction cardnality=”1”>
                   <daml:onProperty rdf:resource="#Location"/>
              </daml:Restriction>
         </rdfs:subClassOf>
         <rdfs:subClassOf>
              <daml:Restriction>
                   <daml:onProperty rdf:resource="#Geometric-Type-Name"/>
             <daml:hasValue rdf:resource=”#point”/>
              </daml:Restriction>
         </rdfs:subClassOf>
</daml:Class>

<daml:Class rdf:ID="Polyline">
     <rdfs:label>Polyline</rdfs:label>
     <rdfs:subClassOf rdf:resource="#Geometric-Type"/>
     <rdfs:subClassOf>
          <daml:Restriction mincardnality=”2”>
              <daml:onProperty rdf:resource="#Central-points"/>
          </daml:Restriction>
     </rdfs:subClassOf>
          <rdfs:subClassOf>
              <daml:Restriction>
                   <daml:onProperty rdf:resource="#Geometric-Type-Name"/>
             <daml:hasValue rdf:resource=”#polyline”/>
              </daml:Restriction>
         </rdfs:subClassOf>
</daml:Class>

<daml:Class rdf:ID="Polygon">
  <rdfs:label>Polygon</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Geometric-Type"/>
  <rdfs:subClassOf>
        <daml:Restriction mincardnality=”2”>
            <daml:onProperty rdf:resource="#Bounding-points"/>
        </daml:Restriction>
  </rdfs:subClassOf>
            <rdfs:subClassOf>
              <daml:Restriction>
                   <daml:onProperty rdf:resource="#Geometric-Type-Name"/>
             <daml:hasValue rdf:resource=”#polygon”/>
              </daml:Restriction>
  </rdfs:subClassOf>
```

```
</daml:Class>

<daml:Class rdf:ID="Coordinate">
  <rdfs:label> Coordinate</rdfs:label>
  <rdfs:subClassOf>
       <daml:Restriction cardnality=”1”>
           <daml:onProperty rdf:resource="#longitude"/>
       </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
       <daml:Restriction cardnality=”1”>
           <daml:onProperty rdf:resource="#latitude-Y"/>
       </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>

<daml:Class rdf:ID="SpatialRelationship">
   <rdfs:subClassOf>
       <daml:Restriction cardnality=”1”>
           <daml:onProperty rdf:resource="#Spatial-Operator"/>
           <daml:hasClass rdf:resource=”# SpatialQuantifier”/>
       </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class >

<daml:Class rdf:ID="SpatialQuantifier">
  <daml:oneOf rdf:parseType="daml:Collection">
       <daml:Thing rdf:about="#part-of"/>
       <daml:Thing rdf:about="#contains"/>
           <daml:Thing rdf:about="#adjacent-to"/>
       <daml:Thing rdf:about="#overlapping"/>
  </daml:oneOf>
< daml:Class >

<daml:Class rdf:ID="Geometric-TypeName">
  <daml:oneOf rdf:parseType="daml:Collection">
       <daml:Thing rdf:about="#point"/>
       <daml:Thing rdf:about="#polyline"/>
            <daml:Thing rdf:about="#polygon"/>
  </daml:oneOf>
< daml:Class >

<daml:Class rdf:ID="Part-Of">
   <rdfs:subClassOf rdf:resource="#Spatial-Relationships"/>
   <rdfs:subClassOf>
       <daml:Restriction>
           <daml:onProperty rdf:resource="#spatial-Operator"/>
           <daml:hasValue rdf:resource=”#part-of”/>
       </daml:Restriction>
   </rdfs:subClassOf>
</daml:Class >

<daml:Class rdf:ID="Adjacent">
   <rdfs:subClassOf rdf:resource="#Spatial-Relationships"/>
   <rdfs:subClassOf>
       <daml:Restriction>
           <daml:onProperty rdf:resource="#spatial-Operator"/>
            <daml:hasValue rdf:resource=”#adjacent”/>
       </daml:Restriction>
   </rdfs:subClassOf>
</daml:Class >
```

