INTERNATIONAL STANDARD

ISO 19132

First edition
2007-10-15

Geographic information — Location-based services — Reference model

Information géographique — Services basés sur la localisation — Modèle de référence

Reference number
ISO 19132:2007(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Figures

Tables

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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

ISO 19132 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*.

Introduction

This International Standard establishes a framework supporting the development of location-based services (LBS). LBS are software services whose request and response pattern or values depend upon the location of some number of things, either real or conceptual. For example, tracking and navigation as defined in ISO 19133 are both location-based. Emergency response services are location-based since the requested assistance is invariably for a location fairly near the requestor at the time of the request. Environmental monitoring and remediation is dependent on the location and motion or other continuous change of the polluting agents. Even yellow-page directory services are dependent on the location, or tentative future location, of the requestor in search of a convenient business location for the acquisition of specific goods or services, either near his current location or his planned route.

A reference model is a conceptual framework consisting of a set of system decisions, both architectural and policy, which construct the logical environment for a set of applications and processes within a specific domain. A framework contains or references a taxonomy of terms and an ontology that defines the target domain. A framework can contain or reference other frameworks for related application sets or design paradigms. An LBS framework may relate to a framework of geographic information services, since much of its activity is associated to manipulation of location representations and the use of location as a key to other services. Models for frameworks exist at a variety of levels of abstraction, each of which is a generalization of the more detailed model, and a specialization of the more general ones. At the highest level, the only entities are the frameworks representing their respective reference models. This is illustrated in Figure 1.

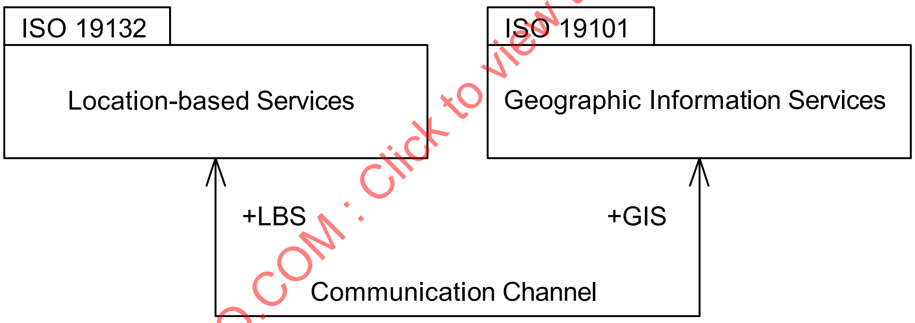


Figure 1 — Relation between LBS and GIS

What this says, in its simplest and most direct terms, is that the two frameworks are coupled and, depending on form more than on functionality, each will invoke services (functions) supplied by the other. This International Standard deals with the communication across the channel depicted in Figure 1. It does so by creating a reference model for the location-based services framework and linking it to the reference model defined in ISO 19101 and ISO/TS 19101-2.

A distinction between an LBS service [1] and a GIS service [2] is that LBS will normally have a larger granularity and significant non-spatial information component, and therefore is able to interact with both geographic data

[1] The term “LBS” includes the word “service”, and so the phrase “LBS service” is logically redundant. When discussing LBSs in relation to other software components, the phrase “LBS service” can be used to maintain symmetry of expression. While logically inconsistent, this is grammatically and poetically acceptable.

[2] It would be useful to redefine GIS as “geographic information service”, but past attempts to override the definition of “geographic information system” with “geographic information science” have not proven very fruitful. In this International Standard, all software components are viewed as services, and so mentions of “GIS” will be taken as “service implementation of GIS functionality”.

frameworks and with general information frameworks containing non-spatial data. Such data may be spatially linked in manners not traditionally used in geographic systems, such as by postal address or telephone number. Another distinction is that LBS services have to deal with the delivery mechanism at a finer level than GIS frameworks. LBS clients are likely to include mobile devices on a multitude of network types, and with a wide variety of capabilities. Thus, an LBS framework supports the same services through a variety of different interface protocols, each tailored for a class of client needs and capabilities. While the details of each client device's interface protocols are beyond the scope of this International Standard, it does address the common semantics of all of the LBS client classes by defining a set of common patterns that provide extensible templates for applications within this domain.

Two of the annexes included in this International Standard are there to highlight the harmonization issue as the LBS domain develops. Organizations that develop standards in LBS need to be aware of other activities. Annex D lists some of the important standards development organizations. Annex E is a crosswalk between common terminology in the geographic information and the intelligent transport system domains. Crosswalks between common terminologies of differing domains are important for semantic interoperability. ITS is used only as an example of one crosswalk.

Geographic information — Location-based services — Reference model

1 Scope

This International Standard defines a reference model and a conceptual framework for location-based services (LBS), and describes the basic principles by which LBS applications may interoperate. This framework references or contains an ontology, a taxonomy, a set of design patterns and a core set of LBS service abstract specifications in UML. This International Standard further specifies the framework's relationship to other frameworks, applications and services for geographic information and to client applications.

This International Standard addresses, for an LBS system, the first three basic viewpoints as defined in the Reference Model for Open Distributed Processing (RM-ODP, see ISO/IEC 10746-1). These viewpoints are the

a) Enterprise Viewpoint – detailing the purpose, scope, and policies of the system,

b) Information Viewpoint – detailing the semantics of information and processing within the system,

c) Computational Viewpoint – detailing the functional decomposition of the system.

The fourth and fifth viewpoints are addressed only in requirements or examples. These are the

d) Engineering Viewpoint – detailing the infrastructure for distribution,

e) Technology Viewpoint – detailing the technology for implementation.

Reference models and frameworks can be defined at a variety of levels, from conceptual design to software documentation. This International Standard

- defines the conceptual framework for, and the type of applications included within, LBS,
- establishes general principles for LBS for both mobile and fixed clients,
- specifies the interface for data access while roaming,
- defines the architectural relationship with other ISO geographic information standards,
- identifies areas in which further standards for LBS are required.

This International Standard does not address the following issues:

- rules by which LBS are developed;
- general principles for roaming agreements for mobile clients and tracking targets.

2 Conformance

Conformance to this International Standard takes on several meanings depending on the type of entity declaring conformance.

- **Semantic conformance** shall imply that the terminology used by the candidate corresponds explicitly to this International Standard where possible.
- **Data conformance** shall imply the usage of data types within application schemas or design specifications that are mappable into types in this International Standard, as in a UML realization of a type by a class.
- **Service conformance** shall imply both the consistent use of message-based request-response interfaces and data conformance for the message packages used by those interfaces.

Conformance may be claimed by a standard, a data structure or schema (such as an encoding definition) or a software module. In all cases, semantics and data conformance are possible. Service conformance is limited to either software or interface specification based on a service-oriented architecture. In service conformance, a data structure may claim this conformance only as part of a larger operational structure (such as the role of XML in SOAP-based SOA applications).

Details for conformance tests are given in Annex A.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 19107, *Geographic information — Spatial schema*

ISO 19109, *Geographic information — Rules for application schema*

ISO 19110, *Geographic information — Methodology for feature cataloguing*

ISO 19112, *Geographic information — Spatial referencing by geographic identifiers*

ISO 19133, *Geographic information — Location-based services — Tracking and navigation*

ISO 19136, *Geographic information — Geography Markup Language (GML)*

4 Terms and definitions

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

4.1
active object
object which is capable of independent actions, and therefore capable of initiating interactions between itself and other objects without immediate prior external stimulation

cf. **passive object** (4.35)

NOTE An active object can represent a **user** or an active **service** that depends on internal (and therefore not visible) triggers to start actions. **Active** and **passive** states can exist for the same object, and such a service can transition between these two states depending on invocation of an activation or deactivation operation protocol.

4.2
basic service
service providing a basic function to other services or applications in a functional manner

cf. **interoperate** (4.18)

NOTE Basic services lack any persistent, user-specific state information between invocations and are not meant for direct access by users. Because they act in a functional manner, they are readily replaceable at runtime by other **services** using the same interfaces.

4.3
candidate route
any route that satisfies all constraints of the routing request, with the possible exception of optimality of the cost function

[ISO 19133]

NOTE Navigation is the process of finding the candidate route that optimizes a chosen cost function.

4.4
cluster
collection of **targets** potentially heterogeneous (each satisfying a different query criteria) whose locations fall within a small neighbourhood

4.5
constraint
restriction on how a **link** or **turn** may be traversed by a **vehicle**, such as **vehicle** classification, or a physical or temporal constraint

[ISO 19133]

4.6
continuous change
change in an attribute whose type has a **distance measure** such that its value can be assumed to take on intermediate values between two known measurements

NOTE The interpolation of continuous change is usually done by taking into consideration constraints on the “curve” joining the two data points (time1, value1) and (time2, value2), looking at the value as a function of time. For example, if the continuous change is for the motion of a vehicle, then the constraints of physics and of the paths appropriate for that vehicle must be taken into consideration.

4.7
cost function
function that associates a measure (cost) to a **route**

[ISO 19133]

NOTE The normal mechanism is to apply a cost to each part of a route, and to define the total route cost as the sum of the cost of the parts. This is necessary for the operation of the most common navigation algorithms. The units of cost functions are not limited to monetary costs and values only, but include such measures as time, distance, and possibly others. The only requirement is that the function be additive and at least non-negative. This last criterion can be softened as long as not zero or less cost is associated to any loop in the network, as this will prevent the existence of a “minimal cost” route.

4.8
coupling
linkage of two or more software systems through information transfer or messaging

NOTE 1 Compare with **integration**. While the conceptual schema of the information transferred shall be agreed upon to some level, **coupling** applications can be and are usually flexible in the data representation of that information as long as the semantics content is correct and mappable to some canonical representation of the conceptual schema. The most common mapping technology used for XML messages is XSLT, and the transformation stylesheet can be supplied either by the service broker or by the service provider. It is considered a best practice for a service provider to supply his functionality through several logically equivalent messaging APIs, each represented by a different URI linked to an XSLT transformation bridge, and implemented by the same internal code.

NOTE 2 Loose coupling and tight coupling are not at present well-defined terms in the literature. Generally, “tight” coupling means that there is some sort of incurred dependency between requester and responder in the use of the interface, while “loose” means no such dependency. The nature of that dependency is not consistently defined between authors. In that light, “tight” coupling or “tight” integration are both bad practices, and have been viewed as such since the inception of the terms. Some literature refers to **integration** as “tight coupling”, but that is a less accurate description.

4.9
digital item
structured digital object [asset, work, service, data or information] with a standard representation, identification and metadata framework

[ISO 21000-1]

4.10
discrete change
change in an attribute value such that it can be assumed to have changed without having taken intermediate values between two known measurements

NOTE Legal changes of parcel changes are discrete, having occurred at a specific time.

4.11
discrete spatiotemporal object
temporal sequence of object representations depicting the same spatial feature at different times

NOTE See Theodoridis, 1999 [31].

4.12
distance measure
distance metric
measure of the pairs of values of an attribute type that assigns a numeric value that is positive, symmetric and satisfies the triangular inequality

NOTE A measure “*d*” is positive if $d(x, y) > 0$ for every *x, y* where $x \neq y$ and $d(x,x) = 0$. A measure “*d*” is symmetric if $d(x, y) = d(y, x)$ for every *x, y*. A measure “*d*” satisfies the triangular inequality if $d(x, y) \leqslant d(x, a) + d(a, y)$ for every *a, x* and *y*. All numeric or vector valued attributes have such a metric, the most common being the Euclidean metric based on the square root of the sum of the squares of the differences in each dimension. Other non-Euclidean metrics take “curvature of space” into account (such as along the surface of the spheroid).

4.13
geocoding
translation of one form of location into another

[ISO 19133]

NOTE Geocoding usually refers to the translation of “address” or “intersection” to “direct position.” Many service providers also include a “reverse geocoding” interface to their geocoder, thus extending the definition of the service as a general translator of location. Because routing services use internal location encodings not usually available to others, a geocoder is an integral part of the internals of such a service.

4.14
identity
data sufficient to identify an object over time, independent of its state

NOTE An **identity** is usually a persistent and constant key member attribute value of the **object**. Since it is temporally constant and unique, it will be the same in any **state** associated to the **object** regardless of its **timestamp**. A moving **object's identity** is independent of both time and **location**.

4.15
instantiate
represent (an abstraction) by the creation of a concrete instance or to create the ability to create an instance

[ISO 19133]

NOTE A class or data element definition instantiates a type if it creates the ability to create objects or data elements respectively that can represent the concepts (instance data and/or operations) defined by that type. A class is instantiated by an object if the class defines that object's structure and function. A data schema is instantiated by a data element, if the data schema defines that element's structure.

4.16
integration
linkage of two or more software systems by the use of a common data and method base

cf. **coupling** (4.8)

NOTE **Integration** and **coupling** are the two major mechanisms for the interoperation of systems.

4.17
interoperability
capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units

[ISO/IEC 2382-1]

4.18
interoperate
communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units

cf. **interoperability** (4.17)

4.19
junction
single topological node in a **network** with its associated collection of **turns** and incoming and outgoing **links**

[ISO 19133]

NOTE Junction is an alias for node.

4.20
license
permission or proof of permission granted to a system participant by a competent authority to exercise a **right** which would otherwise be disallowed or unlawful

4.21
linear referencing system
linear positioning system (ISO 19116)
positioning system that measures distance from a reference point along a route (feature)

[ISO 19133]

NOTE The system includes the complete set of procedures for determining and retaining a record of specific points along a linear feature, such as the location reference method(s) together with the procedures for storing, maintaining, and retrieving location information about points and segments on the highways. See *NCHRP Synthesis 21*, 1974 [25].

4.22
link
directed topological connection between two nodes (**junctions**), consisting of an edge and a direction

[ISO 19133]

NOTE Link is an alias for directed edge.

4.23
link position
position within a network on a **link** defined by some strictly monotonic measure associated with that **link**

[ISO 19133]

NOTE Link positions are often associated to a target feature that is not part of the network. The most common link measures used for this are the distance from start node or address. The most common use of a link position is to geolocate an "address."

4.24
location
identifiable geographic place

[ISO 19112]

NOTE A location is represented by one of a set of data types that describe a **position**, along with metadata about that data, including coordinates (from a coordinate reference system), a measure (from a **linear referencing system**), or an address (from an address system).

4.25
location-based service
LBS
service whose return or other property is dependent on the **location** of the client requesting the service or of some other thing, object or person

[ISO 19133]

NOTE Queries like "find the nearest restaurant" depend on the location of the questioner and are thus appropriate for an LBS.

4.26
location-dependent service
LDS
service whose availability is dependent upon the **location** of the client

[ISO 19133]

NOTE It is often the case that the supplier of information or services may wish to restrict their usage to particular places. For example, Emergency Services request are often routed through the local supplier to ensure jurisdiction or prompt timing. In another common case, cell phones "roam" among providers of local services dependent on which "cell" they are currently in. Such roaming capability can be applied to any essentially "local" service.

4.27
loosely coupled interface
message-based service interface based on a common taxonomic definition and independent of the particulars of message format or representation and of the internal implementation of the service

cf. **coupling** (4.8)

4.28
main-road rule
set of criteria used at a turn in lieu of a **route instruction**; default instruction used at a **node**

[ISO 19133]

NOTE This rule represents what is “most natural” to do at a node (intersection), given the entry link used. The most common version is “as straight as possible”, or to exit a turn on the most obvious extension of the entry street, which is usually, but not always, the same named street that was the entry. Every node in a route is either associated to an instruction or can be navigated by the main-road rule.

4.29
maneuver US
manœuvre GB
collection of related **links** and turns used in a **route** in combination

[ISO 19133]

NOTE Maneuvers are used to cluster turns into convenient and legal combinations. They can be as simple as a single turn, a combination of quick turns (“jogs” in the American mid-west, consisting of a turn followed immediately by a turn in the opposite direction) or very complex combinations consisting of entry, exit, and connecting roadways (“magic roundabouts” in the UK).

4.30
motion
change in the **position** of an object over time, represented by change of coordinate values with respect to a particular reference frame

[ISO 19116]

4.31
navigation
combination of **routing**, **route traversal** and **tracking**

[ISO 19133]

NOTE This is essentially the common term “navigation”, but the definition decomposes the process in terms used in the packages defined in this International Standard.

4.32
neighbourhood
geometric set containing a specified direct position in its interior, and containing all direct positions within a specified distance of the specified direct position

[ISO 19107]

4.33
network
abstract structure consisting of a set of 0-dimensional objects called **junctions**, and a set of 1-dimensional objects called **links** that connect the **junctions**, each **link** being associated with a start (origin, source) **junction** and end (destination, sink) **junction**

[ISO 19133]

NOTE The **network** is essentially the universe of discourse for the **navigation** problem. **Networks** are a variety of 1-dimensional topological complex. In this light, junction and topological node are synonyms, as are **link** and directed edge. The two sets of terms come from graph theory and topology, in the field of mathematics. The two fields are logically related but have historically separate roots. Graph theory deals mainly with the algebra of abstract graphs, while topology has it roots in the geometry or network representations. It is beyond the scope and capability of any International Standard to change 150 years of mathematical literature.

4.34
passive tracking
tracking dependent on stationary sensors external to the **vehicle** or **traveller** allowing for measurements of location when the vehicle's or traveller's **tracking device** passes through the range of external sensors of known position

4.35
passive object
object which can only react to external stimulation and cannot initiate actions on its own

NOTE A **passive object** is usually accessed through an external interface, through which it receives requests, processes those requests and returns data as a response to that request. Since objects can implement more than one type, it is possible for a single object to pass through active and passive states. For example, a tracking service can lie dormant until a tracking request activates a period where the internals of the object initiate tracking activities based on internal triggers as specified within the request. When the tracking request is deactivated, the object may return to a passive state.

4.36
position
data type that describes a point or geometry potentially occupied by an object or person

[ISO 19133]

NOTE A direct position is a semantic subtype of **position**. Direct positions as described can only define a point and therefore not all positions can be represented by a direct position. That is consistent with the “is a type of” relation. An ISO 19107 geometry is also a position, just not a direct position.

4.37
resource
〈ICT〉 digital item controlled by a system participant

4.38
right
action, activity or class of actions that a system participant may perform on or using an associated **resource**

4.39
rights management
control, management, allocation and tracking of the **rights** granted to system participants

4.40
route
sequence of **links**, and/or partial **links**, that describe a path, usually between two **positions** within a **network**

[ISO 19133]

4.41
route instruction
information needed at a point along a **route** in a **network** that allows that **route** to be traversed

[ISO 19133]

NOTE To minimize the number of **instructions** needed to complete a **route traversal**, a default instruction can be assumed at **junctions** without specifically associated **instructions**. This default is called the **main-road rule**.

4.42
route traversal
process of following a **route**

[ISO 19133]

4.43
routing
finding of optimal (minimal **cost function**) **routes** between **locations** in a **network**

[ISO 19133]

4.44
segment
point or polygon from a set

4.45
service
distinct part of the functionality that is provided by an entity through interfaces

[ISO/IEC TR 14252]

NOTE An equivalent definition is “**coupled** software process based on a functional call, invoked by a request and returning a response”. **Services** are often incorrectly defined by the mechanism used for invocation. Under this definition any functional programming mechanism is viable as a server architecture.

4.46
service broker
application that combines or offers lower-level services for specific **user** needs

4.47
service-oriented architecture
SOA
software architecture consisting of coupled services

NOTE The most common SOAs in use today are Web services (using SOAP, UDDI, and WSDL), CORBA and DCOM.

4.48
slope
rate of change of elevation with respect to curve length

[ISO 19133]

4.49
spacestamp
value of a spatial attribute of an **object** at a given time, at which time the **object's state** is measured and recorded

NOTE See **timestamp** (4.53).

4.50
state
⟨of an object⟩ persistent data object reflecting the internal values of all the member attributes or measurable descriptions of a object at a given time

NOTE **State** is usually associated to an **object** by its **identity** and to a time by a **timestamp**.

4.51
target
object or person subject to being located

NOTE There is little logical difference between **traveller** (4.56) and **target** except that the former is normally used for a moving object which is being tracked, and the latter is used for either an object that is not moving, or an object for which a location is needed only once. A **traveller** is the subject of a **tracking** service; a **target** is the subject of a **locating** service. Since this International Standard does not make a distinction between the protocols for these logically similar services, but does need to differentiate between the two concepts, both terms will be used as appropriate to the underlying semantics of the situation. Since all of these terms refer to entities represented by **objects** within the system, they can be combined with adjectives defined for **objects**. So, an **active target** (a target represented by an **active object**) can be used to represent a moving object, since the act of motion is modifying the **target's** internal state and is therefore initiating actions.

4.52
temporal sequence
ordered sequence of **timestamps** associated to a sequence of representations of the same object

NOTE Temporal sequences are not assumed to be evenly spaced in time, nor equidistant in space. For discrete change, the default logic is to sample at temporal points of change if possible, so that the timestamp is the first temporal instance where the attributes listed have taken on that combination of values. For the sake of space savings, some samples in a sequence only list those values that have changed since the immediately preceding temporal sample. For this reason, a sample should only be considered in the context of its containing sequence. For rigid motions (such as vehicle tracking), only centroid (a point value) and orientation (direction of travel) are needed for a temporal sequence describing location and spatial extent. A motion in combination with an object deformation would require more information.

4.53
timestamp
value of time at which an object's state is measured and recorded

4.54
tracking
monitoring and reporting the **location** of a **vehicle**

[ISO 19133]

NOTE A tracking system is usually thought of as following a vehicle, giving a sequence of locations, but the same semantics are involved in finding and giving the location of something once. Therefore, this International Standard does not make a distinction in the protocol for tracking a vehicle and locating a target.

4.55
tracking device
device (tag) carried by a **vehicle** to allow it to determine its location or to be sensed by external objects of known location

NOTE 1 The most common tracking devices are cell phones, GNSS chips, RFID (Radio Frequency ID) tags, or printed tags which are scannable by optical sensors such as “bar codes”.

NOTE 2 The common usage of “vehicle” means a “form of conveyance” or, more simply “thing that conveys (carries) something else”. Thus, a tracked object that carries a **tracking device** to allow it to be tracked is, by definition, a conveyance or **vehicle** for that device. Thus, a cell phone that carries a GNSS device is the vehicle for that device, and the **traveller** carrying the cell phone, allowing him to be tracked, is the **vehicle** for the phone and all of its internal electronics.

4.56
traveller
person subject to being navigated or tracked

cf. **vehicle** (4.59)

[ISO 19133]

4.57
turn
part of a **route** or **network** consisting of a **junction** location and an entry and exit **link** for that **junction**

[ISO 19133]

4.58
user
active object that initiates service requests to the system

NOTE **Users** are usually objects that act as proxies for people accessing the functionality of the system.

4.59
vehicle
object subject to being navigated or tracked

cf. **traveller** (4.56)

[ISO 19133]

NOTE Includes pedestrians. See ISO 14825. In this International Standard, whenever either term is used, the other can be substituted without any change of intent.

4.60
vehicle classification
type of **vehicle**, based on the nature of its construction or intended purpose

[ISO 19133]

NOTE Classifications based on construction may include automobile, truck, bus, bicycle, etc. Classifications based on purpose can include taxi, emergency vehicle, etc. Vehicle classification can be used to determine the application of navigation constraints.

4.61
version (temporal)
complete representation of an object at a given instance in time

NOTE **Temporal versions** differ from samples in that a complete description is required. In this sense a **version** is a complete sample able to be considered outside the domain of the temporal sequence to which it may belong.

4.62
waypoint
location on the network that plays a role in choosing candidate routes potentially satisfying a routing request

[ISO 19133]

5 Symbols and abbreviated terms

5.1 Acronyms

3GPP	3rd Generation Partnership Program
API	Application Programming Interface
ARIB	Association of Radio Industries and Businesses
CALM	Continuous Air-interface Long and Medium range
CORBA	Common Object Request Broker Architecture
CRS	Coordinate Reference System
CSL	Conceptual Schema Language

CTIA	Cellular Telecommunications Industry Association
CWTS	China Wireless Telecommunication Standard group
DCOM	Distributed Component Object Model
ECCMA	Electronic Commerce Code Management Association
EM	Electromagnetic
ETSI	European Telecommunications Standards Institute
FGDC	Federal Geographic Data Committee
GDF	Geographic Data Files
GI	Geographic Information
GML	Geography Markup Language
GNSS	Global Navigation Satellite Systems (formerly GPS — Global Positioning System)
G-XML	Geographic XML
IAEC	International Address Element Code
ICT	Information Communication Technology
IETF	Internet Engineering Task Force
LAN	Local Area Network
LBS	Location-based Service
LDS	Location Dependent Service
LIF	Location Interoperability Forum
LRM	Linear Referencing Method
LRS	Linear Reference System
MMS	Multimedia Messaging Services
OGC	Open Geospatial Consortium [name changed August 2004 from OpenGIS Consortium (now deprecated)]
OMA	Open Mobile Alliance
PDA	Personal Digital Assistant
POI	Point of Interest
RDF	Resource Description Format
RFID	Radio Frequency Identification
RM-ODP	Reference Model for Open Distributed Processing

SDTS	Spatial Data Transfer Standard
SOAP	Simple Object Access Protocol
SVG	Scalable Vector Graphics
T1	Standards Committee T1 Telecommunications
TIA	Telecommunications Industry Association
TTA	Telecommunications Technology Association
TTC	Telecommunication Technology Committee
UDDI	Universal Description Discovery and Integration
UML	Unified Modeling Language
USNG	US National Grid
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WIN	Wireless Intelligent Network
WSDL	Web Services Description Language
XML	Extensible Markup Language

5.2 UML Notation

The UML notation used in this International Standard is described in ISO/TS 19103, and differs from standard UML (ISO/IEC 19501) only in the existence and interpretation of some special stereotypes, in particular, “CodeList”, and “Union,” and looser restrictions on the use of “DataType”.

