**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY**

# BUILDING SEMANTIC BASED PUBLIC TRANSPORTATION GEOPORTAL COMPLIANT WITH THE INSPIRE TRANSPORT NETWORK DATA THEME

**Ph.D. THESIS**

**Arif GÜNAY**

**Department of Geomatics Engineering**

**Geomatics Engineering Programme**

**FEBRUARY 2013**

**<u>ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY</u>**

# BUILDING SEMANTIC BASED PUBLIC TRANSPORTATION GEOPORTAL COMPLIANT WITH THE INSPIRE TRANSPORT NETWORK DATA THEME

**Ph.D. THESIS**

**Arif GÜNAY**
**(501072601)**

**Department of Geomatics Engineering**

**Geomatics Engineering Programme**

**Thesis Advisor: Prof. Dr. M. Orhan ALTAN**

**FEBRUARY 2013**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# SEMANTİK TABANLI TOPLU TAŞIMA GEOPORTALININ INSPIRE ULAŞIM VERİ MODELİNE UYUMLU OLARAK GELİŞTİRİLMESİ

**DOKTORA TEZİ**

**Arif GÜNAY**
**(501072601)**

**Geomatik Mühendisliği Anabilim Dalı**

**Geomatik Mühendisliği Programı**

**Tez Danışmanı: Prof. Dr. M. Orhan ALTAN**

**ŞUBAT 2013**

**Arif Günay**, **a Ph.D.** student of ITU **Graduate School of Science Engineering and Technology** student ID 501072601, successfully defended the **dissertation** entitled "**BUILDING SEMANTIC BASED PUBLIC TRANSPORTATION GEOPORTAL COMPLIANT WITH THE INSPIRE TRANSPORT NETWORK DATA THEME**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

| | | |
|---|---|---|
| **Thesis Advisor :** | **Prof. Dr. M. Orhan ALTAN**<br>İstanbul Technical University | ………………… |
| **Jury Members :** | **Prof. Dr. Nuran Zeren GÜLERSOY**<br>İstanbul Technical University | ………………… |
| | **Prof. Dr. Ferruh YILDIZ**<br>Selçuk University | ………………… |
| | **Prof. Dr. Haluk GERÇEK**<br>İstanbul Technical University | ………………… |
| | **Asst. Prof. Özgün AKÇAY**<br>Çanakkale Onsekiz Mart University | ………………… |

**Date of Submission : 28 December 2012**
**Date of Defense : 18 February 2013**

## FOREWORD

Firstly, I would like to thank my superviser Prof. Dr. M. Orhan Altan for his support and advice. He has provided me valuable assistance and guidance to carry out this research. Also, I appreciate Prof. Dr. Nuran Zeren Gülersoy and Prof. Dr. Ferruh Yılmaz who have given me complete freedom and assurance needed to develop my dissertation.

I owe special thanks to Asst. Prof. Özgün Akçay who had great influence on the development of my research interest in gespatial semantics and INSPIRE. His expertise and encouragement are invaluable throughout completion of this thesis.

Furthermore, I would like to thank my colleagues and friends, especially Res. Asst. Umut Aydar and Res. Asst. Özgür Avşar in the Department of Geomatics Engineering for their encouragement and support throughout my research study.

I also appreciate Assoc. Prof. Murat Ergün in the Transportation Division of Civil Engineering Department who has provided the sample public transportation dataset for this thesis research.

February 2013

Arif GÜNAY
Geomatic Engineer, M.Sc.

**TABLE OF CONTENTS**

**Page**

**ABBREVIATIONS**

**DL** : Description Logic
**EU** : European Union
**EuroSDR** : European Spatial Data Research
**GCM** : Generic Conceptual Model
**GIS** : Geographic Information Systems
**GML** : Geographic Markup Language
**GNM** : Generic Network Model
**GPRS** : General Packet Radio Service
**GPS** : Global Positioning System
**GSM** : Global System for Mobile Communications
**HTTP** : Hypertext Transfer Protocol
**ICT** : Information and Communication Technology
**INSPIRE** : Infrastructure for Spatial Information in the European Community
**ISO** : International Organization for Standardization
**LBS** : Location Based Services
**NCGIA** : National Center for Geographic Information and Analysis
**NMA** : National Mapping Agencies
**nRQL** : New Racer Query Language
**OGC** : Open Geospatial Consortium
**OWL** : Web Ontology Language
**RDBMS** : Relational Database Management System
**RDF** : Resource Description Framework
**SGA** : State Geodetic Administration
**SDI** : Spatial Data Infrastructure
**SOAP** : Simple Object Access Protocol
**TCP-IP** : Transmission Control Protocol and Internet Protocol
**TWG-TN** : INSPIRE Thematic Working Group Transport Networks
**UDDI** : Universal Description, Discovery and Integration
**UML** : Unified Modelling Language
**UMTS** : Universal Mobile Telecommunications System
**URI** : Uniform Resource Identifier
**WMS** : Web Map Service
**WFS** : Web Feature Service
**WCS** : Web Coverage Service
**WLAN** : Wireless Local Area Network
**WSDL** : Web Services Description Language
**W3C** : World Wide Web Consortium
**XMI** : XML Metadata Interchange
**XML** : Extensible Markup Language