```
<daml:Class rdf:ID="Contains">
    <rdfs:subClassOf rdf:resource="#Spatial-Relationships"/>
    <rdfs:subClassOf>
        <daml:Restriction>
            <daml:onProperty rdf:resource="#spatial-Operator"/>
             <daml:hasValue rdf:resource=”#contains”/>
        </daml:Restriction>
    </rdfs:subClassOf>
</daml:Class >

<daml:Class rdf:ID="Overlap">
    <rdfs:subClassOf rdf:resource="#Spatial-Relationships"/>
    <rdfs:subClassOf>
        <daml:Restriction>
            <daml:onProperty rdf:resource="#spatial-Operator"/>
             <daml:hasValue rdf:resource=”#overlap”/>
        </daml:Restriction>
    </rdfs:subClassOf>
</daml:Class >

<daml:DatatypeProperty  rdf:ID="Feature-ID">
   <rdfs:domain rdf:resource="#Geo-Feature"/>
   <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#Integer"/>
</daml:DatatypeProperty >

<daml:DatatypeProperty  rdf:ID="Description">
   <rdfs:domain rdf:resource="#Geo-Feature"/>
   <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#String"/>
</daml:DatatypeProperty >

<daml:ObjectProperty rdf:ID="Standard-Name">
       <rdfs:domain rdf:resource="#Geo-Feature"/>
       <rdfs:range rdf:resource="#Feature-Name"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="Alternative-Name">
       <rdfs:domain rdf:resource="#Geo-Feature"/>
       <rdfs:range rdf:resource="#Feature-Name"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="Feature-Type">
       <rdfs:domain rdf:resource="#Geo-Feature"/>
       <rdfs:range rdf:resource="#Geo-Feature-Type"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="RelatedTo">
       <rdfs:domain rdf:resource="#Geo-Feature"/>
       <rdfs:range rdf:resource="#Spatial-Relationship"/>
</daml:ObjectProperty>

<daml:DatatypeProperty  rdf:ID="Name">
   <rdfs:domain rdf:resource="#Feature-Name"/>
   <rdfs:range rdf:resource=" http://www.w3.org/2000/10/XMLSchema#String"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="Date">
   <rdfs:domain rdf:resource="#Feature-Name"/>
   <rdfs:range rdf:resource=" http://www.w3.org/2000/10/XMLSchema#Date"/>
</daml:DatatypeProperty>
```

```
<daml:DatatypeProperty rdf:ID="Language">
  <rdfs:domain rdf:resource="#Feature-Name"/>
  <rdfs:range rdf:resource=" http://www.w3.org/2000/10/XMLSchema#String"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="Resource">
  <rdfs:domain rdf:resource="#Feature-Name"/>
  <rdfs:range rdf:resource=" http://www.w3.org/2000/10/XMLSchema#String"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty  rdf:ID="Feature-Type-Name">
  <rdfs:domain rdf:resource="#Feature-Type"/>
  <rdfs:range rdf:resource=" http://www.w3.org/2000/10/XMLSchema#String"/>
</daml:DatatypeProperty>

<daml:ObjectProperty rdf:ID="RT">
         <rdfs:domain rdf:resource="#Geo-Feature-Type"/>
        <rdfs:range rdf:resource="#Geo-Feature-Type"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="NT">
        <rdfs:domain rdf:resource="#Geo-Feature-Type"/>
        <rdfs:range rdf:resource="#Geo-Feature-Type"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="BT">
        <rdfs:domain rdf:resource="#Geo-Feature-Type"/>
        <rdfs:range rdf:resource="#Geo-Feature-Type"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="USE">
        <rdfs:domain rdf:resource="#Geo-Feature-Type"/>
        <rdfs:range rdf:resource="#Geo-Feature-Type"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="UF">
        <rdfs:domain rdf:resource="#Geo-Feature-Type"/>
        <rdfs:range rdf:resource="#Geo-Feature-Type"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="location">
        <rdfs:domain rdf:resource="#Point"/>
        <rdfs:range rdf:resource="#Coordinate"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="central-points">
        <rdfs:domain rdf:resource="#Polyline"/>
        <rdfs:range rdf:resource="#Coordinate"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="bounding-points">
        <rdfs:domain rdf:resource="#Polygon"/>
        <rdfs:range rdf:resource="#Coordinate"/>
</daml:Class>