The term “context diagram”, used extensively in the naming of figures in this International Standard, means a diagram that illustrates the context of a specified central type. The context of a type includes, at least, the types of attributes, operations and association targets. This is the information most useful to the implementer of this central class. Other information on the type may be included if it illustrates the semantics of the central type.

5.3 Taxonomy of data and services — Mapping to RDF

This International Standard deals mainly with semantic information entities represented by data objects, and informational (persistently stored) or functional (calculated on demand) relationships between these entities. At the abstract modelling level, UML relationships between classifiers and associations between object instances represent informational relationships. Processing services that map between semantically related objects represent functional relationships.

Thus, the taxonomy that makes up this framework is representable as an RDF graph data model, see W3C [35]. The inclusion of members in classifiers, the relationships between classifiers, and the operations each represent an arc between subject and object (a “predicate”) in the RDF graph. A simplified instance of a navigation service invocation is given by an RDF graph such as in Figure 2. The darker-line predicate is functional, while the others are most likely informational.

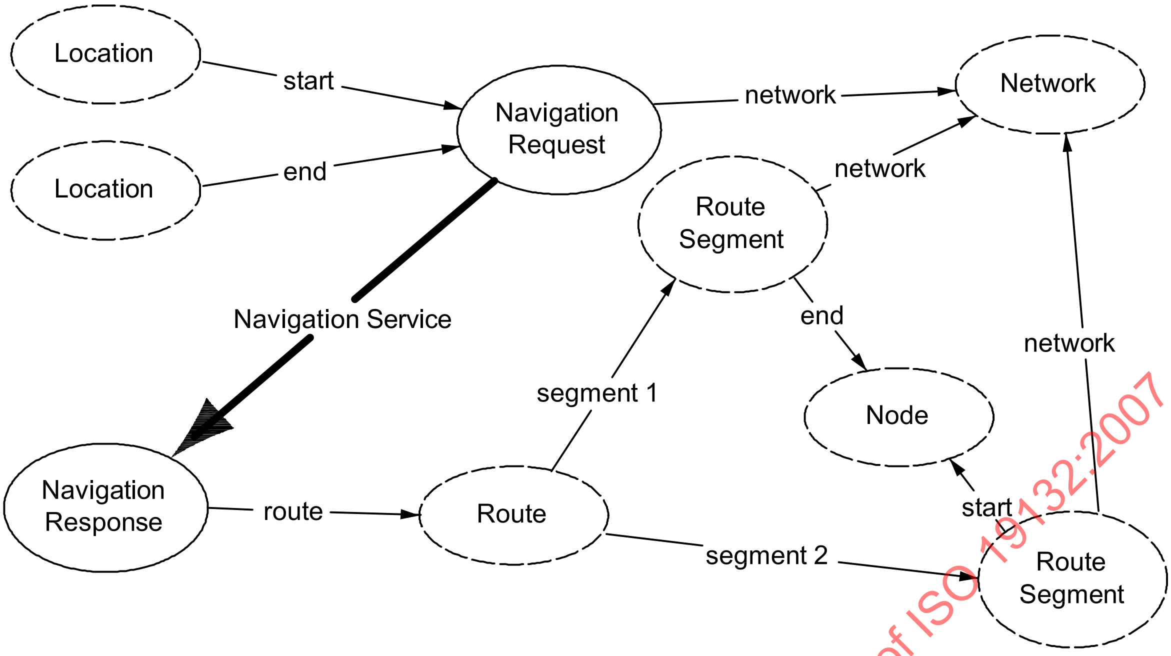


Figure 2 — Simplified navigation service represented as an RDF graph

A service often requires information in a particular format, or otherwise requires preprocessing of input. This results in a decomposition of the RDF graph into more primitive operations as shown in the example in Figure 3.

For the orchestration of simpler services into more complex ones, it would be necessary to match the user request to the input of appropriate services to eventually match the required output of the final services in the chain to the requirement of the user response. The service pattern in Figure 3 would be repeated whenever the location format of the user request did not match the location format of the underlying primitive, basic service.

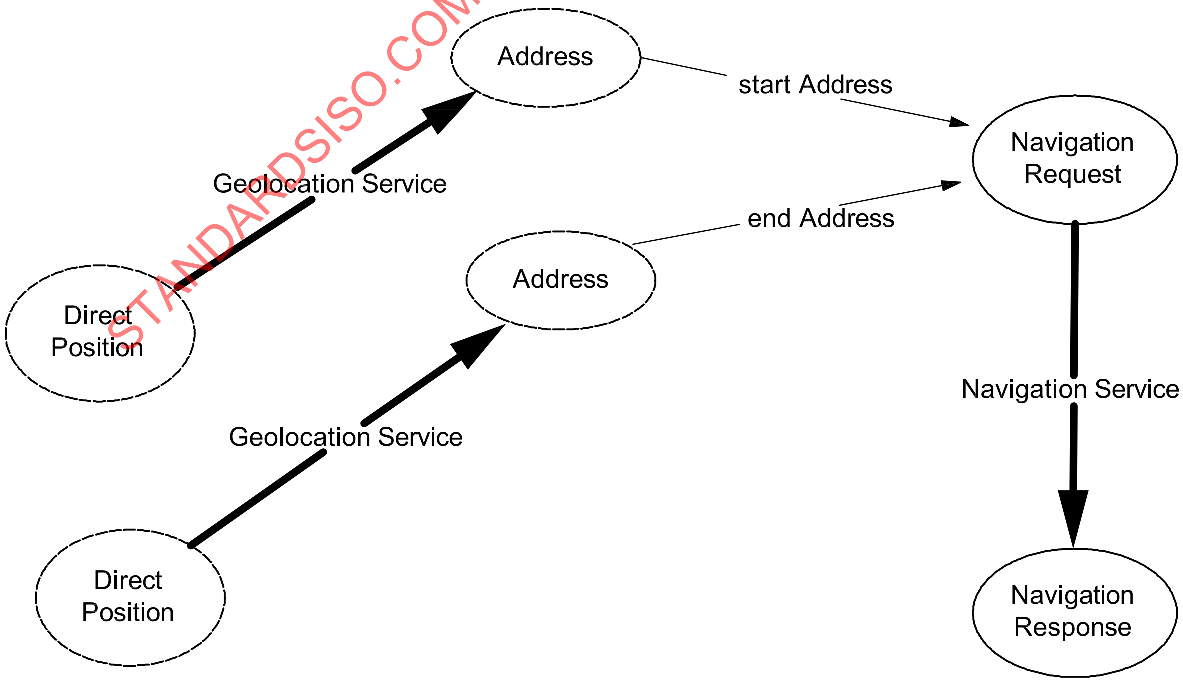


Figure 3 — Example of composition of services

6 ODP Viewpoints used

6.1 Enterprise specification

The enterprise specification provides a description of the requirements and objectives that the environment imposes on the system (ISO/IEC 10746-1). The enterprise concepts of enterprise objects fulfilling roles of **performative** actions are used to describe the multiparty service orchestration inherent in the system concept described above. The roles that the stakeholders in location-based services can play with respect to a service are user, broker or provider. For the application and supporting service broker system, this specification results in the identification of consumer objects (**users**) and of objects managing these users through applications (**service brokers** and **application provider**). For the supporting network infrastructure, a **service provider** object manages the binding object.

6.2 Information specification

In the information specification, the semantics and requirements for the processing of the service information are specified. This is done using the UML schema definitions in the framework clauses of this International Standard (Clauses 8 and 9). Since the roles of the participants of the system vary in their view, distinct schemata may be required in some situations for user–broker and broker–service interactions.

6.3 Computational specification

The computational specification is a description of the system's functionality consistent with the enterprise and information specifications. This is done in the UML operation definitions in the framework clauses of this International Standard (Clauses 8 and 9). The correspondences between objects in the information specification and objects in the computational specification are specified in each case so that consistency between the specifications can be assured.

7 Participation Model

7.1 Model overview – Package: ISO 19132 (this International Standard)

The model supporting this International Standard consists of several packages describing the participants in the LBS community and the services and data employed by them. The package structure for this International Standard is detailed in Figure 4. The contents of the packages are described in the remainder of this clause for LBS Participants and in the following clauses for the packages LBS Services and Message Data Types.

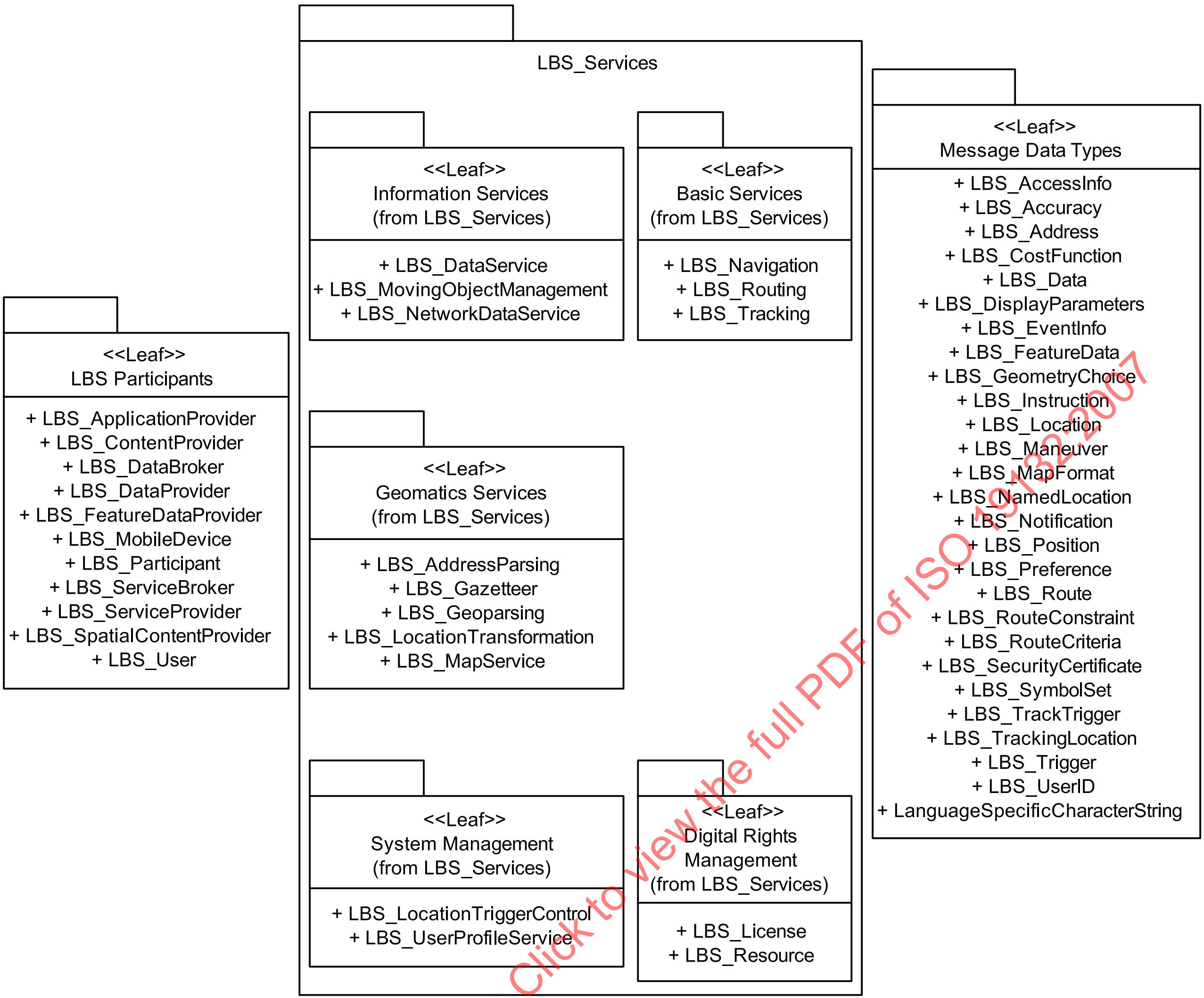


Figure 4 — Overview of UML package structure

Figure 5 shows the dependencies of this package to models defined by other International Standards.

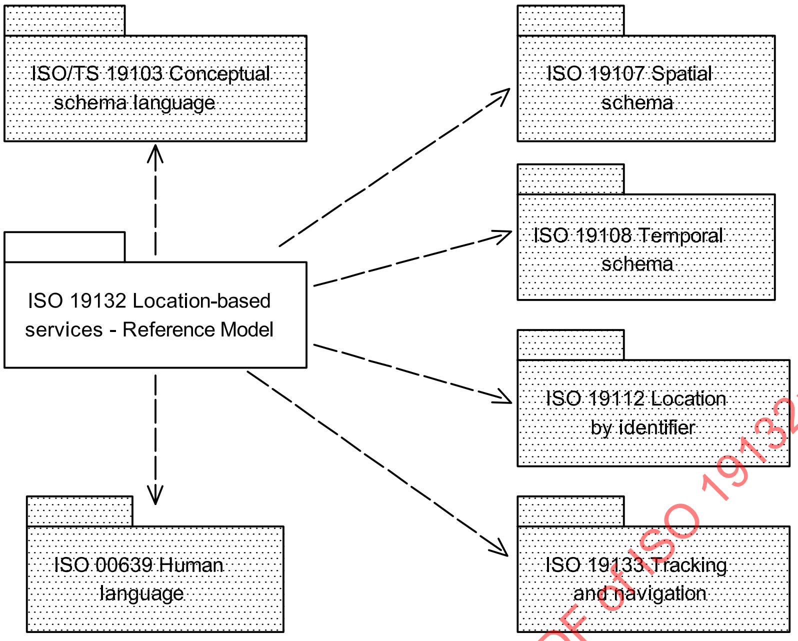

ISO/TS 19103 Conceptual schema language
ISO 19107 Spatial schema
ISO 19108 Temporal schema
ISO 19132 Location-based services - Reference Model
ISO 19112 Location by identifier
ISO 00639 Human language
ISO 19133 Tracking and navigation

Figure 5 — Package dependencies to other ISO standards

7.2 Package – LBS Participants

7.2.1 Scenarios and semantics

Location-based services, while predominantly regarded as applications that service mobile clients, also encompass applications related to the changing spatial relationship of both fixed and mobile objects. As described in Annex C in greater detail, these sorts of scenarios can include the “pull” and “push” of location information related to people, natural disaster events, civil emergency events, and public transport and traffic vehicles. Whilst this list is not exhaustive, it does emphasise the range of services that an LBS framework (and architecture) needs to include.

In Figure 6, the types of Enterprise view participants are laid out. The potential and logically necessary multiple “role playing” of individual objects is reflected in the use of UML types and the subclassing between roles that allows each participant to play the role of any of its supertypes.

In Figure 7, the communication channels of the Enterprise view are modelled as associations. In using this diagram to understand the “calling tree” of a particular application, two types of graph manipulations can occur: folding and unfolding. If an application provider acts as its own service broker, then these two steps of the calling tree fold in on one another and the “service broker” association role essentially disappears. The same sort of folding can occur if the service provider acts as its own broker, eliminating the service provider association role from the instance diagram. If both occur and the application provider provides “in-house” solutions to the user, only the user and the application provider (role and instance) are directly visible in the calling tree.

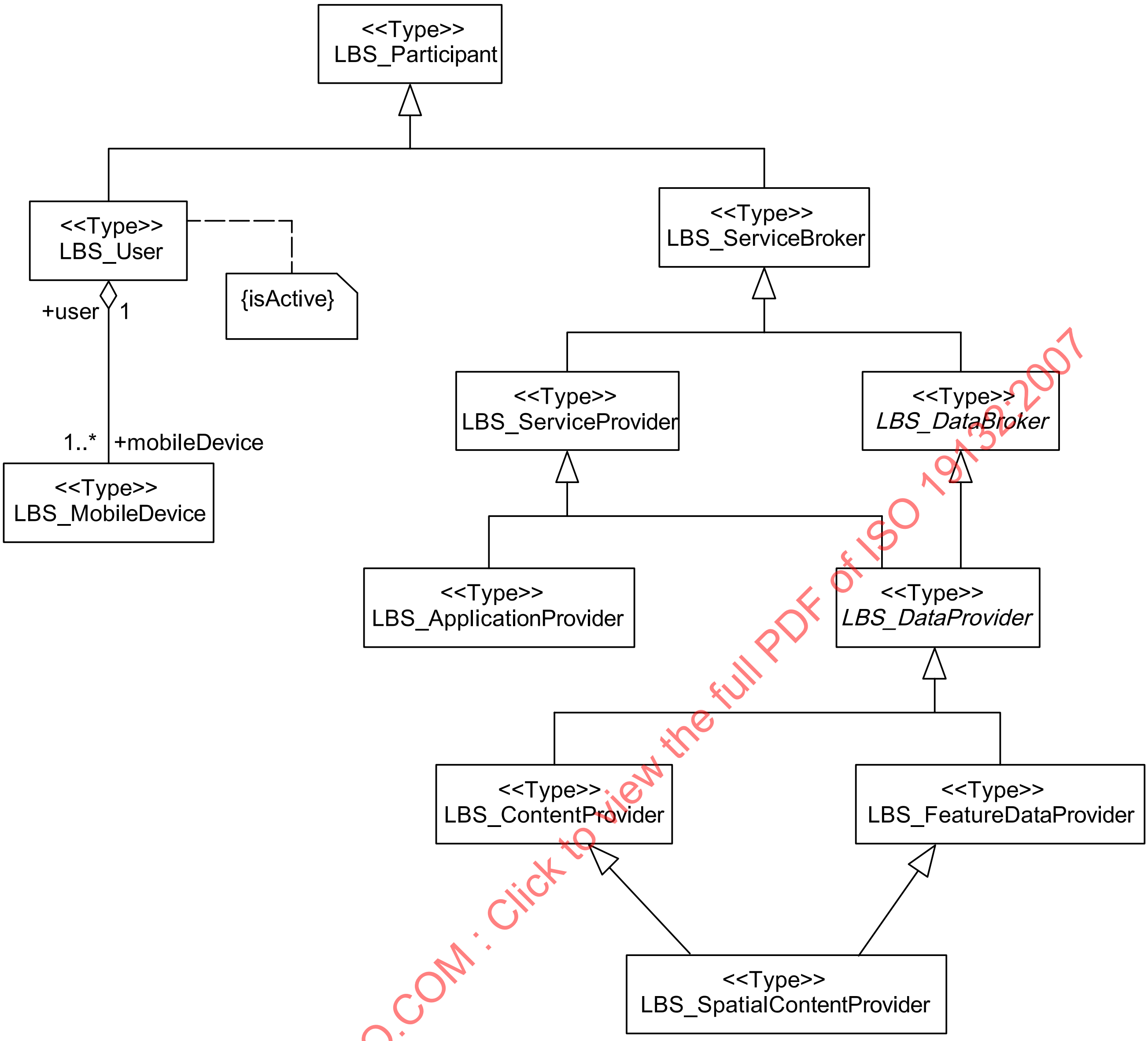


Figure 6 — Roles of the Enterprise view

The second sort of graph manipulation is seen when an object plays multiple roles for different services. If a broker binds multiple instances of services, then the service provider association from broker to service provider unfolds, giving us multiple instances of that type of link. Another sort of unfolding can occur when a service provider acts as a service broker for other service providers. This may provide a cascade of service-provider links in the calling tree. There are two separate but equally valid UML mechanisms to model the second unfolding: multiple inheritance and multiple instantiation. In multiple inheritance, a new classifier realizes both the service provider and service broker types for possibly different services. In languages such as Java, which do not support multiple inheritance, this can be accomplished in several different manners, but the most common is the use of interfaces. A Java class is capable of implementing multiple interfaces because these interfaces do not place multiple internal structural requirements on classes as is the case with multiple inheritance. Depending on the requirements of the implementation, this can be handled in many different ways.

In multiple instantiation, a single object acts correctly on different but unrelated operation protocols. This is quite common in internet services that can handle multiple types of service request “envelopes” using the same URI. Internally, there may be logic that delegates the tasks to appropriate internal objects, such as through a dispatch table that associates interface “events” to the appropriate “event handler”. This

International Standard makes no requirement and its profiles shall make no requirement as to how a conformant implementation handles this issue.

This UML package models the participants described above. It contains classifiers for LBS_ApplicationProvider, LBS_SpatialContentProvider, LBS_FeatureDataProvider, LBS_ContentProvider, LBS_ServiceProvider, LBS_ServiceBroker, LBS_DataProvider, LBS_User, LBS_Participant, LBS_MobileDevice, and LBS_DataBroker.

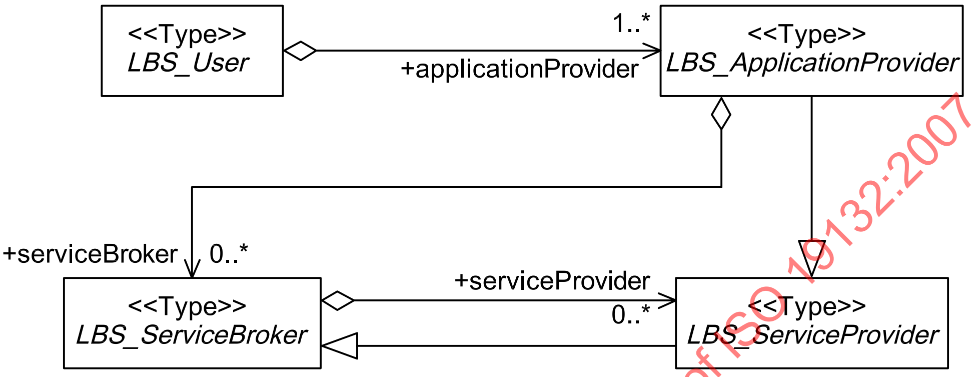


Figure 7 — Enterprise view communication channels as associations

Figures 6 and 7 provide a general overview of the participant model. Figure 7 delineates the major communication channel relationships, and Figure 6 shows the subtyping relationships between types. Both structures, together with the possibility of multiple instantiation (a concrete participant class implementing multiple interface types), support the wide variety of communication structures between client (user) and server (supplier) in potential LBS systems. These two diagrams supply sufficient information for the following type classifiers that usually negates the need for individual context diagrams.

Communication channel associations are unidirectional in order to correspond to the "request" direction of an operation or service invocation. Responses use the "return" information contained in the request, and therefore no continuous association is required.

7.2.2 Type – LBS_Participant

7.2.2.1 Class semantics

The participant class provides an abstract root for all of the members of the LBS community.

7.2.2.2 Association role – License

Participants use the license role to track rights granted to them by their associates, see Figure 8.

```
LBS_Participant::license[0..*]: LBS_License
```

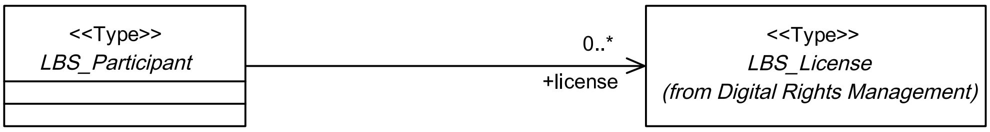


Figure 8 — License associations for LBS_Participant

7.2.3 Type – LBS_User

7.2.3.1 User view of the system

7.2.3.1.1 Users

The **users** are the ultimate active objects in the model and initiate, directly or indirectly, all actions in the system, making informational requests, usually as a service request, to meet their personal needs through lightweight browser-based applications. As such, they are modelled in this International Standard as **active objects**.

The service requests are delivered through a network supported by the **application provider** (see Figure 9). This network is usually one running systems specific to the hardware. These application- and device-specific network providers supply many services directly to the user, and provide others by calling Internet service providers or brokers for location-based services and integrating that functionality with their applications in order to support user request. The application and network provider plays a pivotal role in the system since he is ultimately responsible for supporting the user and, in many cases may be the only provider for some key services. For example, for cell phones that are not equipped with GNSS systems, the network provider may be the only feasible supplier of tracking information necessary to support the location-based services being requested. Further, the application provider is the ultimate arbiter of supply, often making the choice of Internet service supplier based on location-dependent information driven by user location and preferences.