## LIST OF FIGURES

**Page**

**BUILDING SEMANTIC BASED PUBLIC TRANSPORTATION GEOPORTAL COMPLIANT WITH THE INSPIRE TRANSPORT NETWORK DATA THEME**

## SUMMARY

Interoperability is the important key for the today's Geographic Information Systems (GIS). However, the information sources are more heterogeneity (syntactic, structure and semantic), and it shows differences in modeling, representation, and accessing of sources. The development of Spatial Data Infrastructure (SDI), such as the European Infrastructure for Spatial Information in the European Community (INSPIRE), is necessitating existing spatial data standards to be harmonized or revised, based on modern international standards to overcome the interoperability issues. Also, researches in geospatial interoperability must go beyond geometry-related and database-structure concerns to take into account of semantics heterogeneity. Different terms may refer to similar concepts and the same terms may refer to different concepts, which cause semantic heterogeneity in the dataset. Relational Database Management System (RDBMS) has being used for a long time to handle spatial problems with using their query capabilities on huge data repositories. However, this approach falls short to fully express the user needs and perform intelligent queries, while the information sources are suffer from the semantic heterogeneity. The aim of this research is to investigate the implementation of INSPIRE as a SDI standard and Semantic Technology to empower the traditional GIS approach.

The methodology of this research firstly applied for the existing transportation dataset to harmonized based on the INSPIRE Transport Data Specification to meet the interoperability in European SDI Level. Data standards are typically represented by conceptual schemas on a high abstraction level, together with application schemas on lower levels of abstraction. They serve the purpose of creating interoperability at all levels (data format, data structure and data content) between the resources that comply with these standards. Application schema holds information about data structure and data content of the data set. In this regard, INSPIRE Transport Network Data Content Specification is used as an application schema to build the public transportation geodatabase during this research.

The design of the Geodatabase has been done using the Microsoft Visio Case Tool, based on the Unified Modeling Language (UML) formalisms object-oriented paradigm. In particular, in the case of the Geodatabase, several diagrams able to represent the hierarchical structure of the database are proposed. The term "hierarchical" means that the Geodatabase is constituted by "feature datasets" containing in turn "feature classes", the latter being characterized by attributes. All the attributes are set as public, in order to allow a total vision among the classes. Each feature dataset has been modeled as a package. Inside the packages are present the UML classes, with the relevant attributes, modeling the feature classes. Some of

the attributes are of common type (Integer, String, Date, etc.), while other refer to values belonging to defined domains. In order to properly model this case, the choice to create stereotype domains as <<CodedValueDomain>> (numerical or textual) and to directly refer the attributes of interest to these domains. The UML modeling of the feature datasets are created as Administrative_Units, Transport_Networks packages. Administrative_Units feature dataset include feature classes of Province and District. Transport Network feature dataset designed by covering the transportation features of road and waterway. Road and waterway link feature models are derived from the CommonTransportLink object as defined in the INSPIRE Transport Network Data Content. All the attribute and necessary information of RoadLink and WaterwayLink feature models are created in UML in according to the INSPIRE Transport Network Data Content. As for the node of the Transport Network, CommonTransportNode class is used an abstract class to derive Road node, waterway node and port node as defined in INSPIRE Transport Network Data Content as well.

According to definition of INSPIRE Transport Network Link Sequence Abstract Class: “A network element which represents a continuous path in the network without any branches. The element has a defined beginning and end and every position on the link sequence is identifiable with one single parameter such as length. A link sequence may represent a route”. Starting from this definition, Transportation Routes (bus, sharedtaxi and ferry) data models are designed to derive from the Common TransportLink Sequence class in our UML Transportation Data Model. Transportation Stops (bus, sharedtaxi and ferry) are derived from the CommonTransportPoint class as referred to the INSPIRE Transport Network Data Content as well. Other attributes refer to range of values have been modeled as <<coded and range>> stereotype domains compliant with the INSPIRE Transport Network Domains. The modeled Transport Network Schema in UML is exported to a XML Metadata Interchange (XMI) file. Afterwards, this exported XMI file is used in order to generate Transportation Geodatabase with the use of Schema Wizard tool of ArcCatalog.

In addition to this, geospatial semantic by means of ontologies is implemented public transportation dataset, which allows for automated reasoning and intelligent spatial queries during this research. Traditional GIS Systems are not able to handle intelligent queries (semantic) just by seeking of required data as defined in RDBMS, while the information sources are, suffer from the semantic heterogeneity. Semantic technology empowers machines to understand and response the human queries based on their meaning and it makes searching more effective. Ontology is the backbone of semantic technology. Using ontology as a knowledge base improves the efficiency of data retrieval and users can get data in meaningful way. In order to solve the problems of semantic heterogeneity, caused by synonyms – homonyms, ontology as a concept model is implemented in this research. The relevant concepts and relations, which are reasoned by inferred semantics, are discovered in SDI to search suitable GIS data and services.