<daml:DatatypeProperty  rdf:ID="Latitude">
  <rdfs:domain rdf:resource="#Coordinate"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#Real"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty  rdf:ID="Longitude">
```

```
  <rdfs:domain rdf:resource="#Coordinate"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#Real"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty  rdf:ID="Geometric-Type-Name">
  <rdfs:domain rdf:resource="#Geometric-Type"/>
  <rdfs:range rdf:resource=" Geometric-TypeName"/>
</daml:DatatypeProperty>

<daml:ObjectProperty rdf:ID="NT">
      <rdfs:domain rdf:resource="#Spatial-Relationship"/>
      <rdfs:range rdf:resource="# Spatial-Relationship"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="BT">
      <rdfs:domain rdf:resource="#Spatial-Relationship"/>
      <rdfs:range rdf:resource="# Spatial-Relationship"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="USE">
      <rdfs:domain rdf:resource="# Spatial-Relationship"/>
      <rdfs:range rdf:resource="# Spatial-Relationship"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="UF">
      <rdfs:domain rdf:resource="#Geo Spatial-Relationship"/>
      <rdfs:range rdf:resource="# Spatial-Relationship"/>
</daml:Class>


</rdf:RDF>
```

REFERENCES

1. B.A. El-Geresy and A.I. Abdelmoty, *Towards a General Theory for Modelling Qualitative Space.* International Journal on Artificial Intelligence Tools, 2002. **11**(3): p. 347-367.
2. ADL, ADL Feature Type Thesaurus, http://www.alexandria.ucsb.edu/gazetteer/FeatureTypes/, 2002.
3. ADL, ADL Gazetteer Content Standard, http://alexandria.sdc.ucsb.edu/~lhill/adlgaz/, 2002.
4. ADL, Alexandria Digital Library Project, http://www.alexandria.ucsb.edu/, 2002.
5. D. Mark and D. Comas et al. *Evaluating and Refining Computational Models of Spatial Relations Through Cross-Linguistic Human-Subjects Testing*. in *COSIT `95*. 1995.
6. M. Uschold and M. King et al, *The Enterprise Ontology.* Knowledge Engineering Review, 1998. **13**(1): p. 31-89.
7. I. Horrocks et al. *OIL in a Nutshell*. in *ECAI ?0 Workshop on Application of Ontologies and PSMs,.* 2000. Berlin.
8. Barwise, A*n Introduction to First-Order Logic, Handbook of Mathematical Logic,* ed. J. Barwise. 1977: Amsterdam.
9. H. Alani, D. Tudhope and C.B. Jones. G*eographical information retrieval with ontologies of place.* in S*patial Information Theory Foundations of Geographic Information Science, COSIT 2001.* Lecture Notes in Computer Science 2205, 323-335.
10. D. J. Peuquet and Z. Ci-Xiang, A*n Algorithm to Determine the Directional Relationship between Arbitrary-Shaped Polygons in the Plane.* Pattern Recognition, 1987. 2**0(**1): p. 65-74.
11. Cycorp, The Syntax of CycL, http://www.cyc.com/cycl.html, 2002.
12. DARPA, The DARPA Agent Markup Language Homepage, http://www.daml.org/, 2002.
13. M. Egenhofer. A *Formal Definition of Binary Topological Relationships.* in I*nternational Conference on Foundations of Data Organization and Algorithms.* 1989, 457-472.
14. M.R. Genesereth and R.E. Fikes, K*nowledge Interchange Format, version 3.0 reference manual.* 1992, Computer Science Department, Stanford University.