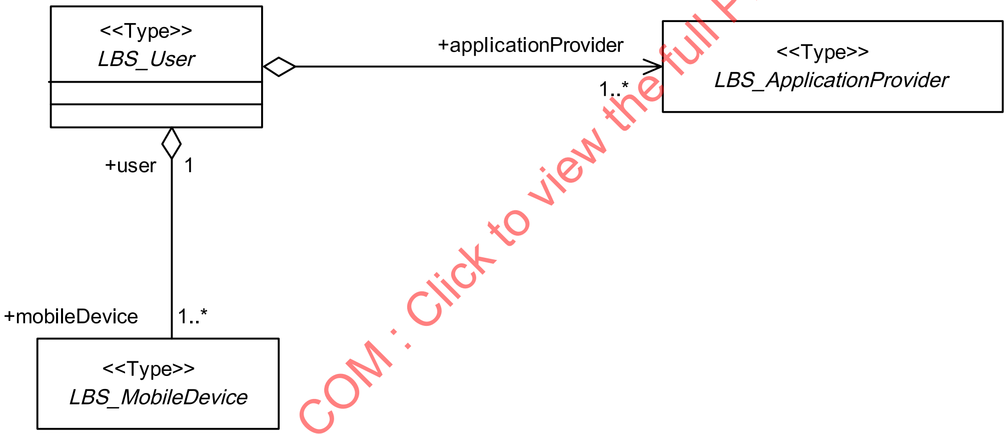


Figure 9 — LBS_User associations

7.2.3.1.2 Location-based services

7.2.3.1.2.1 Types of questions

In an environment in which things move or are movable, the answers to very simple questions change as the environment changes its spatial configuration. For example:

- Where am I?
- When will the next bus arrive?
- Where is the nearest restaurant with steak on the menu, which is open for lunch?
- Where are my car keys?
- What films are playing at nearby theatres?
- How do I turn off (on) the nearest light?

Any question or request for action that is dependent on the location of anything directly or indirectly involved in the request can be framed as a location-based service. If one or more of the items involved in that service request is moving or is capable of being moved, then the service can be framed in dynamic terms, essentially adding the command phrase “Inform me of any significant change to the status of these objects that would change the current response” to the request.

7.2.3.1.2.2 Scale of operation

The area in which an LBS service operates does not change its status as an LBS service. For example, the following questions have different associated scales of operation:

- Where is Beijing? – World scale
- Where is the best Chinese restaurant nearby? – Neighbourhood scale
- Where is my car? – Building scale
- Where are my car keys? – Personal scale

Some services are inherently “macroscopic” in scale, such as requesting round-trip airline tickets between Boston and Beijing. Others are restricted to a smaller neighbourhood scale, such as finding a conveniently located Chinese restaurant of good quality. The assumption is that the Boston-Beijing ticket should not be required for going to lunch, regardless of the quality of the restaurants in Beijing. Some services are at a much more personal scale, such as finding one's car keys which should be fairly close by.

7.2.3.1.3 Mobile devices

The **mobile devices** provide access to services and data to the users. Their ability to function in the virtual mobile, spatial world of location-based services is dependent on the capabilities of the **mobile device** and the network that acts as its gateway to the **service brokers** and **service providers**. At a minimum, the device shall be able to locate itself or be located by its network to some reasonable degree of accuracy to facilitate the location-based functionality of the service providers. The reporting of that location shall supply some form of location object (see ISO 19133) and estimate the accuracy of that location. That location should be sufficient to approximate the map position of the mobile device, and hence should be able to approximate the map position of its user or carrier.

Location capability falls into three basic categories:

a) self location devices usually based on embedded GNSS systems;

b) network location devices located by signal strength or direction finding in the receivers of a network communication system such as a cell phone or radio network with multiple antennae covering most areas (for location purposes);

c) passive devices such as RFID or printed bar-code tags that are sensed by external electronics, optical readers or other sensors

   - where the passive device or tag carries user information and is the tracking device, and location is reported when the tag is scanned by a sensor in a known location, or
   - where the sensor is the tracking device, and the tags are immovable and carry location information (essentially are “benchmarks” for an LBS tracking service) and the location is derived from the tag and reported by the sensor when it passes the tag.

In all cases, but especially for passive devices, the location may be enhanced by other means, such as

- planned route following, where there exists a planned route for the device or its vehicle, and the most current locations and schedule are extrapolated or interpolated to estimate current location (the planned route may be a physical restriction such as road or railways, or it may be a route that accompanied a submitted travel plan), or

- on-board inertial devices, where a location is used as a base point, and sensed motion and direction are used to extrapolate current position.

The mobile device may further enhance that location with other information that would be useful to various types of service providers. That additional information may include any or all of the following:

- the vertical position of the device, which allows the user to be located in 3D space for location-based services that can utilize that information;
- the bearing or “forward” direction of the device, which allows the user's current direction of motion, or direction of view, allowing the location-based service to transform between cardinal directions (north, east, west, south) and deictic references (left, right); this also allows the device to act as a pointing device in 2D space;
- the angle of inclination of the device that, with the bearing, gives the device the ability to act as a pointing device in 3D space;
- a forward image (such as from a camera phone);
- a display capable of some form of virtual reality or image display;
- an audio transmission and reception capability for a voice-based interface.

7.2.3.1.4 Tracking devices

Depending on the scale or the type of application, the strategy for tracking a moving thing will be different because of information requirements, economics or other pragmatic concerns. In LBS associated with passenger vehicles, the most appropriate mechanism may be a GNSS device, augmented with an electronic compass, and possibly further associated to inertial measurement devices to aid in instances where satellite signals are lost due to interference of any type, such as physical blockage, or EM signal from non-GNSS sources.

In other instances, a passive responder such as an unpowered RFID tag may be appropriate for the same vehicle to assure accuracy in measurements that assure the vehicle has passed a given point, such as through a tollbooth.

In tracking packages or household items involved in a micro-LBS environment, economics or use may require the use of RFID tags. In both these application scales, sufficient tracking sensors can be set into the environment to distinguish when an RFID passes important “choke points” such as doorways between rooms, or sort points within a sort-and-load facility for a package delivery service. This sort of approach may limit the type of response possible to a tracking request for a “last-seen” statement, such as:

Request:	Where is my package?
Response:	Loaded onto truck bound for Tulsa at 3:00 PM. Truck on schedule to arrive Tulsa at 7:30 PM.
Request:	Where are my car keys?
Response:	Last seen entering kitchen at 7:00 PM yesterday.

As long as systems “watch” the exit points associated to these sorts of location “state descriptions”, the responses should be considered sufficient for the purposes intended. Since RFID tags may be small enough to fit on a business card, priced about the same as such, the economic payback will often justify the apparent lack of spatial and temporal accuracy associated to such systems.

7.2.3.1.5 Location benchmarks and hybrid systems

A tag (RFID, or optical, such as a barcode) can carry any type of information. In the passive tracking case, the tag carries the user ID information and when sensed by a reader, the reader can determine the position of the traveller (tagged vehicle). By reversing the roles, tagging the locations and using sensors as tracking devices, when a tag is read, its location is read as data, and the sensor can then determine the location of its associated traveller.

In a hybrid system where GNSS and ID tag readers are combined in the tracking device, the sensor can feed the benchmarked location to the processor which can then use it to determine the systematic error associated to its GNSS device. Thus, a differential GNSS approach can improve the accuracy of the location information derived from the GNSS system. In a variation on this concept, inertial systems associated to GNSS systems can read the configuration of the roadway (location of turns and curves), match that to the local mapping data and adjust GNSS readings over a period of time to “fit” the geometry of the vehicle's route to the mapping geometry.

7.2.3.2 Basic requirements for users

Users of LBS are likely to be mobile, and therefore are likely to have some intermittency in their web connection. Furthermore, they are likely to require location-dependent services as they move from area to area. This puts certain requirements on the LBS framework and on some of its basic services.

1) Users need to receive device-independent output.

2) Users need a single-sign-on service broker.

3) Users need identity management from that broker to maintain security.

4) Services for mobile clients need to be immune to message dropping and message repetition.

5) Services should be loosely coupled to their implementation and to the specifics of the service interface that invokes them.

7.2.3.3 Association role – applicationProvider

The users are directly associated to applications that act as their gateway into the system. The application may be as simple as a browser, or as complex as a full-blown GIS application.

```
LBS_User::applicationProvider[1..*]: LBS_ApplicationProvider
```

7.2.3.4 Association role – mobileDevice

The user is also directly associated to devices which are either mobile or acting as if they were. Non-mobile devices can invoke LBS services and operations by supplying either their actual location or hypothetical ones. The actions of the system should not be significantly different for the two cases, unless the digital rights constraints require actual locations to meet criteria. This association is bidirectional, since the user's device must be able to identify its user to clear digital rights issues.

```
LBS_User::mobileDevice[1..*]: LBS_MobileDevice
```

7.2.4 Type – LBS_ApplicationProvider

7.2.4.1 Class semantics

The **application provider** is a **service provider** that supplies services for a specific type of user, usually bundling related services and data into a usable suite of functionality aimed at specific needs (see Figure 10). The application provider can work through a service broker or supply his services directly. Each **user** shall have at least one application directly linked to it, but that application may pass through requests from the user to other providers as part of a delegation chain.

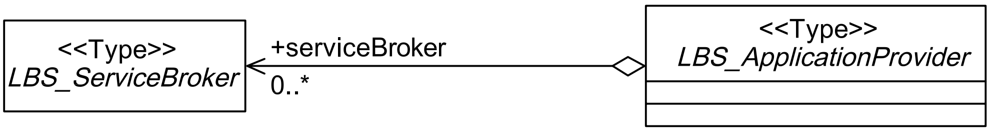


Figure 10 — LBS_ApplicationProvider associations

7.2.4.2 Association role – serviceBroker

The service broker aggregation lists the brokers (or direct service providers) used by the application. The association is not bidirectional, since all communications from the broker to a client application will be accomplished by “unction” or “service” responses.

```
LBS_ApplicationProvider::serviceBroker[0..*]: LBS_ServiceBroker
```

7.2.5 Type – LBS_DataProvider

The **data provider** is a **service provider** that supplies access to data. **Data providers** provide services in the form of data, based on queries or simple requests. Like their supertype **service providers**, they may act either through a broker or as their own broker. The interfaces used for the service will involve either a free form query or sets of query templates for which the user of the service will supply parameter values.

7.2.6 Type – LBS_FeatureDataProvider

The **feature data provider** is a **data provider** that supports spatial extensions to the query interface, either through a spatial query language or through spatial parameters to query templates. Feature data providers supply data in the form of “features”. These data packages describe and represent any item as defined by its application schema (see ISO 19109 and ISO 19110 for descriptions of feature schemas).

7.2.7 Type – LBS_ContentProvider

The **content provider** is a **data provider** who supplies data in the normal web, non-feature formats (such as HTML or XML) through fixed labels (such as URLs). These act as pre-executed, potentially stored and preformatted, data content

7.2.8 Type – LBS_SpatialContentProvider

The **service providers** that supply spatial content are **spatial content providers**. Spatial content is web-based spatial data and spatially related data accessible through spatial query or through fixed labels. Spatial content is normally linked to using uniform resource locators (URL) or as web services. It may contain feature and non-feature content, usually linked to form a connected, consistent representation of facts.

7.2.9 Type – LBS_ServiceProvider

7.2.9.1 Class semantics

The Internet **service provider** supplies location-based services through standard protocols. These service providers will engage in contractual relationships with the service brokers and **application providers** for supplying particular services for specific areas or localities. It is unlikely that the Internet service provider and the application provider will be in direct competition for users because of the device-dependent nature and functionality-bundling nature of the applications in question. In some cases, the application provider may choose to provide many services directly to the users, but the enormous data demands of maintaining an acceptable quality of service and currency of data over wide areas to support wide ranging users will probably make contracting with local suppliers of data and services more attractive than the expensive proposition of maintaining data bases with wide area coverages. At a minimum, the **service provider** may act as **service broker** for his own service.

7.2.9.2 Association role – dataProvider

The service provider shall have access to the pertinent data to support his operations. It will therefore have persistent information to connect (either externally or internally) to data providers. The service and connected data providers will have an established relation, and will have all digital rights management issues and procedures settled and agreed upon.

```
LBS_ServiceProvider::dataProvider[0..*]: LBS_DataProvider
```

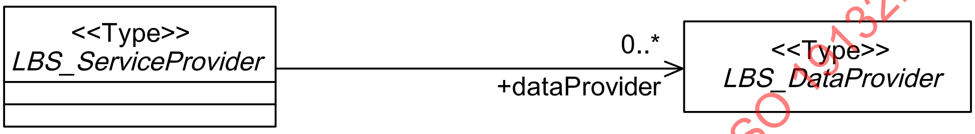


Figure 11 — Service provider associations

7.2.10 Type – LBS_ServiceBroker

7.2.10.1 Class semantics

The **service broker** is similar to the application provider except that it provides services in a specific API to other service and application providers (Figure 12). The major purpose of the broker is to hide the actual location and interface of the more primitive service. A service broker can also be used to change the details of the API so that the actual details of the service provider can be completely hidden. Thus, the service broker can be both business facilitator and ODP binder. The service broker can be collapsed out of the model if the service and application providers act as their own broker.

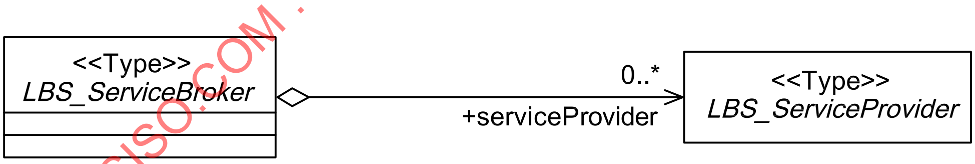


Figure 12 — Service broker associations

7.2.10.2 Association role – serviceProvider

The serviceProvider aggregation role provides linkages to the service provider supporting the broker's offerings.

```
LBS_ServiceBroker::serviceProvider[0..*]: LBS_ServiceProvider
```

7.2.11 Type – LBS_MobileDevice

7.2.11.1 Class semantics

The mobile device type is an abstraction of the user's method of access, and the enabler for tracking. It is the mobile device that is tracked, with the assumption that the device and its user are co-located.

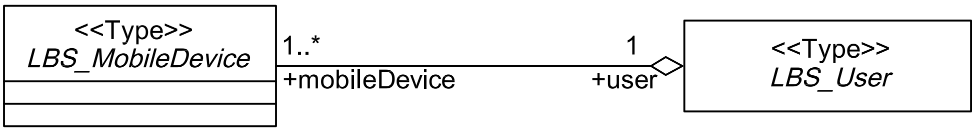


Figure 13 — Mobile device associations

7.2.11.2 Association role – User

The role user associates the device to information about the user, and allows it to act using the user licenses and rights.

```
LBS_MobileDevice::user: LBS_User
```

7.2.12 Type – LBS_DataBroker – Class semantics

The Internet **data broker** is similar to the **service broker** except that it provides brokered data services. The major purpose of the broker is to provide users and the application a single access point to clear licensing issues associated to data use. A data broker may also be used to change the details of the API so that the actual details of the data provider can be completely hidden. Thus, the service broker can be both business facilitator and ODP binder.

8 Service model

8.1 Package – LBS_Services

8.1.1 Package structure

The root services for LBS are in the subpackages of this package. These subpackages include: Basic Services, Geomatics Services, Information Services, System Management, and Digital Rights Management. Figure 14 shows the contents of these packages. The level of integration between these services may vary according to the architecture of the compliant system. But user application should not depend on internal service implementation, and thus should be loosely coupled with these and other services.

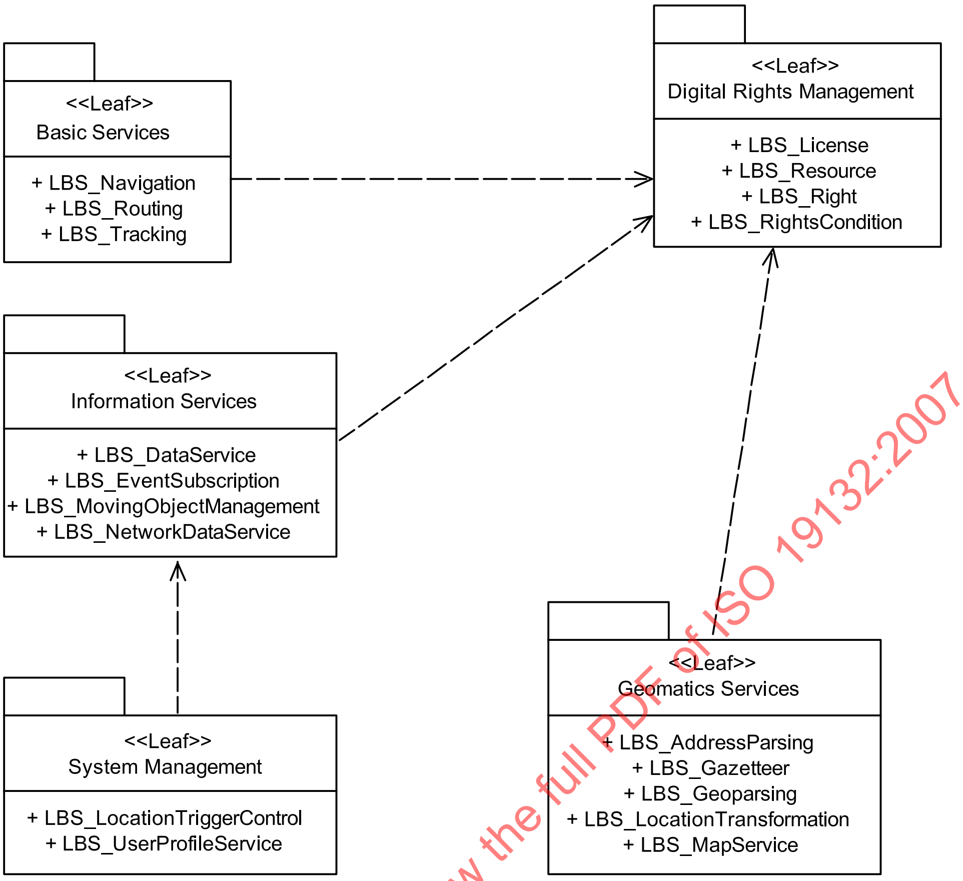


Figure 14 — Subpackages of LBS_Services

8.1.2 Service taxonomy

The non-composite services used for LBS application can be divided into four subtypes:

— Location services (see 8.2);

— Geomatics services (see 8.3);

— Information services (see 8.4);

— System and User management (8.5).

Hybrid services using functionality from more than one arena are by nature composite services. Implementation of such services may choose alternative communication mechanisms between basic services for use in hybrid service implementation.

8.2 Package – Basic Services

8.2.1 Package structure

The Basic Services package contains the root classes for the basic services for LBS. These include the classes LBS_Tracking, LBS_Routing, and LBS_Navigation, see Figure 15. Under a digital rights management system these services and related digital items (data, metadata, identification or security tokens, etc.) are resources that may require a license exchange.

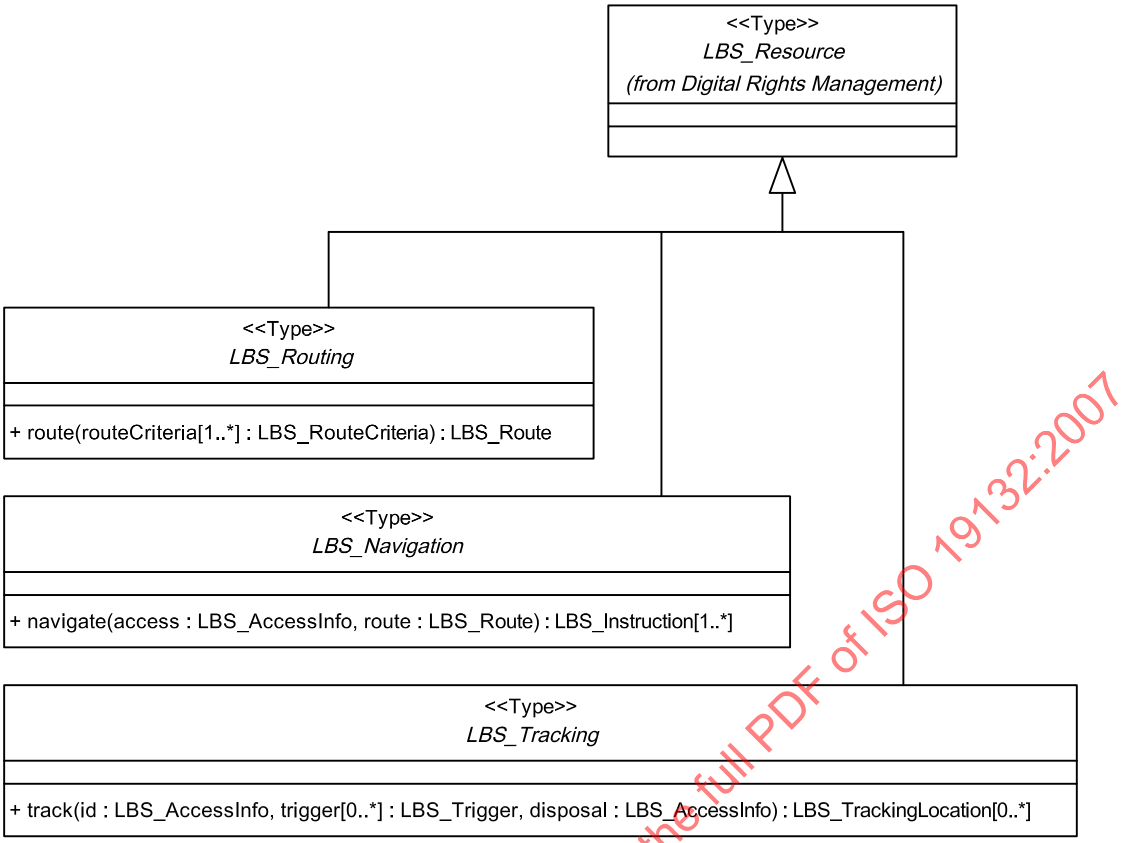

<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_Routing
+ route(routeCriteria[1..*] : LBS_RouteCriteria) : LBS_Route
<<Type>>
LBS_Navigation
+ navigate(access : LBS_AccessInfo, route : LBS_Route) : LBS_Instruction[1..*]
<<Type>>
LBS_Tracking
+ track(id : LBS_AccessInfo, trigger[0..*] : LBS_Trigger, disposal : LBS_AccessInfo) : LBS_TrackingLocation[0..*]

Figure 15 — Basic services

8.2.2 Type – LBS_Tracking

8.2.2.1 Tracking a mobile device

Regardless of whether the mobile device can locate itself (such as those equipped with GNSS systems internally) or must be located through its network (such as cell phone triangulation based on signal strength at more than one receiving tower), the interface to the position determination (tracking) is the same. The inputs are

a) mobile client identity,

b) access information (to gain permission to access the mobile client's information),

c) trigger information (if multiple positions are required),

d) disposal instructions.

The outputs to the tracking request are a sequence of tracking locations sent as per disposal instructions generated by the client device.

In the case of passive object tracking (such as those based on RFID), the triggering mechanism shall always involve the object coming in range of a sensor. Tracking history can be maintained by including the storage of the data. Tracking is covered in detail in ISO 19133.

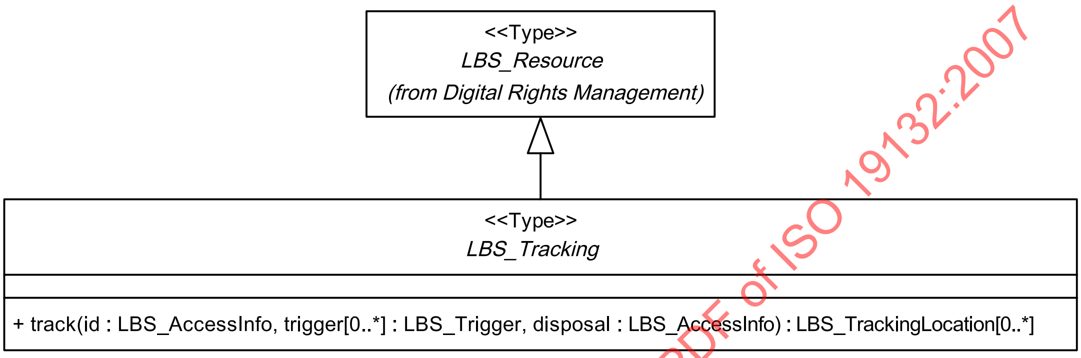
8.2.2.2 Class semantics

The “LBS_Tracking” service checks the credentials of the requestor, and then, if appropriate, reports the position of the indicated user as per the triggers specified in the request.