15. Getty, The Art and Architecture Thesaurus (AAT), http://www.getty.edu/research/tools/vocabulary/aat/, 2002.
16. Getty, Getty Thesaurus of Geographic Name, http://www.getty.edu/research/tools/vocabulary/tgn/, 2002.
17. R. M. Gregor, T*echnology of Classification-Based Knowledge Representation Systems,* in P*rinciple of Semantic Networks,* J. Sowa, Editor. 1991.
18. D. Brickley and R.V. Guha, Resource Description Framework (RDF) Schema Specification, www.w3.org/TR/PR-rdf-schema, 1999.
19. I. Horrocks and F. van Harmelen, Reference Description of the DAML+OIL Ontology
Markup Language, www.daml.org/2000/12/reference.html, 2001.
20. A. G. Cohn and S. M. Hazarika, Q*ualitative Spatial Representation and Reasoning: An Overview.* Fundamenta Informaticae, 2001. 4**6(**1-2): 1-29

21. S. Luke and J. Heflin, SHOE 1.01 Proposed Specification, www.daml.org/2000/12/reference.html, www.cs.umd.edu/projects/plus/SHOE/spec1., 2000.
22. D. Hernandez, *Qualitaitve Representation of Spatial Knowledge,* in v*olume LNCS804 of Lecture Notes in Artificial Intelligence.* 1994, Springer,. p. 69-76.
23. ISO19112, G*eographic Information - Spatial referencing by geographic identifiers.* 2002.
24. ISO/TC-211, ISO TC/211 Geographic Information/Geomatics, http://www.isotc211.org/, 2002.
25. C.B. Jones, G*eographical Information Systems and Computer Cartography.* 1997: Prentice Hall.
26. R. Kent, Conceptual Knowledge Markup Language, www.ontologos.org/CKML/CKML%200.2.html, 1998.
27. NIMA, GEOnet Names Server, http://164.214.2.59/gns/html/index.html, 2002.
28. OGC, Open GIS Consortium, Inc., http://www.opengis.org/, 2002.
29. OGC, OpenGIS Abstract Specification, http://www.opengis.org/techno/abstract.htm, 2002.
30. OGC, OpenGIS Geography Markup Language (GML) Implementation Specification, http://www.opengis.org/techno/implementation.htm, 2002.
31. OS, Ordnance Survey 1:50,000 Scale Gazetteer, http://www.ordsvy.gov.uk/productpages/50kgazetteer/, 2002.
32. OS, Ordnance Survey Address-Point, http://www.ordsvy.gov.uk/productpages/addresspoint/, 2002.
33. OS, Ordnance Survey Code-Point, http://www.ordsvy.gov.uk/productpages/codepoint/, 2002.
34. A. Borgida and R. J. Brachman and D. L. McGuinness and L. A. Resnick. C*lassic: A Structural Data Model for Objects.* in S*IGMOD Conference.* 1989. 58-67
35. S. Balley and B. Bucher and D. Petrelli and A. Ruas and M. van Kreveld and M. Sanderson and M. Sester, *User Requirements Specification, SPIRIT deliverable report D3-7101.* 2003.
36. R. Karp and V. Chaudhri and J. Thomere, XOL: An XML-Based Ontology Exchange Language, www.ai.sri.com/~pkarp/xol, 1999.
37. W3C, OWL Web Ontology Language, http://xml.coverpages.org/owl.html, 2002.
38. O. Lassila and R. Webick, Resource Description Framework (RDF) Model and Syntax Specification, www.w3.org/TR/PR-rdf-syntax, 1999.
39. K Zimmerman. Q*ualitative Reasoning - Combining Orientation and Distance.* in C*OSIT'93.* Lecture Notes in Computer Science 716, 1993. Berlin, Springer Verlag: 69-76
40. B. Bucher . *Metadata Models and Mark-up Languages , SPIRIT deliverable report D2-6101.* 2002