8.2.2.3 Operation – track

The operation “track” uses id: LBS_AccessInfo to identify the item and determine the digital rights to track that item, the trigger list (trigger[0..*]: LBS_Trigger) to determine when location is to be reported, and the disposal information (disposal: LBS_AccessInfo) to route the results. Its return is a sequence of tracking locations (possibly asynchronously delivered). Figure 16 is a UML context diagram for tracking.

```
LBS_Tracking::track( id: LBS_AccessInfo,
      trigger[0..*]: LBS_Trigger,
      disposal: LBS_AccessInfo): LSB_TrackingLocation[1..*]
```



<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_Tracking
+ track(id : LBS_AccessInfo, trigger[0..*] : LBS_Trigger, disposal : LBS_AccessInfo) : LBS_TrackingLocation[0..*]

Figure 16 — Context Diagram: LBS_Tracking

8.2.3 Type – LBS_Routing

8.2.3.1 Routing services

A routing service calculates an optimal route, usually within a network. The inputs of such a service are

a) Start point,

b) Waypoints (optional),

c) Endpoint, and

d) Preferences.

The returned response contains: Route description or handle.

Routing is covered in ISO 19133 and ISO 19134.

8.2.3.2 Class semantics

The “LBS_Routing” service is functionality that finds optimal or near optimal routes meeting requested criteria. The specifics of the type of request that can be made, the type of cost functions that can be used, and the types of constraints and preferences that can be specified would all be part of the registered metadata of the routing service. Figure 17 is a UML context diagram for routing.

8.2.3.3 Operation – route

The operation “route” in a routing service finds the optimal or near-optimal route meeting as many of the criteria passed to it as possible. The returned route must start and end where selected, and pass through the waypoints during its extent. It shall meet the criteria, and as many of the preferences as practical. The route criteria may be passed as individual subroutes, each ending where the next begins. This is useful if a schedule or list of points need be reached in a particular order. The resultant route should meet the criteria for each leg or subroute of the journey.

The parameter "routeCriteria" (routeCriteria[1..*]: LBS_RouteCriteria) contains criteria for each leg of the journey that will result in a set of continuous subroutes in the returned route. Each subroute will be the optimal or near-optimal route meeting the criteria for that section. The mechanism for handling overall criteria or personal criteria is left to the implementation of the service.

The return "LBS_Route" is an optimal or near-optimal solution to the problem posed by the request. Although the algorithms involved are capable, with most commonly encountered networks, of always finding an optimal solution, heuristics introduced to satisfy "fuzzy" constraints or multiple non-unified cost functions may introduce ambiguity into the process.

```
LBS_Routing::route( routeCriteria[1..*]: LBS_RouteCriteria ): LBS_Route
```

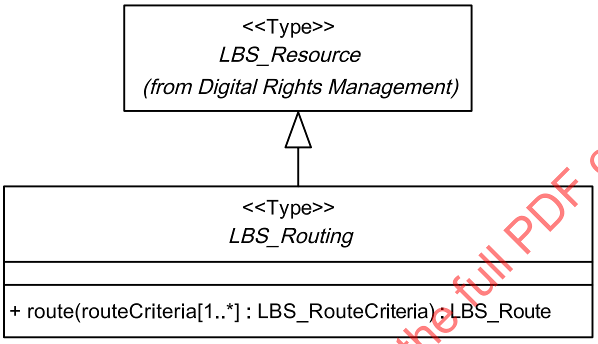

<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_Routing
+ route(routeCriteria[1..*] : LBS_RouteCriteria) : LBS_Route

Figure 17 — Context Diagram: LBS_Routing

8.2.4 Type – LBS_Navigation

8.2.4.1 Navigation services

Navigation is the execution of a route. A navigation service will, given a route and tracking information, return a stream of instructions for following or refinding that route appropriate for the positions returned. Instructions can include local navigation maps or cartoons (simple view simulation) for the execution of the route. The service inputs are

a) Route,

b) Tracking identity, and

c) Access information.

Detailed driving directions or detailed walking directions services are services whose instruction returns are appropriate for their respective clients. They shall generally follow the same pattern as above.

Navigation is covered in ISO 19133 and ISO 19134.

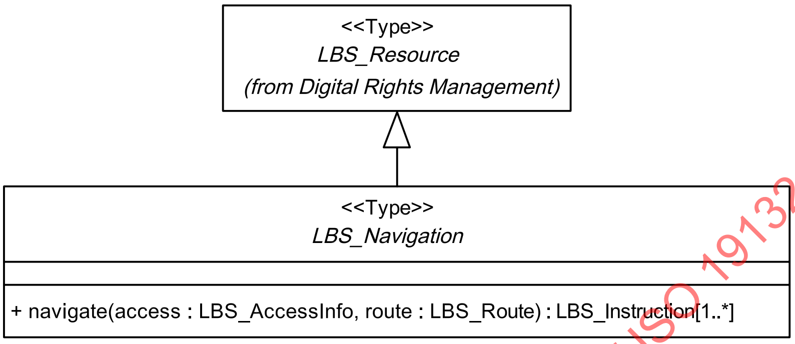
8.2.4.2 Class semantics

The type "LBS_Navigation" is the basic navigation service interface. Navigation is the execution of a calculated route. Dynamic navigation is navigation where the route may be altered during its execution due to changed or changing circumstances. Figure 18 is a UML context diagram for navigation.

8.2.4.3 Operation – navigate

The operation “navigate” uses access information (access: LBS_AccessInfo) and route information (route: LBS_Route) to create a list of instructions to traverse the route (LBS_Instructions).

```
LBS_Navigation::navigate( access: LBS_AccessInfo, route: LBS Route ):
      LBS_Instruction[1..*]
```



<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_Navigation
+ navigate(access : LBS_AccessInfo, route : LBS_Route) : LBS_Instruction[1..*]

Figure 18 — Context Diagram: LBS_Navigation

8.3 Package – Geomatics services

8.3.1 Package structure

The Geomatics services package contains root classes for each of the LBS services most likely to derive from more standard geographic information technology. These classes include LBS_LocationTransformation, LBS_AddressParsing, LBS_Geoparsing, LBS_Gazetteer, and LBS_MapService. Figure 19 gives an overview of these services.

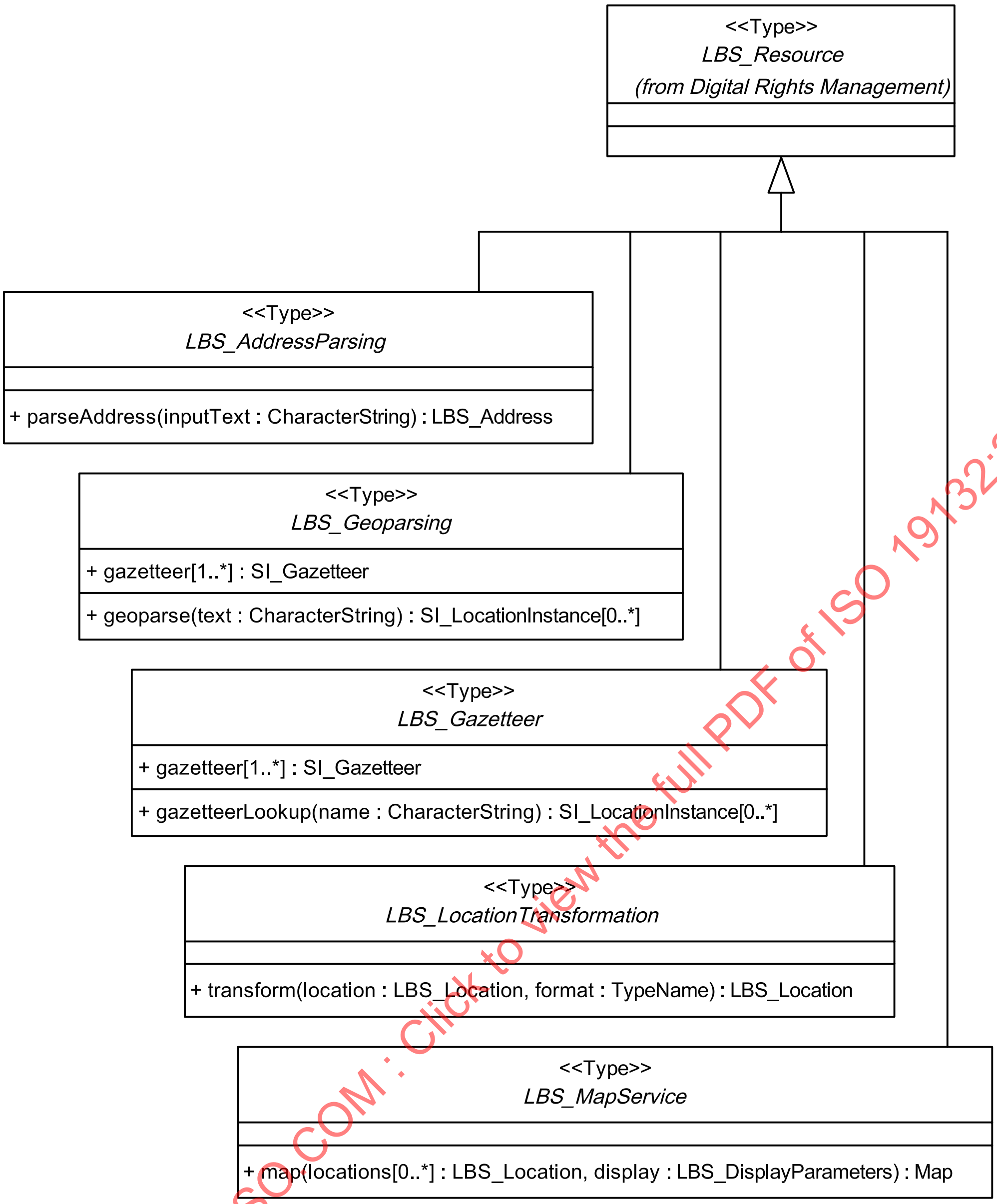

<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_AddressParsing
+ parseAddress(inputText : CharacterString) : LBS_Address
<<Type>>
LBS_Geoparsing
+ gazetteer[1..*] : SI_Gazetteer
+ geoparse(text : CharacterString) : SI_LocationInstance[0..*]
<<Type>>
LBS_Gazetteer
+ gazetteer[1..*] : SI_Gazetteer
+ gazetteerLookup(name : CharacterString) : SI_LocationInstance[0..*]
<<Type>>
LBS_LocationTransformation
+ transform(location : LBS_Location, format : TypeName) : LBS_Location
<<Type>>
LBS_MapService
+ map(locations[0..*] : LBS_Location, display : LBS_DisplayParameters) : Map

Figure 19 — Geomatics services

8.3.2 Type – LBS_LocationTransformation

8.3.2.1 Location Transformations

There are many ways to represent location, both analytically (by coordinate systems) or descriptively, such as place names. Any of these is a viable location reference.

Location references include:

- Direct position with its Coordinate reference system ID;
- Linear reference system location specification;
- Place name;
- Map Position (see 8.3.6.1);

- Address;
- Telephone number;
- Feature identity.

A location transformation service takes:

a) a location of any type;

b) a location format name or description;

and returns:

c) a location in the new format;

d) a quality of service (accuracy) estimate.

Giving a location expression (in one form from a list of types) and a requested type (in one form from that same list) returns the same location (in the requested form) and a quality of service estimate (how good is your new location representation?, with a temporal window of validity if needed).

Coordinate transformation services are location transformation services that are capable of transforming coordinates in one system to coordinates in another.

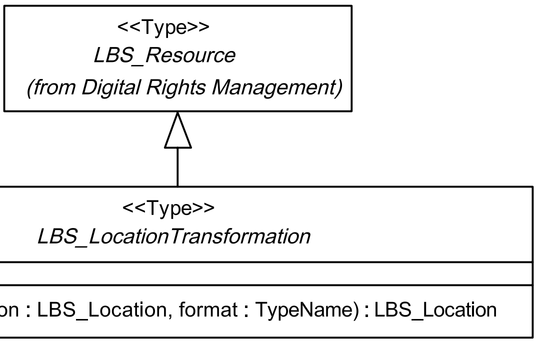
Linear reference system services are location transformation services that are capable of inputting location references with respect to a linear reference system, as defined in ISO 19133. Geocoding is a location transformation service that takes address locations and produces coordinate locations. Reverse Geocoding does the opposite: determining a complete normalized place name/street address/postal code given a geographic position's coordinates.

Location transformation is covered in more detail in ISO 19133. Figure 20 is a UML context diagram for LBS_LocationTransformation.

8.3.2.2 Operation – transform

The operation “transform” takes any supported location representation, and transforms it into a near equivalent location in another representation.

```
LBS_LocationTransformation::transform(
      location: LBS_Location, format: TypeName ): LBS_Location
```



<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_LocationTransformation
+ transform(location : LBS_Location, format : TypeName) : LBS_Location

Figure 20 — Context Diagram: LBS_LocationTransformation

8.3.3 Type – LBS_AddressParsing

8.3.3.1 Address parsing

An address parsing service shall take the text of an address, and produce a parsed address where address elements and subelements have been identified, parsed according to country-specific address formats. For example, the elements of an address can be identified as street, city, state and postal code. This is useful for cleaning up and standardizing address databases and for enabling users to find one-line addresses.

8.3.3.2 Operation – parseAddress

The operation "parseAddress" takes a free form address as a text string and parses it into its address elements. Figure 21 is a UML context diagram for LBS_AddressParsing.

```
LBS_AddressParsing::parseAddress( inputText: CharacterString ): LBS_Address
```

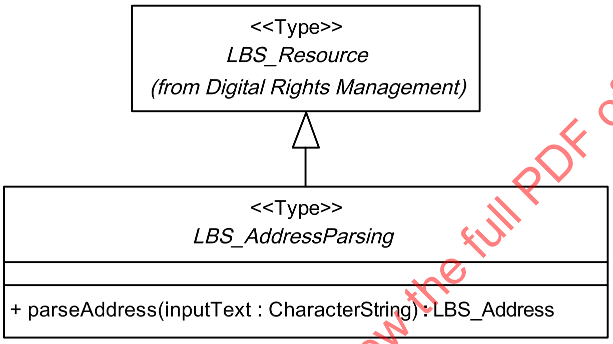

<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_AddressParsing
+ parseAddress(inputText : CharacterString) : LBS_Address

Figure 21 — Context Diagram: LBS_AddressParsing

8.3.4 Type – LBS_Geoparsing

8.3.4.1 Geoparsing

A geoparsing service input is: text containing location references, such as place names, addresses, phone numbers, or postal codes.

The response of a geoparsing service is: location references or coordinates.

Figure 22 is a UML context diagram for LBS_Geoparsing.

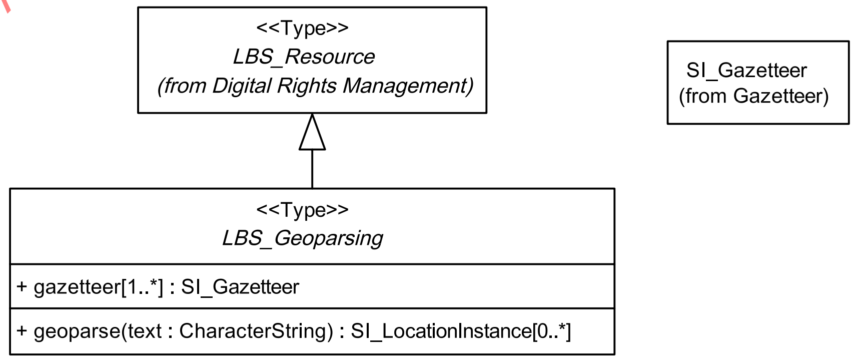

<<Type>>
LBS_Resource
(from Digital Rights Management)
SI_Gazetteer
(from Gazetteer)
<<Type>>
LBS_Geoparsing
+ gazetteer[1..*] : SI_Gazetteer
+ geoparse(text : CharacterString) : SI_LocationInstance[0..*]

Figure 22 — Context Diagram: LBS_Geoparsing

8.3.4.2 Attribute – gazetteer

The “gazetteer” attribute lists the various location gazetteers used by the geoparser.

```
LBS_Geoparsing::gazetteer[1..*]: SI_Gazetteer
```

8.3.4.3 Operation – geoparse

The operation “geoparse” takes text and returns a list of references to locations contained in that text.

```
LBS_Geoparsing::geoparse(
      text: CharacterString ): SI_LocationInstance[0..*]
```

8.3.5 Type – LBS_Gazetteer

8.3.5.1 Gazetteer

Gazetteer services input

a) Place or landmark, i.e.

   1) Names, or

   2) Query based on name or description,

b) Format requested.

Gazetteer services return the location of that landmark either as an address (if available) or some other location format.

Gazetteer functionality is described in ISO 19112. Figure 23 is a UML context diagram for LBS_Gazetteer.

8.3.5.2 Attribute – gazetteer

The attribute “gazetteer” lists the gazetteers accessible through this service.

```
LBS_Gazetteer::gazetteer[1..*]: SI_Gazetteer
```

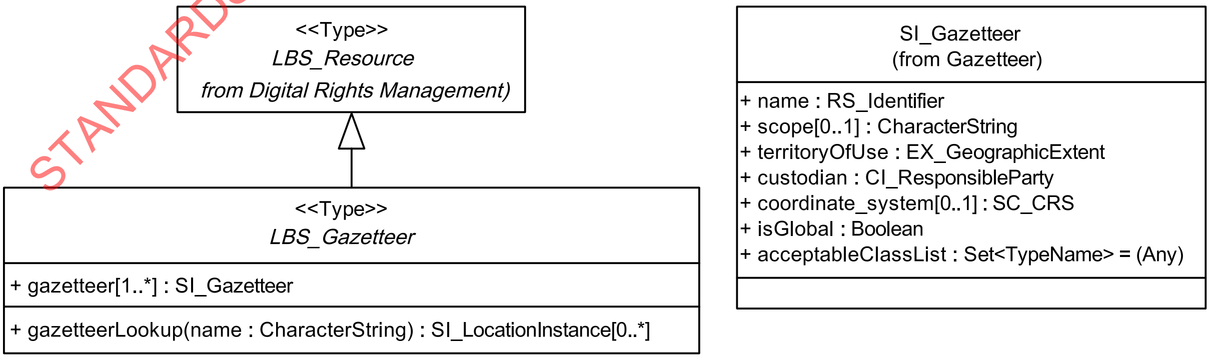

<<Type>>
LBS_Resource
from Digital Rights Management)
<<Type>>
LBS_Gazetteer
+ gazetteer[1..*] : SI_Gazetteer
+ gazetteerLookup(name : CharacterString) : SI_LocationInstance[0..*]
SI_Gazetteer
(from Gazetteer)
+ name : RS_Identifier
+ scope[0..1] : CharacterString
+ territoryOfUse : EX_GeographicExtent
+ custodian : CI_ResponsibleParty
+ coordinate_system[0..1] : SC_CRS
+ isGlobal : Boolean
+ acceptableClassList : Set<TypeName> = (Any)

Figure 23 — Context Diagram: LBS_Gazetteer

8.3.5.3 Operation – gazetteerLookup

The operation "gazetteerLookup" takes place names and returns location instances, usually as geometric objects or direct positions.

```
LBS_Gazetteer::gazetteerLookup(name: CharacterString ): SI_LocationInstance[0..*]
```

8.3.6 Type – LBS_MapService

8.3.6.1 Map services

The map services, Figure 24, retrieve or create displays of 2-D maps or 3-D virtual realities. The input of such service requests consists of

a) location equivalent (tracking information or location),

b) scale and size,

c) display parameters (optional, dependent on format type),

d) format of requested map.

The output of the service is a file containing a displayable 2-D or 3-D image: Map centred at location, of scale and of size, in requested format.

If the displayable image is interactive, then the mobile device should be able to select a position in the image and transmit that location as part of an additional service request (containing the original map parameters to allow the service provider to determine the location of the selection). The response to such a request may be an updated map or can be used in any service request requiring position.

If the format is for a base map, then the service will supply a standard map, potentially precompiled and stored *a priori*.

8.3.6.2 Operation – map

The operation "map" produces a map containing the location in the locations parameters using the display parameters passed in. A Map (the return value of the operation map) may be any displayable datatype such as an image or a vector rendition of a graphic map.

```
LBS_MapService::map( locations[0..*]: LBS_Location, display:
      LBS_DisplayParameters ): Map
```

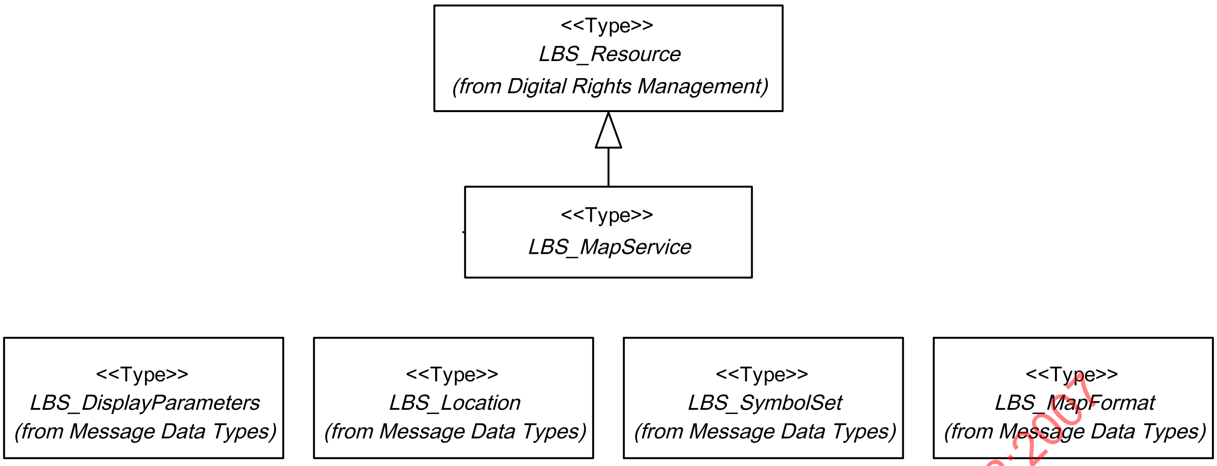


Figure 24 — Context Diagram: LBS_MapService

8.4 Package – Information Services

8.4.1 Package structure

The Information Services package contains data retrieval (query) services, including: LBS_DataService, LBS_NetworkDataService, LBS_MovingObjectManagement, and LBS_EventSubscription, see Figure 25.

8.4.2 Type – LBS_DataService

8.4.2.1 Data Services semantics

A data service (Figure 26) supports queries against a feature or other data schema and returns formatted results in the manner requested by the client. The input to a data service consists of

a) a query in the schema supported by the service,

b) control information on how the data is to be returned including information like data format and number of answers per return block.

The return of the service consists of

1) a dataset in the requested schema, in the requested format,

2) optional stream information if only partial results were returned.

Place of Interest (POI) services are data services with well-known feature types and schemas consisting of places (defined by particular feature types) that are commonly queried within the application. For example, “Find all ATMs in the airport”, “Find all restaurants near my meeting place” or “Find places where I can take my morning jog near my hotel”. Searches can be performed for POIs along a specified route.

A “Finding nearby places” service would include a proximity point of reference for the mobile user (usually representing its current location or some future or potential location) and would find features based on description and proximity to mobile user. The return list can be sorted based on the feature's proximity to a selected point.

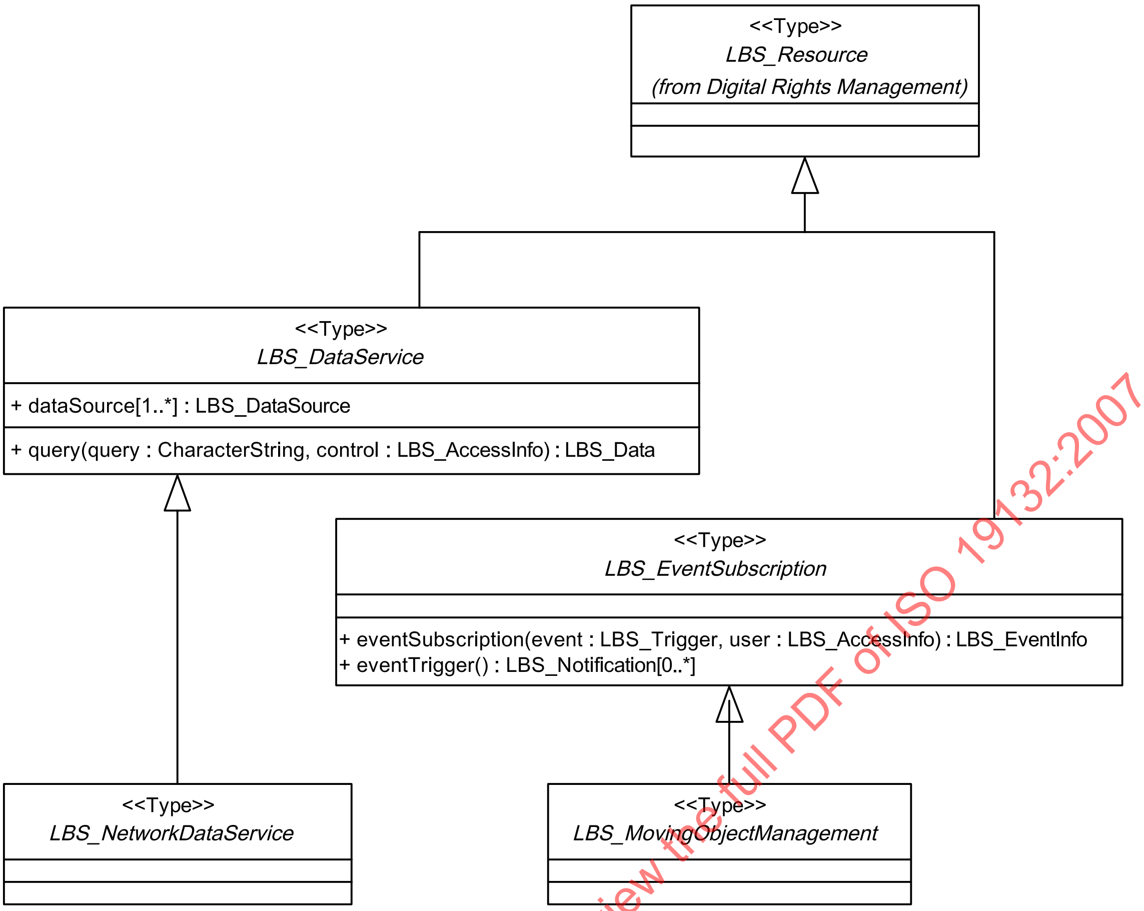

<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_DataService
+ dataSource[1..*] : LBS_DataSource
+ query(query : CharacterString, control : LBS_AccessInfo) : LBS_Data
<<Type>>
LBS_EventSubscription
+ eventSubscription(event : LBS_Trigger, user : LBS_AccessInfo) : LBS_EventInfo
+ eventTrigger() : LBS_Notification[0..*]
<<Type>>
LBS_NetworkDataService
<<Type>>
LBS_MovingObjectManagement

Figure 25 — Information services

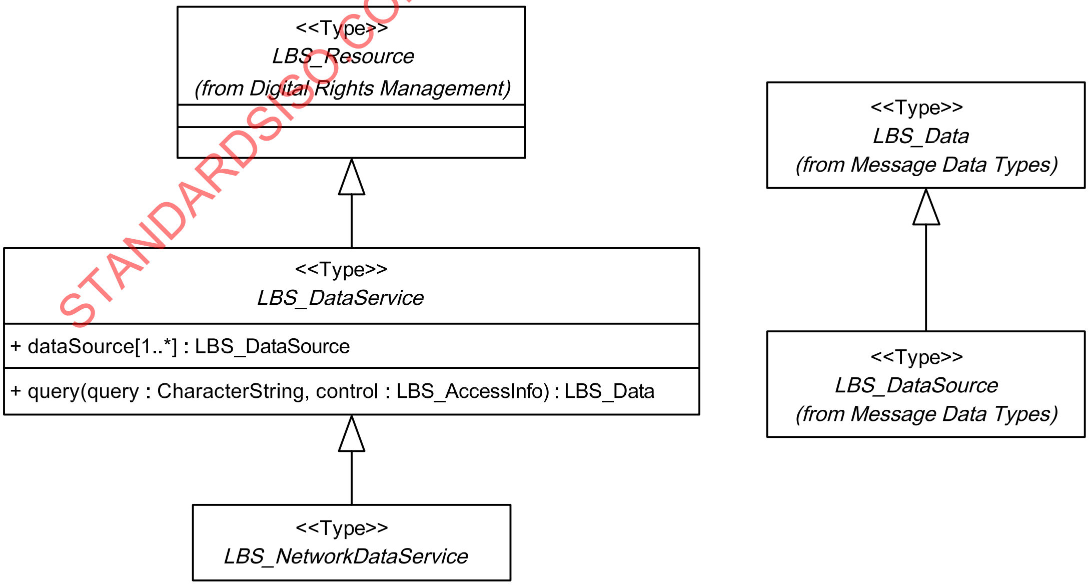

<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_Data
(from Message Data Types)
<<Type>>
LBS_DataService
+ dataSource[1..*] : LBS_DataSource
+ query(query : CharacterString, control : LBS_AccessInfo) : LBS_Data
<<Type>>
LBS_DataSource
(from Message Data Types)
<<Type>>
LBS_NetworkDataService

Figure 26 — Context Diagram: LBS_DataService

8.4.2.2 Attribute – dataSource

The attribute “dataSource” carries information about the data sources supporting this data service.

```
LBS_DataService::dataSource[1..*]: LBS_DataSource
```

8.4.2.3 Operation – query

The operation “query” is the function protocol for data access by query. The query language supported should be a version of SQL as defined in ISO/IEC 9075.

```
LBS_DataService::query(
      query: CharacterString, control: LBS_AccessInfo ): LBS_Data
```

8.4.3 Type – LBS_NetworkDataService – semantics

Network data services (Figure 27) support client side navigation and similar applications. They supply data coverage for an area of any size, from single records to larger area network data structures sufficient to support navigation as defined in ISO 19133. The input to such requests consists of the following:

a) Location, or area description

b) Query using that location as parameter, and schema information.

The return of such a request consists of the following: Network dataset in requested schema.

Data structures for features and for networks shall be consistent with ISO 19109, ISO 19110, ISO 19133 and ISO 19136, where applicable.

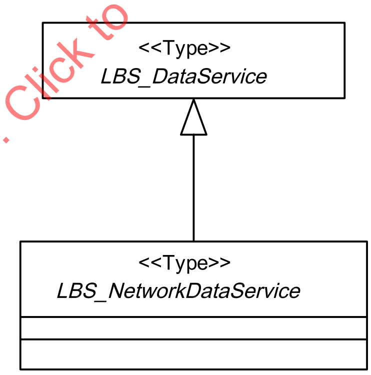


Figure 27 — Context Diagram: LBS_NetworkDataService

8.4.4 Type – LBS_EventSubscription

8.4.4.1 Event information or subscription services

Event information services (Figure 28) are subscription services that notify users of events as they occur. The inputs to the subscription are:

a) a query identifying the triggering event for which notification is requested;

b) tracking identities and access information if user position is needed for trigger;

c) timing information for testing of trigger;

d) disposal instructions that include the query to be answered upon trigger.

The returned response includes:

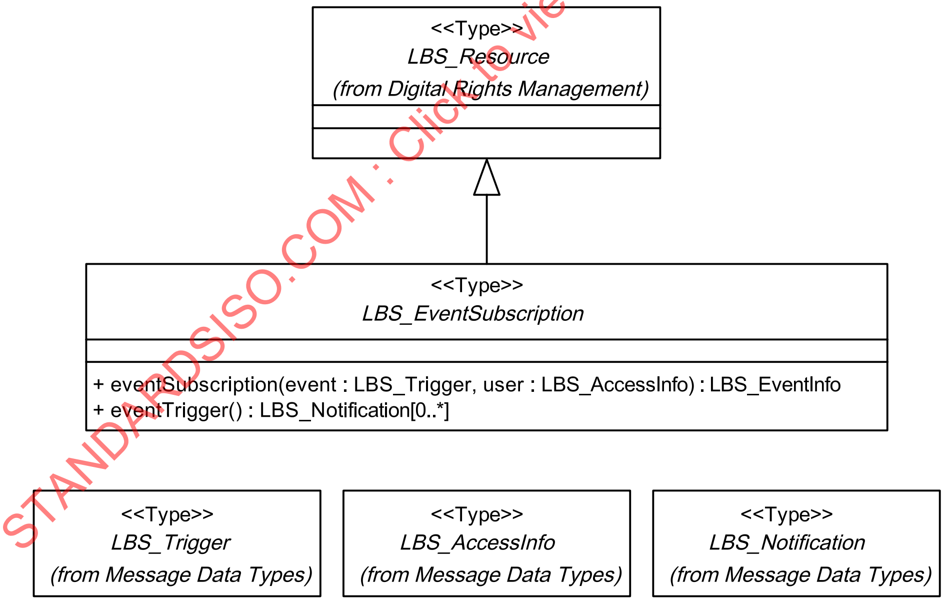
1) time and location of event,

2) additional information as per triggered query.

8.4.4.2 Operation – eventSubscription

The “eventSubscription” operation engages a watch for a triggering event that would cause the user (as identified by the access information) to be notified.

```
LBS_EventSubscription::eventSubscription(
      event: LBS_Trigger, user: LBS_AccessInfo): LBS_EventInfo
```



<<Type>>
LBS_Resource
(from Digital Rights Management)
<<Type>>
LBS_EventSubscription
+ eventSubscription(event : LBS_Trigger, user : LBS_AccessInfo) : LBS_EventInfo
+ eventTrigger() : LBS_Notification[0..*]
<<Type>>
LBS_Trigger
(from Message Data Types)
<<Type>>
LBS_AccessInfo
(from Message Data Types)
<<Type>>
LBS_Notification
(from Message Data Types)

Figure 28 — Context Diagram: LBS_EventSubscription

8.4.4.3 Operation – eventTrigger

The operation “eventTrigger” sets a subscription to receive notification of certain events when they occur. The notification will normally be asynchronous.

```
LBS_EventSubscription::eventTrigger(  ): LBS_Notification[0..*]
```

8.4.5 Type – LBS_MovingObjectManagement

The management of multiple moving objects through the type "LBS_MovingObjectManagement" (Figure 29) allows brokers to subscribe to tracking sequences for as many objects as they have appropriate access information and proof of rights to do so.

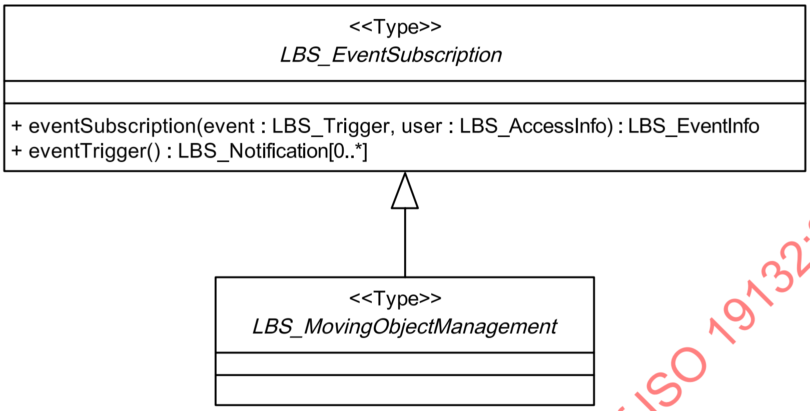

<<Type>>
LBS_EventSubscription
+ eventSubscription(event : LBS_Trigger, user : LBS_AccessInfo) : LBS_EventInfo
+ eventTrigger() : LBS_Notification[0..*]
<<Type>>
LBS_MovingObjectManagement

Figure 29 — Context Diagram: LBS_MovingObjectManagement

8.5 Package – System management

8.5.1 Managing users and groups

Systems shall be able to track the identities of users, maintain their access information for licenses and keep response disposal instructions. Upon an authorized request the system shall be able to maintain a sequence of tracking locations for the use of applications as directed by the user. Tracking triggers will drive this sequence, and the system shall be responsible for maintaining the current and default set of triggers as specified by the user.

In third-party tracking applications such as fleet management and package tracking, the system shall be able to manage sets of tracked vehicles as a single unit, under a single user ID.

In addition to user management, the system shall be able to store, retrieve and update user-associated data, such as query and access information for resources, and personal information set by users, such as custom locations to be used in applications.

8.5.2 Type – LBS_UserProfileService

The type "LBS_UserProfileService" (Figure 30) is a data service supporting user profiles.

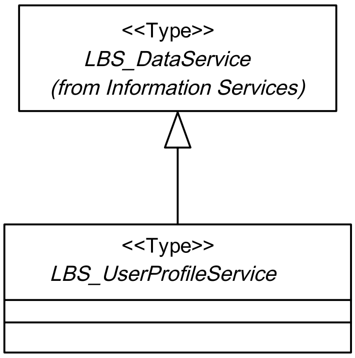


Figure 30 — Context Diagram: LBS_UserProfileService

8.5.3 Type – LBS_LocationTriggerControl

The type "LBS_LocationTriggerControl" (Figure 31) is a data service supporting triggering information.

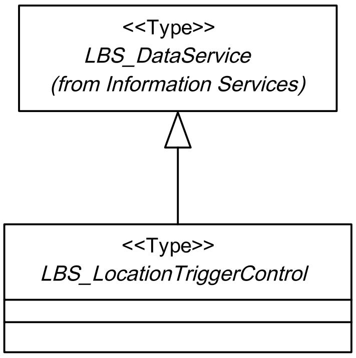


Figure 31 — Context Diagram: LBS_LocationTriggerControl

8.6 Package – Digital rights management

8.6.1 Digital rights management

The actions and activities of the participants constitute a commercial marketplace where the commodities being transferred are not concrete, but digital. This marketplace cannot survive if the rights of the owners of the data and service are not protected from misuse and unauthorised access. The management of such rights is the subject ISO/IEC TR 21000-1. The expression in digital form of such rights is covered in ISO/IEC 21000-5.

Every entity to which a right is associated is referred to as a **resource**. Essentially, the management of digital rights requires a flow of control data that act as licenses to the various clients of a resource. Use of a resource requires that the client of that resource present as part of his request a license (proof of a right) for the exercise of the activity being requested. With the exception of the user, all **participants** can be considered **resources** depending on the situation.

This package contains the types LBS_Resource, LBS_License, LBS_Right, and LBS_RightsCondition (Figure 32). Implementations to this International Standard shall also comply with ISO/IEC 21000.

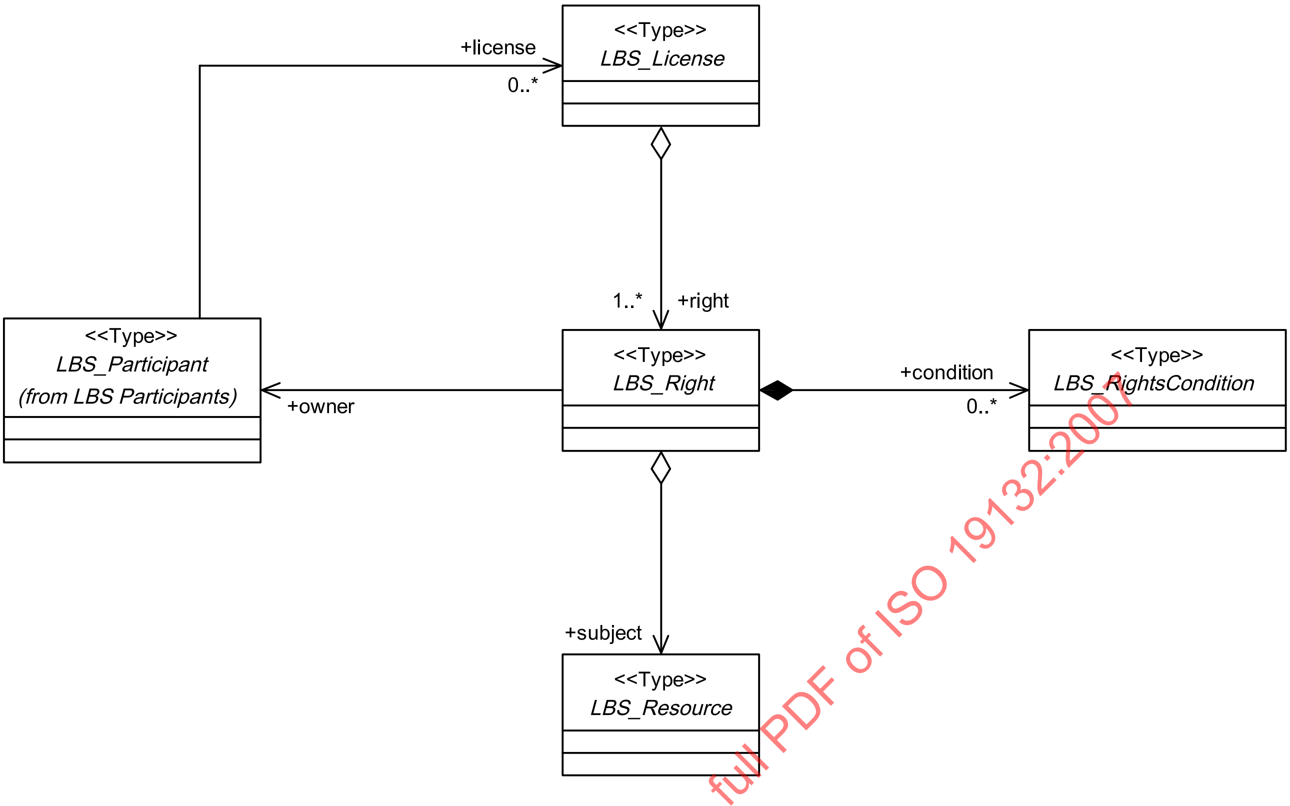


Figure 32 — Digital rights management types

8.6.2 Type – LBS_Resource

The type “LBS_Resource” (Figure 32) is an abstract root class for anything that can be accessed within the system, including services, brokers and data stores.

8.6.3 Type – LBS_License

8.6.3.1 Class semantics

The type “LBS_License” (Figure 32) is an abstract root class for anything that can contain a digital expression of the right to access a resource.

8.6.3.2 Association role – right

The association role “right” links the license to the collection of rights that it expresses.

```
LBS_License::right[1..*]: LBS_Right
```

8.6.4 Type – LBS_Right

8.6.4.1 Class semantics

The type “LBS_Right” (Figure 32) is an abstract root class for anything that can be done to or with a resource. The various types of rights depend upon the type of resource. A database, for example, would have rights to read, update, modify or delete data or metadata stored in the database (including application schemas). Each right can be assigned, with or without restrictions to users on the system.

8.6.4.2 Association role – subject

The association role "subject" associates the right to the resource that is involved.

```
LBS_Right::subject: LBS_Resource
```

8.6.4.3 Association role – condition

The association role "condition" associates the right to any legal conditions that may have to be met to validate the right. For example, rights to a resource may depend on the location of the user, or upon the time or date of the access to the resource.

```
LBS_Right::condition[0..*]: LBS_RightsCondition
```

8.6.4.4 Association role – owner

The association role "owner" associates the right to the owner of the right, who is some participant in the system, either a user or a broker, who wishes to access the resource.

```
LBS_Right::owner: LBS_Participant
```

8.6.5 Type – LBS_RightsCondition

The type "LBS_RightsCondition" (Figure 32) is an abstract root class for anything that can contain a digital expression of a conditional prerequisite to the use of the right to access a resource.

9 Message Data Model

9.1 Semantics

The messages between participants consist of requests and responses for the services described above. This part of the model deals with abstract data types that shall be used in those messages.

9.2 Package – Message Data Types

9.2.1 Package structure

The Message Data Type package (Figure 33) contains types for data used in operation invocation in LBS. These types include:

LBS_Position, LBS_Location, LBS_Accuracy,

LBS_Instruction, LBS_Maneuver, LBS_Route,

LBS_RouteCriteria, LBS_SecurityCertificate, LBS_TrackingLocation,

LBS_Trigger, LBS_UserID, LBS_AccessInfo,

LBS_GeometryChoice, LBS_NamedLocation, LBS_MapFormat,

LBS_SymbolSet, LBS_DisplayParameters, LBS_Data,

LBS_DataSource, LBS_TrackTrigger, LBS_FeatureData,

LBS_EventInfo, LBS_Notification, LBS_Address,

LBS_RouteConstraint, LBS_Preference, LBS_CostFunction,

LanguageSpecificCharacterString.

Figure 33 contains a UML overview of the package.

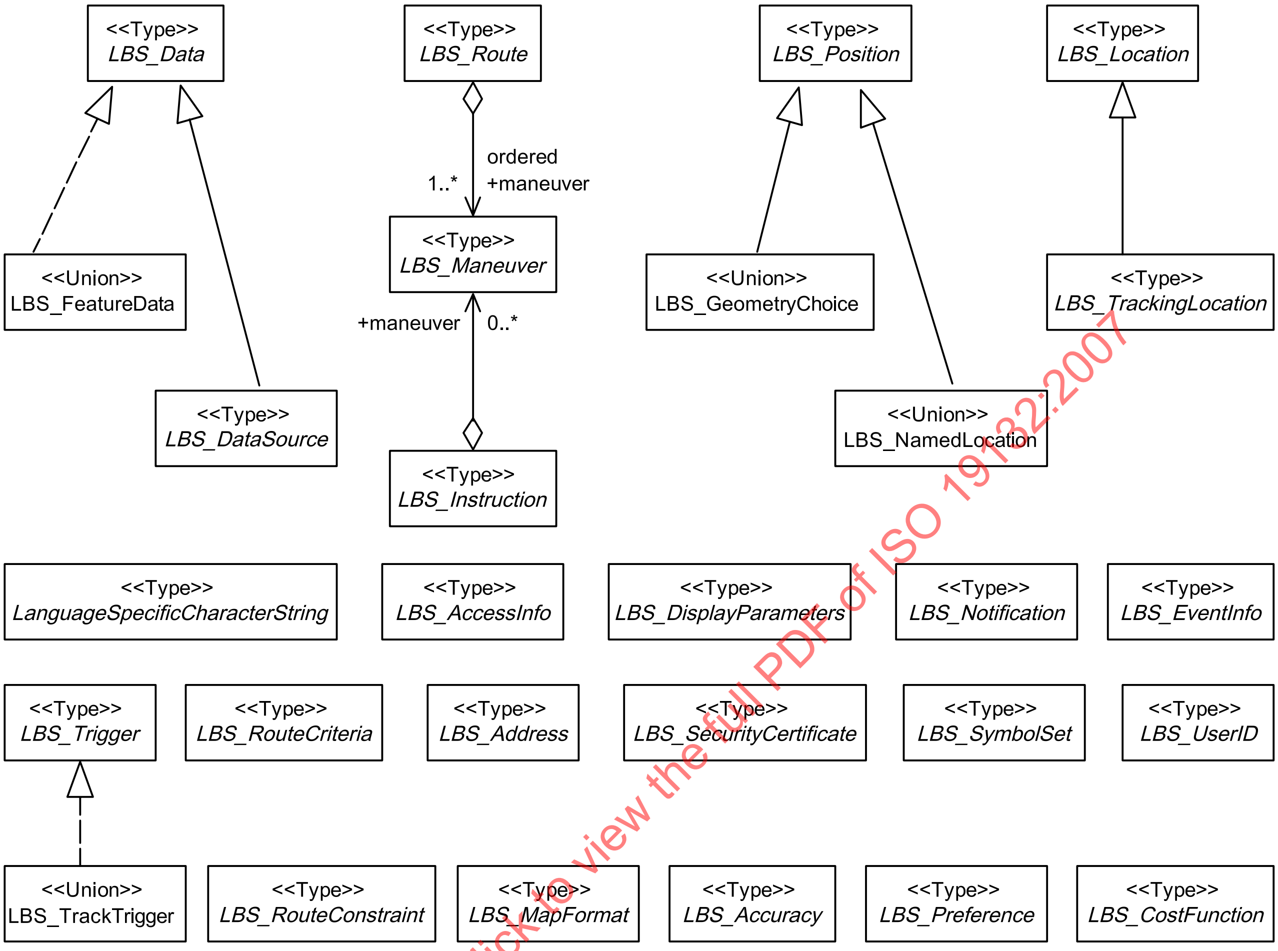


Figure 33 — Message data types

9.2.2 Type – LanguageSpecificCharacterString

9.2.2.1 Class semantics

Much of the message traffic will be carried as textual data (either notes or encoded data). It may be important to specify the languages of these strings for use by the recipient (multiple language strings may be transmitted in a message packet for use by localized versions of the applications. This extended type acts as an addition to the common data types specified in ISO/TS 19103 and, by type substitutability, it may be used in any place where a CharacterString type is appropriate.

9.2.2.2 Attribute – language

The LanguageSpecificCharacterString (Figure 34) carries an additional attribute indicating the ISO 639 language code for the included string.

```
LanguageSpecificCharacterString::language: LanguageCode
```

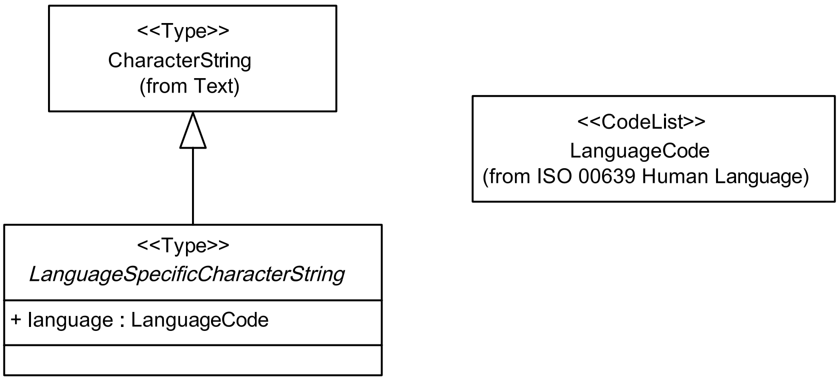

<<Type>>
CharacterString
(from Text)
<<CodeList>>
LanguageCode
(from ISO 00639 Human Language)
<<Type>>
LanguageSpecificCharacterString
+ language : LanguageCode

Figure 34 — Context diagram: LanguageSpecificCharacterString

9.2.3 Type – LBS_AccessInfo

9.2.3.1 Class semantics

The abstract type "LBS_AccessInfo" (Figure 35) holds information about the requestor of information, the target of that request, and sufficient security information to validate appropriate access, dependent on the service and server being used.

9.2.3.2 Attribute – id

The attribute "id" holds sufficient information for the service to be able to identify the ultimate user of the response and to distinguish him from other users, both for the purposes of validation of rights and for the retrieval of any stored preferences.

```
LBS_AccessInfo::id: LBS_UserID
```

9.2.3.3 Attribute – myID

The attribute "myID" identifies the user requesting a service. Most often, this will be the identity of the service broker who is acting as an intermediary between the user and the service.

```
LBS_AccessInfo::myID: LBS_UserID
```

9.2.3.4 Attribute – securityPass

The attribute "securityPass" shall contain proof that the requesting user "myID" has rights to access the sorts of information for the target "id" needed for the requested service. The mechanisms for the security certificate may contain encrypted information that could only be encrypted by the requestor for access to the target. Use of public key (PK), one-way, or temporally dependent encryption or other security measures are highly recommended.

```
LBS_AccessInfo::securityPass: LBS_SecurityCertificate
```

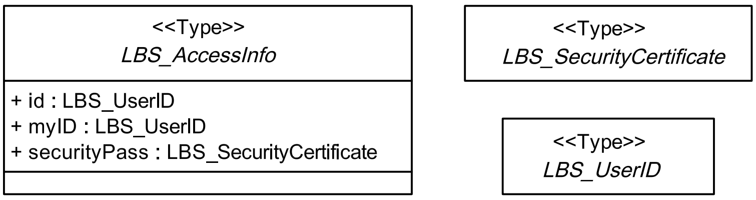
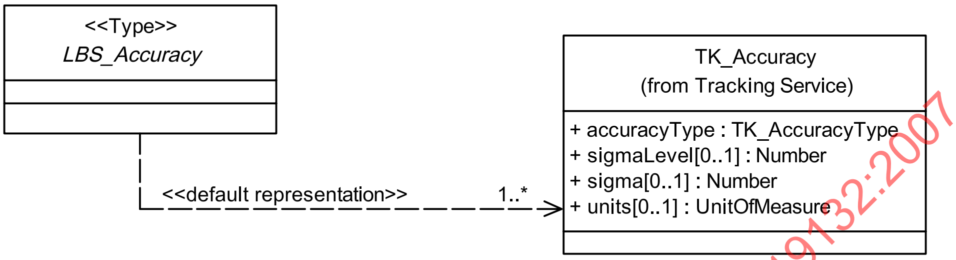


Figure 35 — Context diagram: LBS_AccessInfo

9.2.4 Type – LBS_Accuracy – Class semantics

"LBS_Accuracy" (Figure 36) is an abstract type, which can be realized by any class capable of describing the accuracy of a position representation. The accuracy representation TK_Accuracy from ISO 19133 shall be the default representation for accuracy implementations of LBS_Accuracy.

Figure 36 is a UML context diagram for LBS_Accuracy.

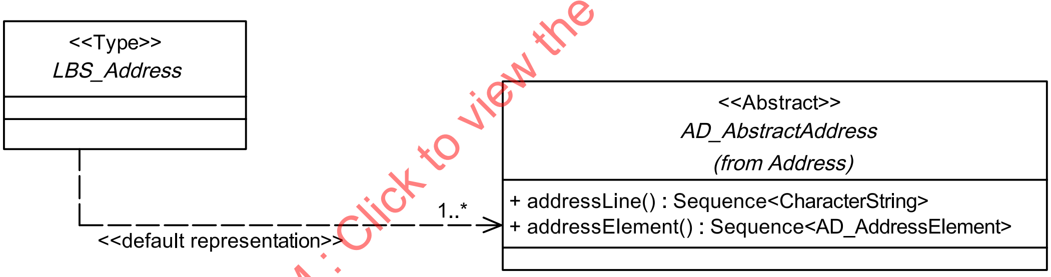


<<Type>>
LBS_Accuracy
<<default representation>>
1..*
TK_Accuracy
(from Tracking Service)
+ accuracyType : TK_AccuracyType
+ sigmaLevel[0..1] : Number
+ sigma[0..1] : Number
+ units[0..1] : UnitOfMeasure

Figure 36 — Context Diagram: LBS_Accuracy

9.2.5 Type – LBS_Address

The type "LBS_Address" (Figure 37) is an abstract class for any implementation class capable of carrying address information. Its default representations are the subtypes of AD_AbstractAddress from ISO 19133.



<<Type>>
LBS_Address
1..*
<<default representation>>
<<Abstract>>
AD_AbstractAddress
(from Address)
+ addressLine() : Sequence<CharacterString>
+ addressElement() : Sequence<AD_AddressElement>

Figure 37 — Context Diagram: LBS_Address

9.2.6 Type – LBS_CostFunction

The type "LBS_CostFunction" (Figure 38) is an abstract class for any implementation class capable of carrying information describing the cost functions used in selecting optimal routes in the navigation service. Its default representations are the subtypes of NS_CostFunction from ISO 19133.

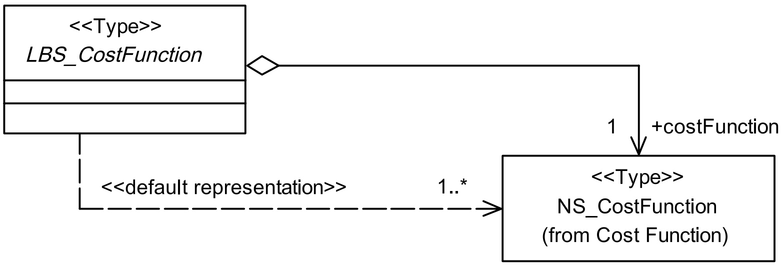


Figure 38 — Context Diagram: LBS_CostFunction

9.2.7 Type – LBS_Data

The type “LBS_ Data” (Figure 39) is an abstract class for any implementation class capable of carrying data in support of any of the services. Its format shall be GML from ISO 19136 or another standard format depending on the type of data being transmitted.

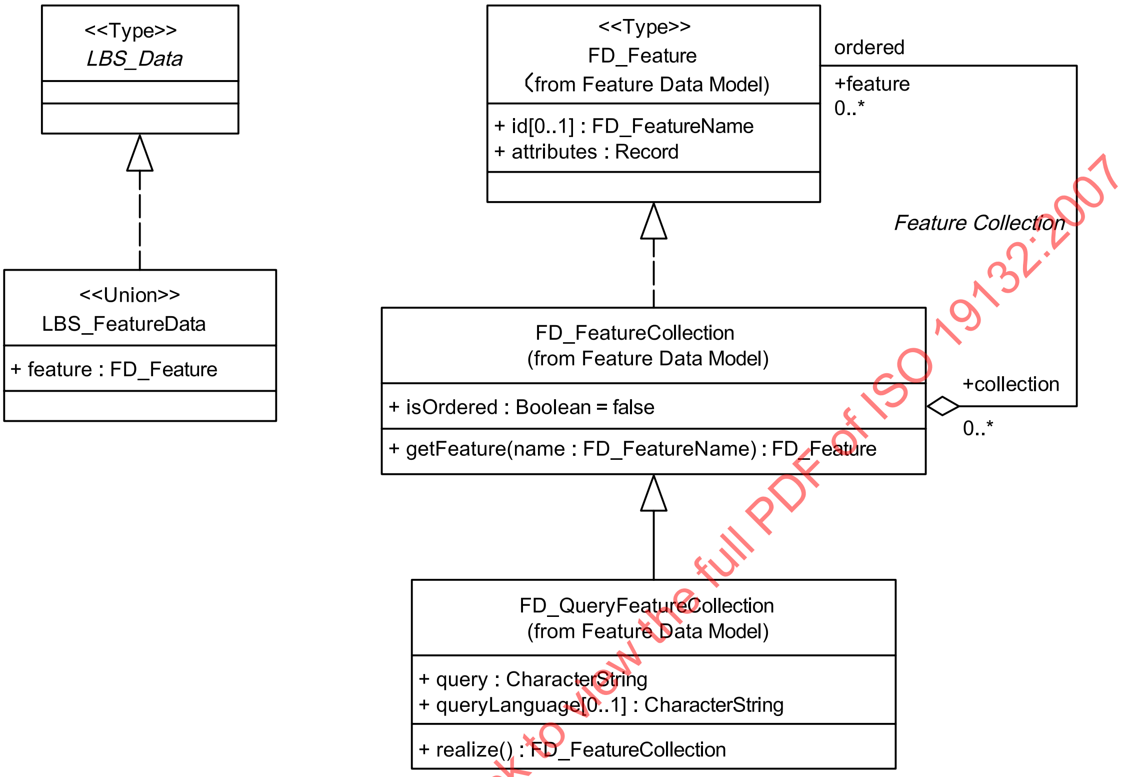

<<Type>>
LBS_Data
<<Union>>
LBS_FeatureData
+ feature : FD_Feature
<<Type>>
FD_Feature
(from Feature Data Model)
+ id[0..1] : FD_FeatureName
+ attributes : Record
ordered
+feature
0..*
Feature Collection
+collection
0..*
FD_FeatureCollection
(from Feature Data Model)
+ isOrdered : Boolean = false
+ getFeature(name : FD_FeatureName) : FD_Feature
FD_QueryFeatureCollection
(from Feature Data Model)
+ query : CharacterString
+ queryLanguage[0..1] : CharacterString
+ realize() : FD_FeatureCollection

Figure 39 — Context Diagram: LBS_Data

9.2.8 Type – LBS_DataSource

The type “LBS_DataSource” (Figure 40) is an abstract class for any resource capable of hosting data in support of any of the services. Its format shall be GML from ISO 19136 or another standard format depending on the type of data being transmitted.

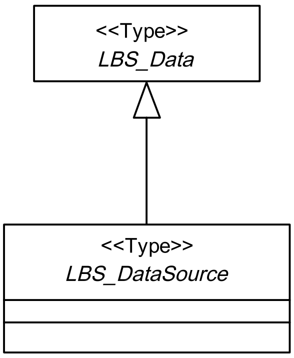


Figure 40 — Context Diagram: LBS_DataSource

9.2.9 Type – LBS_DisplayParameters

9.2.9.1 Class semantics

The display parameter class (Figure 41) carries information about the requested format of any map or other rendered representation of data.

9.2.9.2 Attribute – width

The “width” attribute indicates the horizontal width of the display. The unit of measure of the distance can be anything appropriate to the display (as opposed to the original data).

```
LBS_DisplayParameters::width: Distance
```

9.2.9.3 Attribute – length

The “length” attribute indicates the vertical length or height of the display. The unit of measure of the distance can be anything appropriate to the display (as opposed to the original data).

```
LBS_DisplayParameters::length: Distance
```

9.2.9.4 Attribute – scale

The “scale” attribute indicates the scale of the display. The scale is a number usually less than one which can map real world distances to display distances. Applications that require more than one scale will have to subtype the “Scale” data type to extend to the number of dimensions needed.

```
LBS_DisplayParameters::scale: Scale
```

9.2.9.5 Attribute – symbol

The “symbol” attribute holds a collection of symbols used in the display. This may be done using a reference to a known symbol set

```
LBS_DisplayParameters::symbol: LBS_SymbolSet
```

9.2.9.6 Attribute – format

The attribute “format” gives the data format for the map. The usual formats will include graphics and, potentially, feature data formats.

```
LBS_DisplayParameters::format: LBS_MapFormat
```

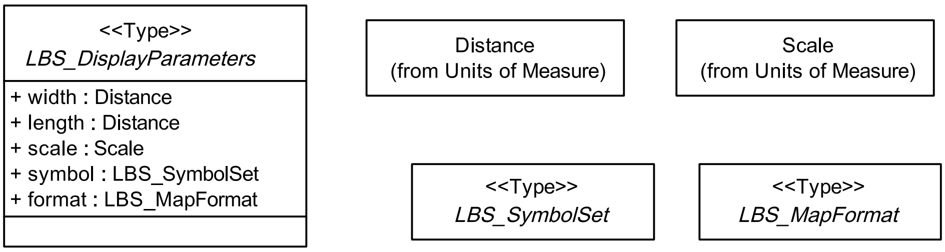


Figure 41 — Context Diagram: LBS_DisplayParameters

9.2.10 Type – LBS_EventInfo

9.2.10.1 Class semantics

The type “LBS_EventInfo” (Figure 42) is an abstract root type for classes that represent event information. The type of event will affect the contents of their associated event information type. The included attributes in this abstract type should be applicable in any situation.

9.2.10.2 Attribute – description

The “description” attribute should contain a summary description of the event. This attribute may be human readable, and may be language-specific. The language choice should correspond to the user preference for the target user.

```
LBS_EventInfo::description: CharacterString
```

9.2.10.3 Attribute – time

The “time” attribute shall contain the time at which the event occurred, or at which the event is planned to occur.

```
LBS_EventInfo::time[0..*]: TM_Primitive
```

9.2.10.4 Attribute – place

The “place” attribute shall contain the location at which the event occurred, or at which the event is planned to occur. Multiple places may be associated to multiple times in the time attribute. If this is the case, the order and number of these two attribute value lists should correspond.

```
LBS_EventInfo::place[0..*]: LBS_Location
```

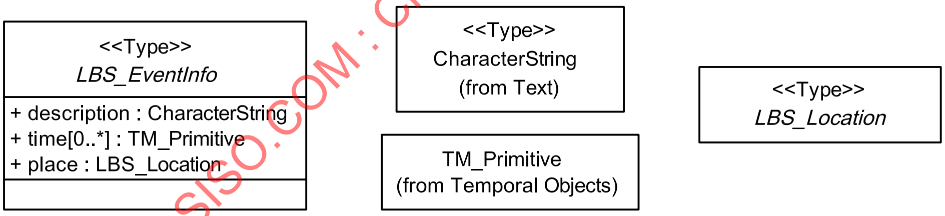


Figure 42 — Context Diagram: LBS_EventInfo

9.2.11 Type – LBS_Instruction

9.2.11.1 Class semantics

An “LBS_Instruction” is sufficient information for a traveller to execute a single step in a route. The content of realizations of this abstract type will depend on the type of navigation and the type of network or other geometry being navigated. The instruction representation NS_Instruction from ISO 19133 shall be the default representation for instruction implementations of LBS_Instruction. Figure 43 is a UML context diagram for LBS_Instruction. The default data types for an instruction are found in ISO 19133.

9.2.11.2 Association role – maneuver

The role “maneuver” gives the maneuver associated to this instruction.

```
LBS_Instruction::maneuver[0..*]: LBS_Maneuver
```

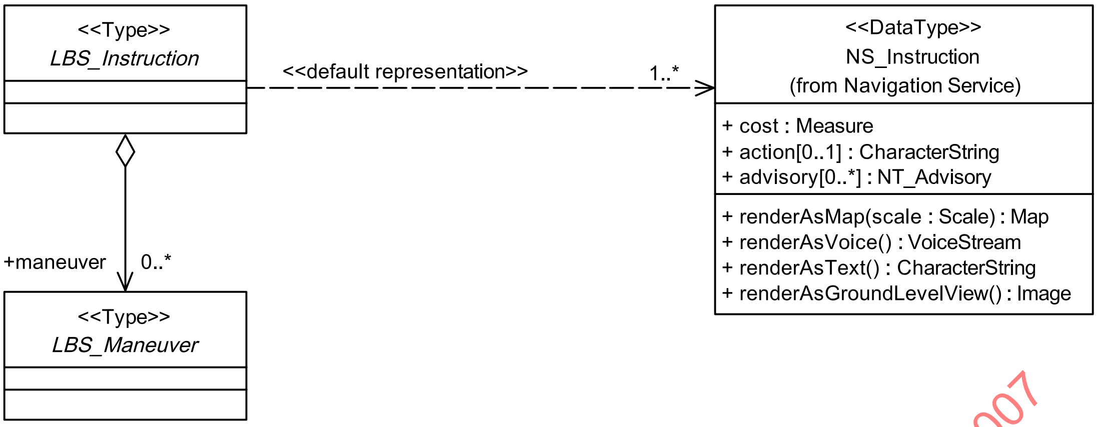

<<Type>>
LBS_Instruction
<<default representation>>
1..*
<<DataType>>
NS_Instruction
(from Navigation Service)
+ cost : Measure
+ action[0..1] : CharacterString
+ advisory[0..*] : NT_Advisory
+ renderAsMap(scale : Scale) : Map
+ renderAsVoice() : VoiceStream
+ renderAsText() : CharacterString
+ renderAsGroundLevelView() : Image
+maneuver
0..*
<<Type>>
LBS_Maneuver

Figure 43 — Context Diagram: LBS_Instruction

9.2.12 Type – LBS_Location

9.2.12.1 Class semantics

All instances of this type specify a location, and a degree of accuracy of that location. Figure 44 is a context diagram for LBS_Location.

9.2.12.2 Attribute – position

The position attribute describes the estimated position of the location.

```
LBS_Location::position: LBS_Position
```

9.2.12.3 Attribute – qualityOfPosition

The “qualityOfPosition” attribute describes the quality of the estimated position of the location.

```
LBS_Location::qualityOfPosition: LBS_Accuracy
```

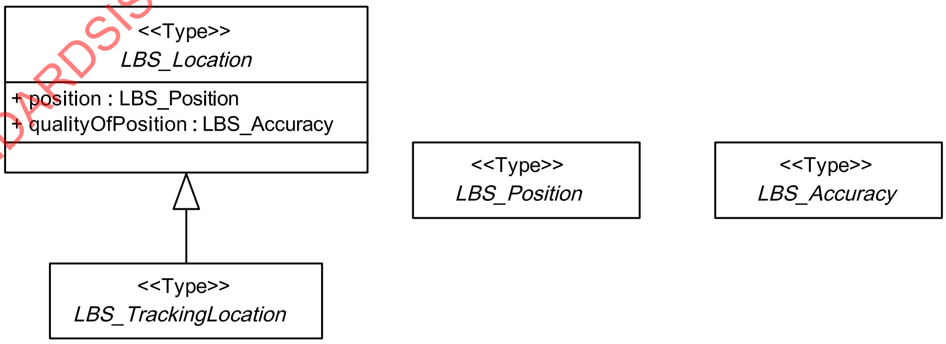


Figure 44 — Context Diagram: LBS_Location

9.2.13 Type – LBS_Maneuver

An implementation of the type LBS_Maneuver describes a means to execute all or part of a route. Its associated instruction (if it has one) details that maneuver in some manner. If the maneuver does not have an instruction, then it shall reflect the main-road rule for the route/network.

The default representation of maneuver is the type NT_Maneuver from ISO 19133.

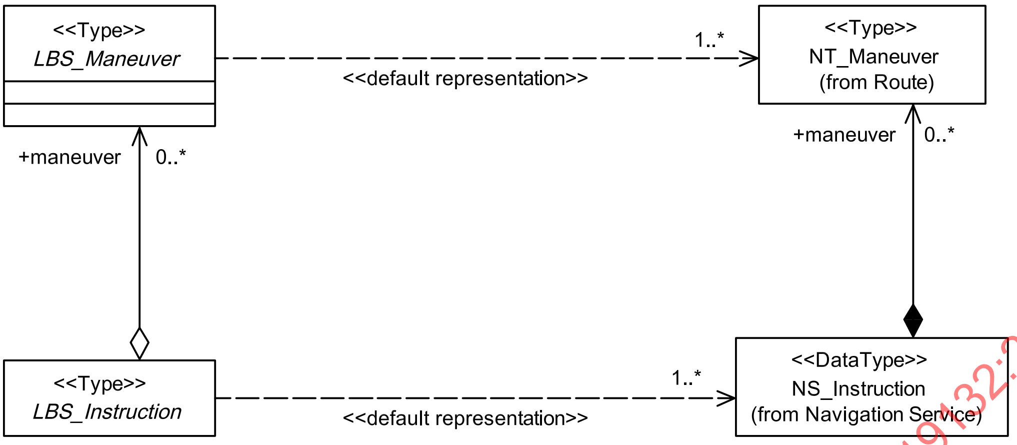


Figure 45 — Context Diagram: LBS_Maneuver

9.2.14 Type – LBS_MapFormat

The type "LBS_MapFormat" (Figure 46) is an abstract root type for any class that can be used to describe a data format for maps. This will be application-specific, see ISO 19117 and ISO 19128.

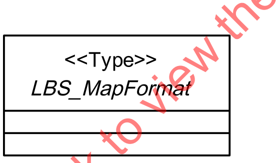


Figure 46 — Context Diagram: LBS_MapFormat

9.2.15 Type – LBS_Notification

9.2.15.1 Class semantics

An implementation of the type "LBS_Notification" (Figure 47) shall describe an event notification.

9.2.15.2 Attribute – user

The "user" attribute describes the target client of the event notification.

```
LBS_Notification::user: LBS_AccessInfo
```

9.2.15.3 Attribute – event

The "event" attribute describes the event, such as type, place and time.

```
LBS_Notification::event: LBS_EventInfo
```

9.2.15.4 Attribute – trigger

The "trigger" attribute describes the triggering condition that resulted in this notification being produced and delivered.

```
LBS_Notification::trigger: LBS_Trigger
```

9.2.15.5 Attribute – triggerTime

The “triggerTime” attribute describes the time at which the triggering condition was satisfied, causing this notification.

```
LBS_Notification::triggerTime: TM_Primitive
```

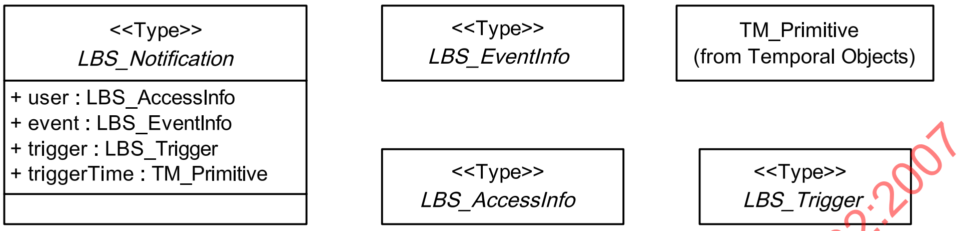


Figure 47 — Context Diagram: LBS_Notification

9.2.16 Type – LBS_Position

“LBS_Position” is an abstract type that can be realized by any class capable of describing a position. Figure 48 is a UML context diagram for LBS_Position.

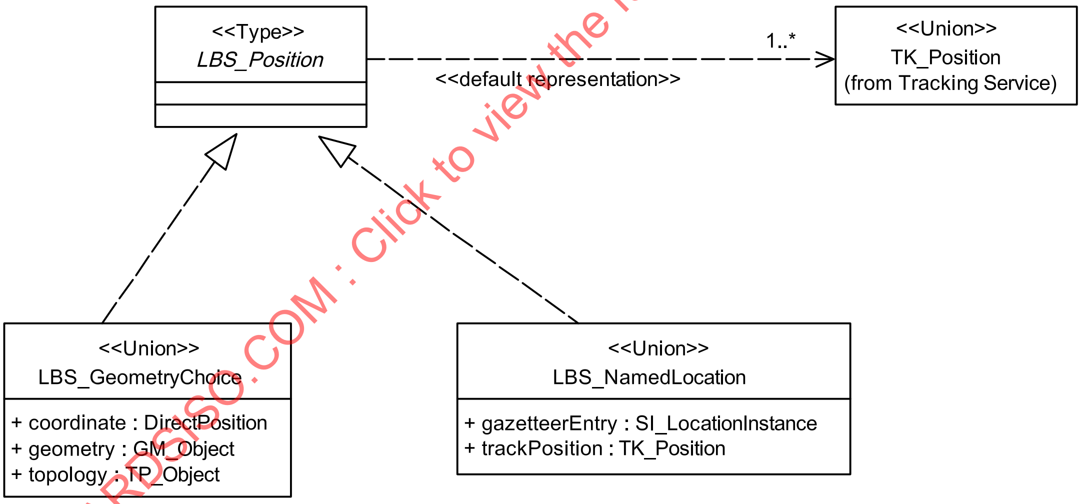


Figure 48 — Context Diagram: LBS_Position

9.2.17 Type – LBS_Preference

The type “LBS_Preference” (Figure 49) is an abstract type for all implementation classes that can express user preferences in LBS services. The default representation for LBS_Preference is the CodeList NS_RoutePreferences from ISO 19133.

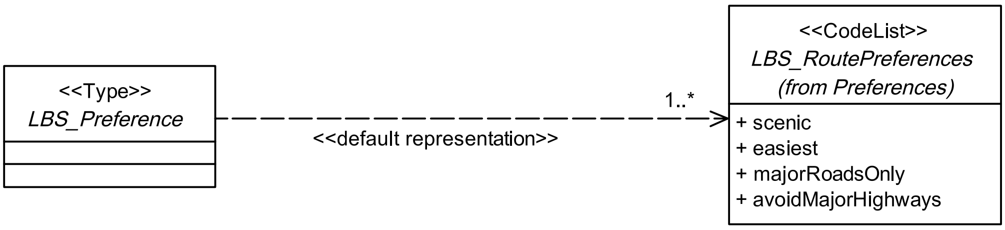


Figure 49 — Context Diagram: LBS_Preference

9.2.18 Type – LBS_Route

9.2.18.1 Class semantics

The class “LBS_Route” (Figure 50) is a sequence of maneuvers that will take a traveller from one location to the next as specified in a route request. The target of a route request may include specific locations, named objects or one location from a specific cluster of objects.

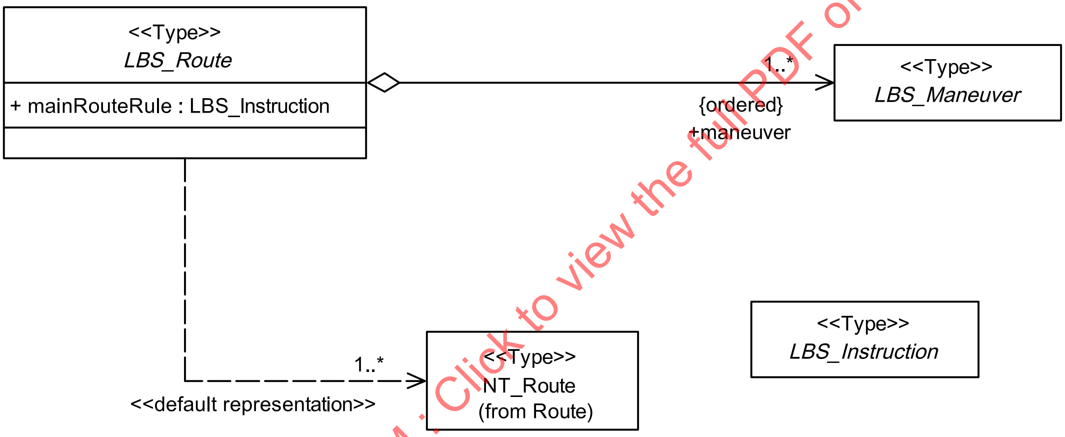


Figure 50 — Context Diagram: LBS_Route

9.2.18.2 Attribute – mainRouteRule

The “mainRouteRule” is the default instruction that is normally followed when executing a route in lieu of an instruction. For example, in highway navigation, the main route rule would indicate some means of determining which way is the equivalent of “follow current route.” It would normally be “as straight as possible” unless there is a routing sign indicating a turn. When the main route rule is violated by a maneuver, it must have an associated instruction.

```
LBS_Route::mainRouteRule: LBS_Instruction
```

9.2.18.3 Association Role – maneuver

The association role “maneuver” lists the maneuvers to be executed to follow this route.

```
LBS_Route::maneuver[1..*]: LBS_Maneuver
```

9.2.19 Type – LBS_RouteConstraint

The type “LBS_ RouteConstraint” (Figure 51) is an abstract type for all implementation classes that can express route constraint in LBS services. The default representation for LBS_RouteConstraint is the data type NT_Constraint from ISO 19133.

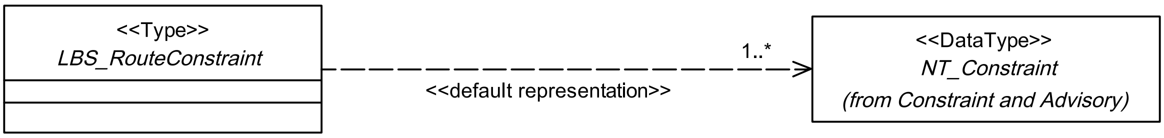


Figure 51 — Context Diagram: LBS_RouteConstraint

9.2.20 Type – LBS_RouteCriteria

9.2.20.1 Class semantics

The type “LBS_RouteCriteria” (Figure 52) is a partial description of a route that indicates which routes are acceptable and what is the cost to be optimized within a routing request. Constraint types are generally application-specific. For examples and default representations, see ISO 19133 and ISO 19134.

9.2.20.2 Attribute – start

The “start” of the route is the location from which the traveller plans to begin navigation.

```
LBS_RouteCriteria::start: LBS_Location
```

9.2.20.3 Attribute – wayPoints

The “wayPoints” of the route are the locations through which the traveller wishes the route to pass. The constraints may indicate whether the interior waypoints (not “start” or “end”) are to be navigated in order or not. The default is order free, since a restriction on order can be accomplished by requesting several routes at one time.

```
LBS_RouteCriteria::wayPoints[0..*]: LBS_Location
```

9.2.20.4 Attribute – end

The “end” of the route is the location at which the traveller plans to terminate this portion of the navigation.

```
LBS_RouteCriteria::end: LBS_Location
```

9.2.20.5 Attribute – constraints

The “constraints” indicate logical, legal or temporal limits to the type of route that is acceptable. For example, in vehicle navigation, the type of vehicle or its physical limitation puts constraints on the types of road segment usable in the route. In multimodal navigation, if the traveller is handicapped, then there is a constraint on access.

```
LBS_RouteCriteria::constraints[0..*]: LBS_RouteConstraint
```

9.2.20.6 Attribute – preferences

"Preferences" are weaker than constraints in that they are not logically necessary to the use of a route, but if they can be added to the constraints without incurring much cost (as set by the user preference), then the user would prefer the alternative.

```
LBS_RouteCriteria::preferences[0..*]: LBS_Preference
```

9.2.20.7 Attribute – costFunction

The "costFunction" indicates to the routing service what the user wishes to optimize. With the exception of the preferences, the route returned should be an optimal or near-optimal choice among all possibilities satisfying the constraints of the request, and passing from start to finish though the waypoints.

```
LBS_RouteCriteria::costFunction: LBS_CostFunction
```

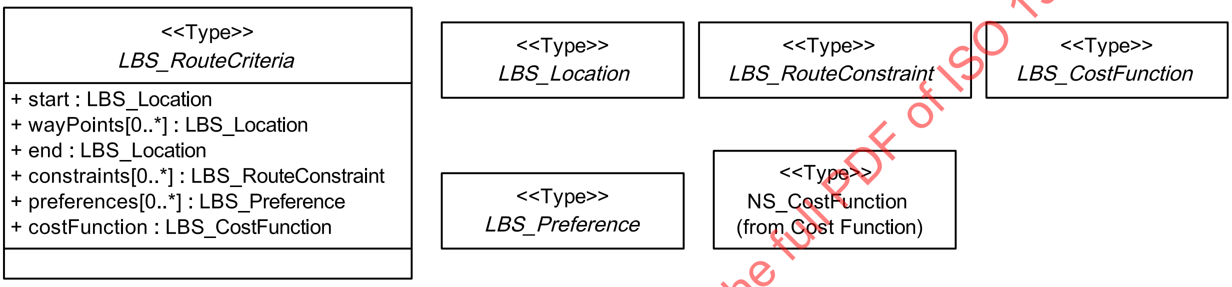


Figure 52 — Context Diagram: LBS_RouteCriteria

9.2.21 Type – LBS_SecurityCertificate

9.2.21.1 Class semantics

An "LBS_SecurityCertificate" (Figure 53) contains information that identifies the user whose data is being accessed, the user who is accessing that data (if different) and validation information that verifies that no unauthorized access is being made. The security model being used by the service provider determines the precise mechanism for this validation.

9.2.21.2 Association Role – user

The association role "user" contains the owner of the licenses contained in this security certificate.

```
LBS_SecurityCertificate:: user: LBS_UserID
```

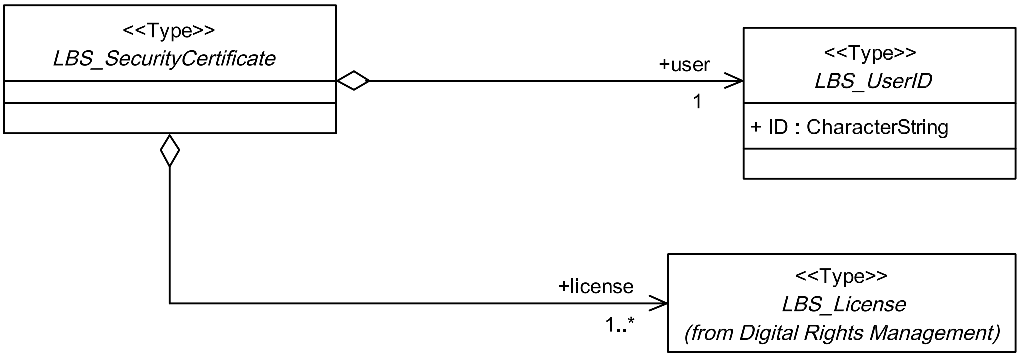


Figure 53 — Context Diagram: LBS_SecurityCertificate

9.2.21.3 Association Role – license

The association role "license" contains the licenses for this security certificate.

```
LBS_SecurityCertificate:: user[0..*]: LBS_License
```

9.2.22 Type – LBS_SymbolSet

The type "LBS_SymbolSet" (Figure 54) is an abstract root type for any mechanism for enumerating the symbols used on a particular type of map. See ISO 19117 and ISO 19128 for more information on map symbology and portrayal.

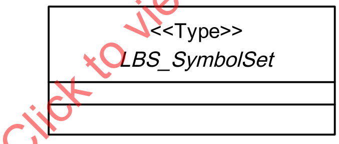


Figure 54 — Context Diagram: LBS_SymbolSet

9.2.23 Type – LBS_TrackingLocation

9.2.23.1 Class semantics

A realization of an "LBS_TrackingLocation" (Figure 55) describes the location of a tracked object at some point in time, and to some degree of accuracy, potentially as triggered by an event. Location may be aggregated into sequences to represent the results of the tracking request. These sequences are discrete temporal objects and represent records of discrete temporal change where each "timestamp" and "spacestamp" is contained in a tracking location.

9.2.23.2 Attribute – time

```
LBS_TrackingLocation::time[0..*]: TM_Primitive
```

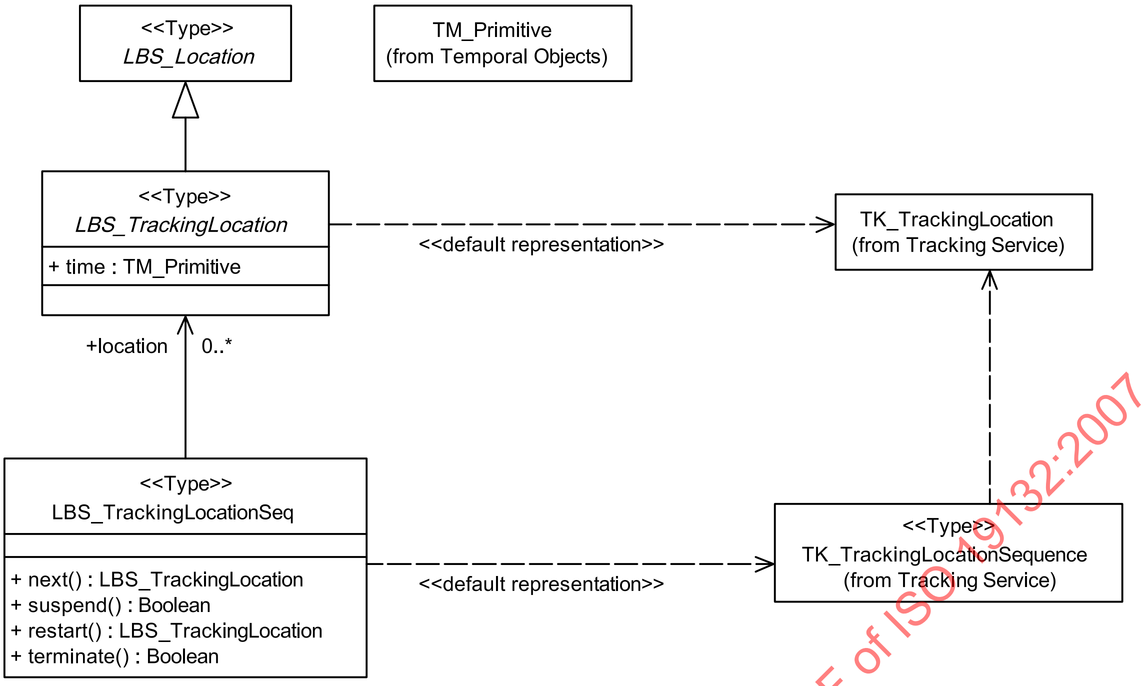


Figure 55 — Context Diagram: LBS_TrackingLocation

9.2.24 Type – LBS_Trigger

A realization of the abstract type "LBS_Trigger" (Figure 56) is a description of the type of event that will cause an action to be taken, usually a notification to a subscriber at the occurrence and location of the event. The default representation class for LBS_Trigger is TK_Trigger from ISO 19133.

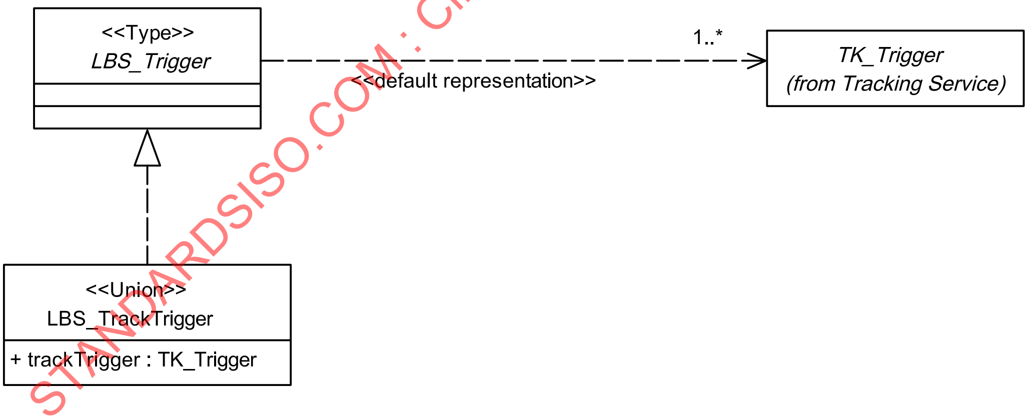


Figure 56 — Context Diagram: LBS_Trigger

9.2.25 Type – LBS_UserID

9.2.25.1 Class semantics

The type "LBS_UserID" (Figure 57) contains sufficient information to identify the user or trackable object in question. The content of the type is determined by the security model of the system and may vary among brokers and suppliers.

9.2.25.2 Attribute – ID

The attribute “ID” is the user ID expressed as a character string.

```
LBS_UserID::ID: CharacterString
```

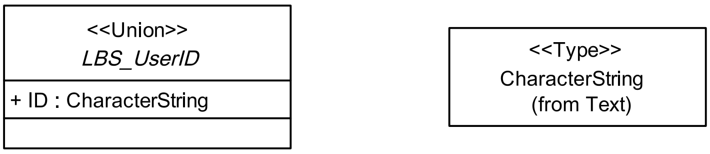


Figure 57 — Context Diagram: LBS_UserID

9.2.26 Union – LBS_FeatureData

9.2.26.1 Class semantics

The union class “LBS_FeatureData” (Figure 58) is an abstract root class for data representing feature class collections and containers. The default representation shall be from ISO 19136. Detailed UML for a conceptual feature model is given in ISO 19133, and detailed models and metamodels for feature data are given in ISO 19109 and ISO 19110.

9.2.26.2 Attribute – feature

The default choice of representation for LBS_FeatureData is the feature. Application may extend the choice to include other representations.

```
LBS_FeatureData::feature: FD_Feature
```

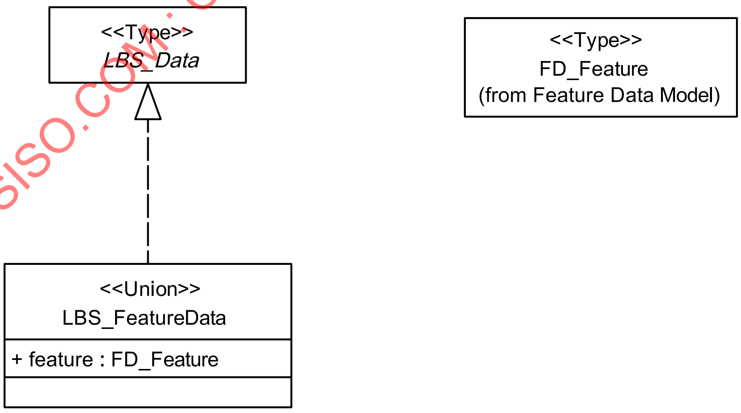


Figure 58 — Context Diagram: LBS_FeatureData

9.2.27 Union – LBS_GeometryChoice

9.2.27.1 Class semantics

The union class “LBS_GeometryChoice” (Figure 59) allows representation of location in a variety of geometric modes, either directly as coordinates, or as geometric or topological objects from ISO 19107 and related International Standards, such as ISO 19141, that define subtypes of the geometry or topology objects from ISO 19107.

9.2.27.2 Attribute – coordinate

The “coordinate” choice allows for locations expressed directly as coordinate tuples in any acceptable coordinate reference system. See ISO 19111 for details on coordinate systems.

```
LBS_GeometryChoice::coordinate: DirectPosition
```

9.2.27.3 Attribute – geometry

The “geometry” choice allows for locations expressed as geometry objects, as defined in ISO 19107.

```
LBS_GeometryChoice::geometry: GM_Object
```

9.2.27.4 Attribute – topology

The “topology” choice allows for locations expressed as topology objects, as defined in ISO 19107.

```
LBS_GeometryChoice::topology: TP_Object
```

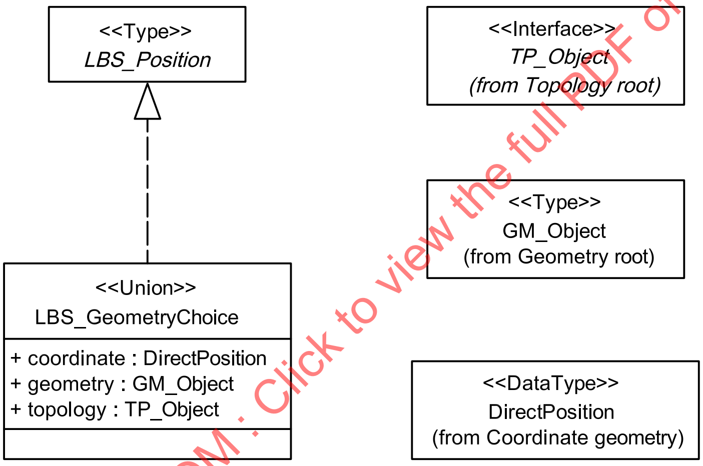


Figure 59 — Context Diagram: LBS_GeometryChoice

9.2.28 Union – LBS_NamedLocation

9.2.28.1 Class semantics

The union type “LBS_NamedLocation” (Figure 60) allows references to locations based on names, either gazetteer entries or the tracked position of a vehicle (equivalent to a user).

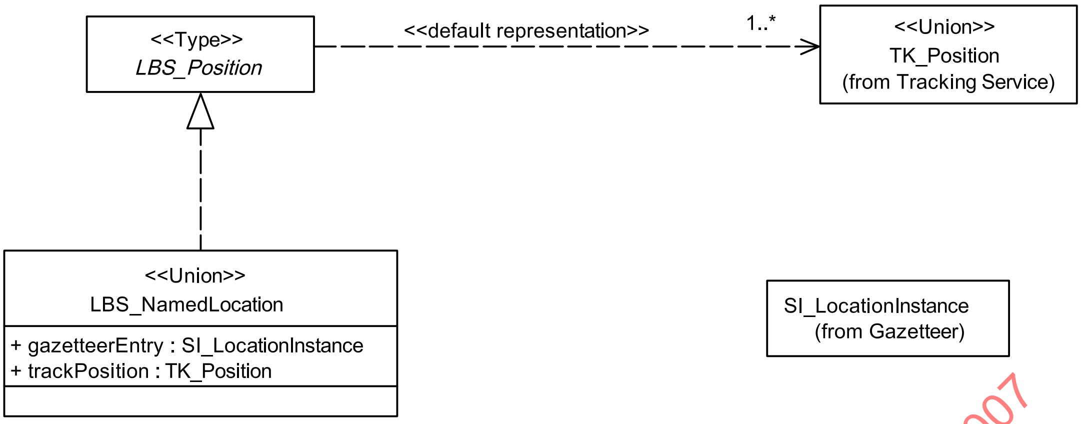


Figure 60 — Context Diagram: LBS_NamedLocation

9.2.28.2 Attribute – gazetteerEntry

The choice “gazetteerEntry” allows for use of named locations as described in ISO 19112.

```
LBS_NamedLocation::gazetteerEntry: SI_LocationInstance
```

9.2.28.3 Attribute – trackPosition

The choice “trackPosition” allows for use of tracking position as described in ISO 19133.

```
LBS_NamedLocation::trackPosition: TK_Position
```

9.2.29 Union – LBS_TrackTrigger

9.2.29.1 Class semantics

The union type “LBS_TrackTrigger” (Figure 61) act as a root class for defining the choice of trigger types for tracking. Applications may extend this union type by addition of other choices beyond the default TK_Trigger from ISO 19133.

9.2.29.2 Attribute – trackTrigger

The choice “trackTrigger” allows for the default representation of a trigger.

```
LBS_TrackTrigger::trackTrigger: TK_Trigger
```

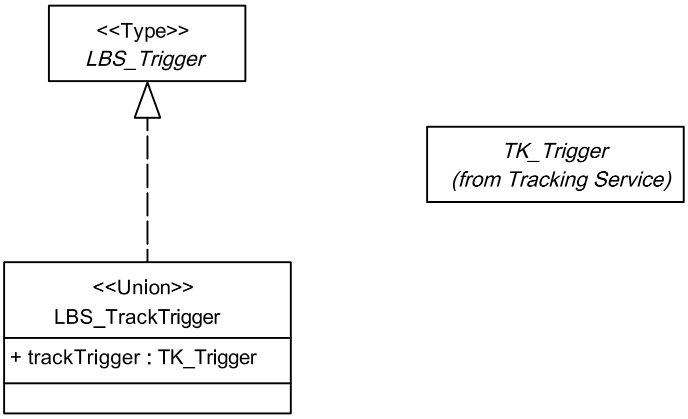


Figure 61 — Context Diagram: LBS_TrackTrigger

Annex A
(normative)

Abstract test suite

A.1 Semantics

Conformance by an implementation, implementation specification or other standard to this International Standard shall consist of **semantics conformance**, **data conformance**, **service conformance** or any consistent combination.

Semantics conformance shall imply that the terminology used by the candidate corresponds explicitly to this International Standard where possible.

Data conformance shall imply the usage of data types within application schemas or design specifications (“application type”) that are mappable into types in this International Standard (“standard type”). In this context, “mappable” shall mean that there is a correspondence between the standard types in the appropriate clause of this International Standard, and the application types of the application schema in such a way that each standard type can be considered as a supertype, realization or type specification of the application types designated by the correspondence. This means that an application type corresponding to a standard type contains sufficient data to recreate that standard type's information content and that the application type fully supports the standard type's behaviour as reflected in its operations. As in all realization relations, a single application type may realize multiple standard types and multiple application types may realize a single standard type.

Service conformance shall imply both the consistent use of message-based request-response interfaces and data conformance for the message packages used by those interfaces. The structure of those request-response pairs shall meet the data conformance criteria stated above.

A.2 Semantics conformance

A.2.1 Terminology

For a specification to be in **semantic conformance** to this International Standard, the terms in Clause 4 shall either be used directly where appropriate, or be associated to subordinate terms in the specification in question where appropriate.

a) Test Purpose: to verify an adequate semantics conformance. Each term used in the specification that has a corresponding term in this International Standard shall be explicitly linked to its corresponding term from Clause 4. The relationship between the terms shall be explained in sufficient detail as to prevent confusion when interpreting multiple specifications in semantics conformance.

b) Test Method: Inspect to ensure that the normative documents exhibit and explicitly state the required correspondence.

c) References: Clause 4.

d) Test Type: Capability.

A.2.2 Package – LBS Participants

For a specification to be in **semantic conformance** to this International Standard, the description of participants in the activities being standardized shall, where possible, be in terms of the participant model in this International Standard. This description need not be the primary model for the participants. The semantics defined in Clause 7 of this International Standard lay out basic semantics for the most common categories of participation.

a) Test Purpose: to verify an adequate participant model. Each participant shall be described in terms of extensions to the basic model presented in this International Standard, either through UML or in a corresponding language.

b) Test Method: Inspect to ensure that the documents exhibit and explicitly state the required correspondence.

c) References: the following specified subclauses from Clause 7:

1)	Type — LBS_Participant	7.2.2
2)	Type — LBS_User	7.2.3
3)	Type — LBS_ApplicationProvider	7.2.4
4)	Type — LBS_DataProvider	7.2.5
5)	Type — LBS_FeatureDataProvider	7.2.6
6)	Type — LBS_ContentProvider	7.2.7
7)	Type — LBS_SpatialContentProvider	7.2.8
8)	Type — LBS_ServiceProvider	7.2.9
9)	Type — LBS_ServiceBroker	7.2.10
10)	Type — LBS_MobileDevice	7.2.11
11)	Type — LBS_DataBroker	7.2.12

d) Test Type: Capability.

A.3 Service conformance

Service conformance of a specification or standard to this International Standard shall require the use of standard service types, where appropriate, in the activities being standardized. The semantics defined in Clause 8 of this International Standard lay out basic semantics for many of the common services.

a) Test Purpose: to verify use of basic services. When a service is, or requires the use of, one of the following services, the implementation of that service or service component shall be an instantiation of a subtype of the corresponding service defined in this International Standard.

b) Test Method: Inspect to ensure that the documents exhibit and explicitly state the required correspondence.

c) References: the following specified subclauses from Clause 8:

1)	Type — LBS_Tracking	8.2.2

2)	Type — LBS_Routing	8.2.3
3)	Type — LBS_Navigation	8.2.4
4)	Type — LBS_LocationTransformation	8.3.2
5)	Type — LBS_AddressParsing	8.3.3
6)	Type — LBS_Geoparsing	8.3.4
7)	Type — LBS_Gazetteer	8.3.5
8)	Type — LBS_MapService	8.3.6
9)	Type — LBS_DataService	8.4.2
10)	Type — LBS_NetworkDataService	8.4.3
11)	Type — LBS_EventSubscription	8.4.4
12)	Type — LBS_MovingObjectManagement	8.4.5
13)	Type — LBS_UserProfileService	8.5.2
14)	Type — LBS_LocationTriggerControl	8.5.3
15)	Type — LBS_Resource	8.6.2
16)	Type — LBS_License	8.6.3
17)	Type — LBS_Right	8.6.4
18)	Type — LBS_RightsCondition	8.6.5

d) Test Type: Capability.

A.4 Data conformance

Data conformance of a specification or standard to this International Standard shall require the use of standard data types, where appropriate, in the activities being standardized. The semantics defined in Clause 9 of this International Standard lay out basic semantics for many of the common services.

a) Test Purpose: to verify use of basic data types. When a data element is, or requires the use of, one of the following types, the implementation of that element or element component shall be an instantiation of a subtype of the corresponding type as defined in this International Standard.

b) Test Method: Inspect to ensure that the documents exhibit and explicitly state the required correspondence.

c) References: the following specified subclauses from Clause 9:

1)	Type — LanguageSpecificCharacterString	9.2.2
2)	Type — LBS_AccessInfo	9.2.3
3)	Type — LBS_Accuracy	9.2.4

4) Type — LBS_Address	9.2.5
5) Type — LBS_CostFunction	9.2.6
6) Type — LBS_Data	9.2.7
7) Type — LBS_DataSource	9.2.8
8) Type — LBS_DisplayParameters	9.2.9
9) Type — LBS_EventInfo	9.2.10
10) Type — LBS_Instruction	9.2.11
11) Type — LBS_Location	9.2.12
12) Type — LBS_Maneuver	9.2.13
13) Type — LBS_MapFormat	9.2.14
14) Type — LBS_Notification	9.2.15
15) Type — LBS_Position	9.2.16
16) Type — LBS_Preference	9.2.17
17) Type — LBS_Route	9.2.18
18) Type — LBS_RouteConstraint	9.2.19
19) Type — LBS_RouteCriteria	9.2.20
20) Type — LBS_SecurityCertificate	9.2.21
21) Type — LBS_SymbolSet	9.2.22
22) Type — LBS_TrackingLocation	9.2.23
23) Type — LBS_Trigger	9.2.24
24) Type — LBS_UserID	9.2.25
25) Union — LBS_FeatureData	9.2.26
26) Union — LBS_GeometryChoice	9.2.27
27) Union — LBS_NamedLocation	9.2.28
28) Union — LBS_TrackTrigger	9.2.29

d) Test Type: Capability.

Annex B
(informative)

Architecture

B.1 LBS Model

This is a short description of the framework architecture, the data and services needed to support location-based services.

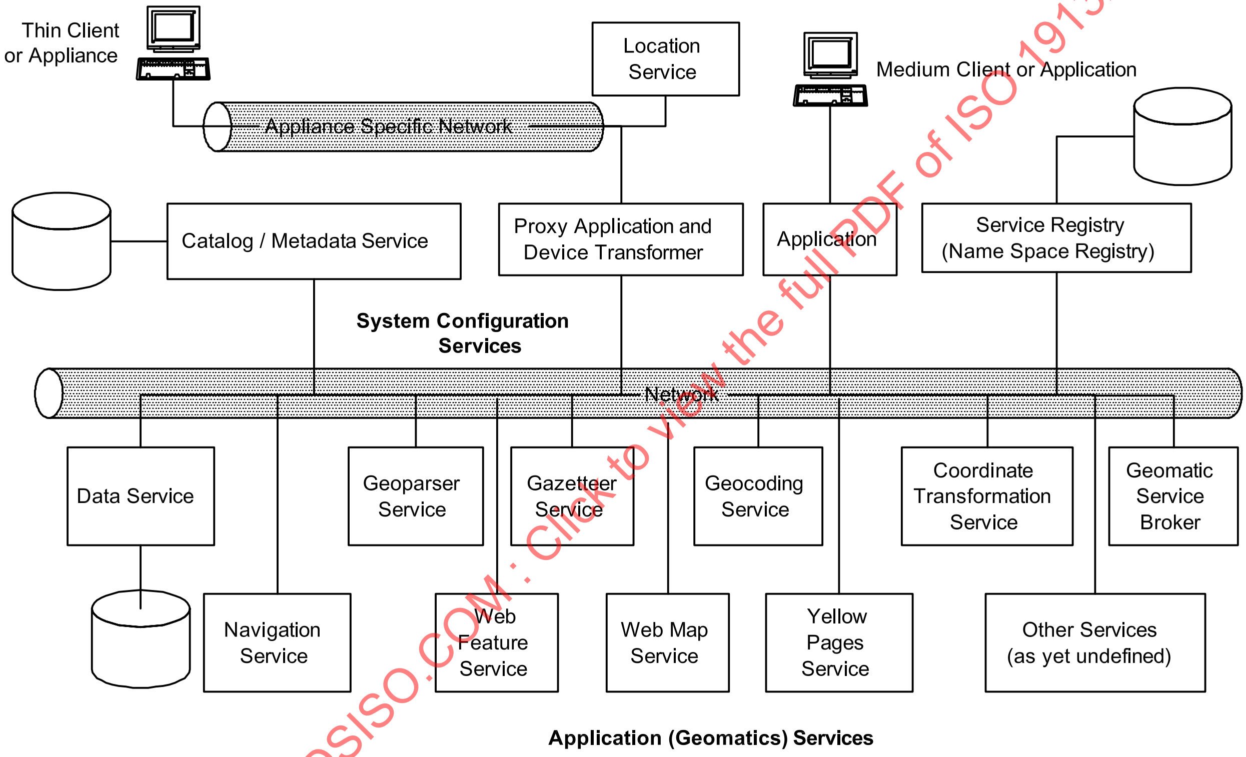

Thin Client
or Appliance
Location Service
Medium Client or Application
Appliance Specific Network
Catalog / Metadata Service
Proxy Application and Device Transformer
Application
Service Registry (Name Space Registry)
System Configuration Services
Network
Data Service
Geoparser Service
Gazetteer Service
Geocoding Service
Coordinate Transformation Service
Geomatic Service Broker
Navigation Service
Web Feature Service
Web Map Service
Yellow Pages Service
Other Services (as yet undefined)
Application (Geomatics) Services

Figure B.1 — Conceptual architecture equating mobile and non-mobile services

The conceptual architecture is shown in Figure B.1. The basic assumption is that services made available on the web will be accessed by mobile devices in a manner similar to on-web clients. On-web proxy applications for the mobile client are required to make this possible by acting as a device transformer for messages and data flowing between the service and the mobile client. The interface between the mobile client and the on-web proxy is outside the scope of this International Standard and is covered by standards written by and within OGC:OLS Transport. Mobile devices may include notebook computers, and their connection to the network can take on any form.

The second assumption is that the state of the mobile client will be maintained by the client application or by its on-web proxy application. This means that all requests for services shall be totally encapsulated in a request-response pair. Any state information for a client or service may be stored either at the client, at the proxy application or at the server, with data for identification of state variables being part of the service interface, if needed.

B.2 LBS Interface Schema

This clause explains in more detail some of the components from the conceptual architecture shown above. Figure B.2 shows the interface schema linking users to the data that they require. The items boxed with a dashed line indicate items identified as requiring standardization.

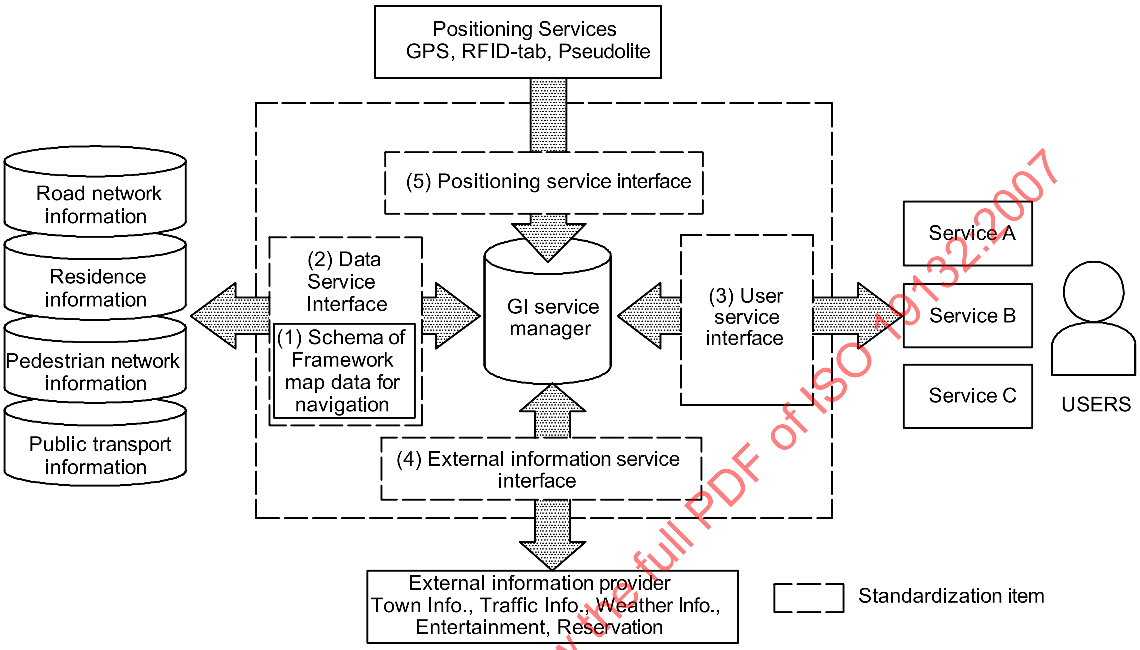

Positioning Services
GPS, RFID-tab, Pseudolite
(5) Positioning service interface
Road network information
Residence information
Pedestrian network information
Public transport information
(2) Data Service Interface
(1) Schema of Framework map data for navigation
GI service manager
(3) User service interface
Service A
Service B
Service C
USERS
(4) External information service interface
External information provider
Town Info., Traffic Info., Weather Info., Entertainment, Reservation
Standardization item

Figure B.2 — LBS interface schema and tentative standardization items

The GI service manager is responsible for linking the three elementary components of a location-based service (Table B.1). These are the map data service interface, incorporating schema or framework map data for navigation [(2) and (1) respectively], the positioning service interface (5), and the external information service interface (4). The user service interface (3) is the interface for the GI service manager and interacts with user requests via services. This interface receives user queries and requests and passes them to the GI service manager, which distributes the requests to the appropriate elementary service. The contents of this service interface are described in this International Standard.

Table B.1 — Elementary components of LBS

Key	Standardization Items	Description
(1)	Schema or Framework Map Data	Logical schema for datastore supporting map data service interface
(2)	Map data service interface	GML, G-XML, SVG
(3)	User service interface	POI service Positioning service Navigation, tracking service Push service
(4)	External information service interface	Traffic information Public transportation information
(5)	Positioning service interface (GNSS, mobile phone, wireless LAN, POI)	Positioning data (X, Y, Z coordinates and accuracy) Location referencing

External information services include other web services, such as town information, entertainment guides and reservation bookings. It is assumed that external information services can be linked through Point of Interest (POI) information.

In addition to these service interfaces, the schema of framework map data for personal navigation services should be standardized. The framework map data should include multimodal network data including roads, sidewalks, and public transport. It is envisaged that the common sharing of the feature catalogue and the multimodal data will be beneficial to navigation service companies.

Annex C
(informative)

Scenarios

C.1 Semantics

In order to appreciate the diversity and common framework elements of location-based services, a sample set of scenarios has been compiled. Location-based services can be broadly classed into either push or pull services – push services provide information based on a user's position or proximity to a target, while pull services require the user to formally make a request for information initiating the use of his/her location as an information filter. The scenarios described in the following clauses often propose a combination of pull and push services.

Each of the scenarios requires elements of the proposed LBS framework. Some scenarios may use one framework component, whilst others may use several. The range of services and their associated varied reliance on the framework elements highlight the need for interoperability and a cohesive set of standards in order to provide useful location-based services.

C.2 Disaster management

Disasters are unforeseeable events that happen in many different ways and take many different forms. The term disaster can include incidents such as earthquake, flood, terrorist attack, tornado, cyclone, nuclear explosion, forest fire, toxic contamination of the environment such as a nuclear plant leak or a train derailment involving toxic product containers. When faced with a disaster situation, quick and efficient reaction times are essential to save lives and minimize propagation of the disaster.

To be useful to emergency services teams, location-based services will have to adapt to a very wide range of cases. They will have to allow all the people working around a problem to communicate to each other critical information, such as

a) which regions are affected,

b) where the restricted areas are,

c) what the facilities are that are still useful and to what extent they are being used,

d) how many injured people there are in a specific area,

e) what kind of emergency workers are needed and a priority order for target areas.

In short, a large quantity of diverse information will have to be shared.

Yet, some applications have been developed using early implementations of Web Map Server and Web Feature Server. These applications can combine information from different sources such as forecast maps from the meteorological services, current maps from the hydrographic services, statistical dispersion simulations, maps from the municipalities or federal agencies, information on the strategic buildings and their capacities. The next step is to allow them to handle new information (push operations) like GNSS traces for marked-off areas or high priority zones, numbers of people in hospitals and in refuges like schools or churches, positions and kinds of emergency teams in the field. When this is done, the information will be available in real time to the emergency workers, the rescue teams and the control centre, thus eliminating delays that would otherwise occur.

C.3 Mountain rescue

Location-based services can provide a means for locating and rescuing people hiking or travelling in mountainous regions who may have lost their way, been injured or who suffer from sudden health problems. In a wider sense, LBS as a navigation tool and geographic information broker can also prevent such situations. This includes people being guided along safe hiking paths, being guided to mountain hotels and to other important points of interest, and being warned about dangerous precipices, etc.

In the strict sense, however, LBS for mountain rescue will be used after such an incident has occurred. The first task would be to locate the persons that need help. The nature of mountainous, wooded areas may mean that positioning cannot be achieved using satellite-positioning techniques alone. In this case, locating can be seen as a special case of tracking since the “history” of the hiker having walked along a certain path before the incident occurred could form important information. Additionally, pre-trip plans of intended routes could be valuable. The second task would be to locate all facilities (medical, transport, infrastructure) that are sufficiently near to be of help. A third task would lead to a survey of routes to reach the person and subsequently to reach the help centres. This needs full GIS support with spatio-temporal-semantic information. It also expands the usual scope of tracking along given routes, since digital spatial information about mountain paths will usually not be as readily available as it is for road networks.

In an extended sense, tracking can go beyond the scope of location and time but can also include tracking of vital medical characteristics of the person (for example monitoring blood pressure, etc.). LBS can in that way also be seen as an online monitoring system for people at risk in mountainous areas, whether this risk stems from medical, geographical or meteorological reasons or a combination thereof.

C.4 Intelligent routing

A traveller would like to go from some origin (not necessarily where they currently are) to some destination, given certain constraints and/or requirements, which could be fuzzily or well defined, static or dynamic. The traveller could be a person (with or without transport) with the ability to make decisions, or an object being sent through a system (such as a parcel or freight) without any intelligence, but which either people monitor or is done automatically.

Issues to consider include the following.

- Time dependence – how much latitude does the traveller have when they start at their origin, when they arrive at their destination, or when they visit any waypoints (a check point to be passed through, but where the traveller does nothing) or via-points (a point to be visited, where the traveller does something, though they don't necessarily stop there)?

- Static data – route data that changes infrequently, such as the gradient and surface of roads; the presence or absence of bridges (vs. fords); airline, train, bus and ferry schedules; etc.

- Dynamic data – route data that changes frequently, such as roadworks, traffic collisions, traffic jams and other traffic data.

- Events – special events that could impact on the possible routes the traveller could use, such as large sports events, public demonstrations, festivals, parades, etc.

- Surrogate or proxy data – statistical assessments of dynamic data that provide surrogates or proxies for dynamic data or events, such as rush-hour traffic.

- Waypoints and via-points – points to be passed through or visited, which could be well defined (e.g. a specific address to make a delivery), generic (e.g. any petrol station to fill up the traveller's car) or abstract (e.g. a point selected by the traveller to bias the resultant route, possibly as a proxy for avoiding things on routes that were more likely to be selected).

- Restrictions – points, lines or areas that the traveller would like to avoid, such as dangerous intersections, pot-holed roads or high crime areas.

- Route requirements – desirable characteristics from the traveller's point of view, such as scenic routes, near-impassable routes (for recreational users of four wheel drive vehicles), routes through historic districts, routes with low traffic, etc.

- Alternative routes – routes to take if the "optimal" route loses its advantage or if the traveller gets "lost" (strays off the route), which could involve returning the traveller to the original route, or generating a new route.

- Monitoring of progress – the rate at which the traveller's progress will need to be monitored will vary, depending on various factors (traveller's speed, traveller's track record of staying on their route, etc.); there will also be technical limitations on which the traveller can be tracked (e.g. GNSS does not work in a tunnel).

- Providing feedback – similarly, the rate at which feedback of the traveller's progress has to be provided will also vary.

Described in more detail in the following two clauses are example scenarios that fall within the intelligent routing domain of location-based services.

C.5 Personal navigation for the visually impaired

A specific scenario for an application of location-based services that expands upon the intelligent routing scenario is the case of a visually impaired or elderly traveller using individual and/or public transport. Such applications typically propose to efficiently increase the mobility of visually impaired pedestrians, particularly in an unknown urban environment. To achieve this aim, a combination of pre-journey planning and journey positioning and routing is required. Travel planning as well as navigation has to meet the special needs of visually impaired people in choosing an optimal route, being guided along the route and being informed about objects in the vicinity while en route.

As an essential component of the system, visual sensors and the use of a digital 3D city model can fill the gap in the disabled person's perceptual capabilities by supporting conventional satellite-based and terrestrial positioning sensors, by identifying large obstacles on the sidewalk, and by recognizing important warning and information signs.

Use of RFID tags mounted near pathways can further enhance the traveller's experience by providing up-to-the-minute local information available to tag readers associated to the traveller's tracking device's sensors.

C.6 Public transport

Also expanding on the intelligent routing scenario is the specific case of a traveller who would like to travel on the public transport system of a particular city. Metropolitan areas pose many options for travellers, including automobiles (private or taxis), public transport (including bus and rail systems) and walking. Often a solution mode (or combination of modes) is selected to minimize travel time or travel costs.

Location-based services can assist travellers in solving their routing problems by providing them with real-time traffic conditions, real-time public transport vehicle locations, and pedestrian navigation instructions.

Considering the case of a traveller who would like to use public transport for his/her journey, the traveller may require an LBS to

- determine an appropriate mode and/or route for the journey,

- restrict the journey to include or exclude a particular mode and/or route of transport that s/he wishes to use or not use,

- assist her/him in planning a trip, or investigate trip alternatives prior to travelling, or
- provide timetable look-up information for a nearby stop or station.

In all cases, travellers require route information including stop/station identifiers, mode, vehicle facilities (e.g. low floor facilities for easy wheelchair/pram access, audible stop announcements, number of carriages, etc.), departure and arrival times, route confirmation (e.g. stop countdowns until disembarkation), navigation information (if the journey origin/destination does not correlate with a public transport embarkation/disembarkation point). Ideally the detail and form of this information should vary depending on the user and the platform on which they access the service. Travellers accessing the service via mobile phone may be limited to text- or voice-based information, whereas those accessing via personal digital assistants or similar could be provided with full colour maps of their journey.

As for the scenario above, it can also be classified into the cases of pre-trip planning and trip execution. In the case of pre-trip planning, static timetable information can be supplemented with real-time information to improve the accuracy of predicted arrival and departure times of public transport vehicles. A form of tracking the user along his/her journey could also prove useful by providing prompts notifying users that they should disembark at the next stop, or by allowing users to plan alternative routes if services are cancelled or disrupted during their journey.

Additionally in some areas, Demand Responsive Transport Services also exist as an alternative form of public transport. Vehicles on these services typically do not operate on a fixed route or on a fixed schedule, but rather in response to patron requests, and hence patrons with differing routes can be conveyed with the same vehicle at the same time. In this context, vehicle tracking and current patronage attributes are of high importance and should also be incorporated into LBS.

C.7 Tracking

Following the location of an object or person as it moves over a period of time is central to location-based services. Tracking can be triggered by some event or by the passage of time, depending on the scenario. For example, a natural disaster event could trigger tracking of the disaster and its modelling, so that people in areas likely to be affected could be warned. On a less dramatic scale, the tracking of public transport vehicles and passengers could be used to help passengers choose alternative routes if vehicles are delayed.

The form of tracking position would vary depending on the object or person being tracked, and could be a position or coordinate in a particular reference system, address, or affected area/polygon extent.

C.8 MicroLBS for home or business use

Through using RFID tags or similar devices, a homeowner can tag all moveable items in his household inventory. By incorporating tag readers at all doorways, the house computer system can track the room location of each item, and use that to keep track of important household or personal items.

Similarly, a business can tag all its moveable items and its inventory. This can be integrated into a tracking and retail system that creates invoices for sales by scanning items, and can provide easy inventory and stocking processes based on tracking tagged items from storage to shelf or other location.

C.9 Road maintenance and Intelligent Transportation Systems (ITS)

C.9.1 Overview

Intelligent Transportation Systems (ITS) are usually part of a larger road administration which also includes maintenance, operations, planning and design. Recently LBS has worked its way into these areas, and is expected to play an ever more important role in the years to come, for example in the following areas:

- **Road maintenance and operation.** LBS functions will be used to identify and manage roadside equipment through electronic tagging, i.e. RFID. The tags will hold important information such as maintenance status, ownership and location. This technology will improve road maintenance and operation through better data management.

- **Traffic management.** LBS functions will be used to provide information directly to the driver about the road and traffic conditions at any location on the road network.

- **eSafety.** LBS functions are used to support in-car accident prevention systems such as ADAS (Adopted Driver Assistant System), and render possible the concept of intelligent roads.

- **ITS.** LBS functions are used to supply road users with instructions and restrictions on how to use the road network in the most efficient manner. For example, RFID equipment or similar (see CALM) posted along the roadway can carry information about road and driving conditions. This information can be beamed to the vehicle as it passes.

There is no distinction between off- and on-board systems. Both will exist and can be used simultaneously. It is generally assumed that the vehicle knows its location through a Linear Referencing System and/or GNSS combined with an off- or on-board geographic database. RFID equipment along the road can thus be used as fixed location points and at the same time give access to helpful information according to its unique ID.

C.9.2 ITS

ITS car navigation systems are dependent on road instructions and restrictions. These instructions and restrictions are basically static. Temporal changes due to planned or immediate roadworks can be beamed to the navigation system from roadside RFID placed there by road workers. The navigation systems respond by warning the driver, recalculating the route and giving the driver appropriate advice.

Correction or updates of the navigation dataset might also be transmitted via roadside equipment to the vehicle while it is passing by. In the same manner, services can be established that transmit information about tunnels, bridges and POI as they are passed *en route*.

Services that involve car-to-car communication, e.g. a car further down the road, could update the navigation system in cars behind on congestion, accidents or weather conditions.

C.9.3 ITS and eSafety

eSafety denotes a range of equipment and systems that will help the driver prevent accidents based on known safety attributes relevant to the road ahead. Suggested attributes might be the following:

- speed limits;
- traffic signs and lights;
- curvature/slope gradients;
- pedestrian crossings;
- road surface conditions;
- volume of cross-traffic;
- complex traffic situations due to schools, kindergartens and malls.

These attributes are subject to change. LBS might apply means to transport these attributes or updates from national road databases to on-board car systems while the car is still on the road.