Public Transportation Ontology Domain is defined considering the designed Transport Network Schema. Modeled class hierarchy in Transportation Ontology Domain described as followed:

1) Public Transportation Route, Public Transportation Link and Public Transportation Stop are subclass of Public Transportation Network

2) Under the Transportation Route Class, BusRoute, FerryRoute and SharedTaxiRoute classes are created
3) BusLink, FerryLink and SharedTaxiLink classes are created under the Transportation Link class
4) In same way, Bus, Ferry and Shared Taxi Stops are created as subclass of Transportation Stops

Then, object properties, which represent the relationships in this ontology, are created as hasLink, hasStop properties. Also, each object property may have a corresponding inverse property. If some property links individual a to individual b then its inverse property will link individual b to individual a. Therefore, in our ontology domain, isStopOf object property is created as inverse property of hasStop and isLinkOf object property is created as inverse property of hasLink. In order to define relationships between classes in ontology restrictions needs to be declared. For example, the class of individuals that have at least one (some) hasStop relationship to members of Public Transportation Stop. In ontology editor (in our case Protégé software is used) the keyword, `some' is used to denote existential restrictions.

After defining classes, object properties and relations using with restrictions, all individuals related to our dataset are created in ontology domain. One important thing we need to emphasis during the individual creation is that if there is any common stop name for the transportation types, it is possible to declare these stops as Same Individual in ontology. For example, Public Shared Taxi Stop – Taksim_Meydan is defined as same individual with Taksim stop which is actually belongs to Public Bus Stop class. Datatype properties are also used to describe relationships between an individual and data values in this ontology domain. A datatype property can be used to relate an individual to a concrete data value that may be typed or untyped. Datatype property hierarchy is created as hasLinkGeoId, hasLinkOrderId, hasStartStopGeoId, hasEndStopGeoId, hasRouteName, hasRouteNumber during this research. After overall definitions, the Public Transportation Ontology Domain is obtained. In order to perform semantic queries in transportation ontology domain, new Racerpro Query Language (nRQL) of RacerPro Reasoning Engine is used. Semantic spatial query algorithms are developed in order to retrieve transportation route alternatives based on the user requested stop points.

In conclusion, homogeneous reasoning system is obtained by covering the transportation modes bus, sharedtaxi and ferry in the same transportation system. Also, the used transportation network data is revised based on the INSPIRE Transport Standards considering the needs of interoperability in European SDI level during this research. The result is presented on a developed geoportal which provides a knowledge-based search (semantic search) and GIS services to publish obtained result. Experimental investigation shows that the developed ontology domain express the rich semantic knowledge of three dimension transporting means (bus, shared taxi, ferry) in formalization and related administrative areas. Ambiguity of the concepts such as differentiating the same station names can be eliminated and hidden information can be discovered by reasoning. In this regards, the use of ontology can allow us to perform intelligent spatial queries by considering the concepts compare to the traditional GIS applications. Also, INSPIRE Transport Network Data Specifications are investigated and methods, technics are gained about building the INSPIRE Data Content as learning process during this research.

## SEMANTİK TABANLI TOPLU TAŞIMA GEOPORTALININ INSPIRE ULAŞIM VERİ MODELİNE UYUMLU OLARAK GELİŞTİRİLMESİ

### ÖZET

Birlikte çalışabilirlik günümüz Coğrafi Bilgi Sistemleri (CBS) teknolojisi için önemli bir rol oynamaktadır. Ancak, bilgi kaynaklarının fazla heterojen yapıda olması (veri formatı, veri yapısı ve veri içeriği), modelleme, gösterim ve kaynakların erişiminde farklılıkların ortaya çıkmasına sebep olmaktadır. Birlikte çalışabilirlik sorunları aşmak için ulusal ve/veya uluslararası olarak mekansal veritabanı standartlarının Spatial Data Infrastructure (SDI) oluşturulması çalışmaları yürütülmüktedir. Avrupa da mekansal veri değişim ağının sağlanabilmesi için geliştirilmiş olan Infrastructure for Spatial Information in the European Community (INSPIRE) mekansal veritabanı standardı bu çalışmaların bir örneğidir. Ayrıca, mekansal veri birlikte çalışabilirlik araştırmalarında, veritabanı yapısı ve geometrik sorunların aşılmasına yönelik çalışmaların yanında semantik heterojenliğin giderilmesi üzerine de çalışmaların yürütülmesi gerekmektedir. Veri seti içerisindeki farklı terimler benzer anlamları ve aynı terimler farklı anlamları çağrıştırabilir. Bu gibi kavramsal farklılıklar semantik heterojenliğe sebep olmaktadır.

İlişkisel veritabanı yönetim sistemleri Relational Database Management System (RDBMS) büyük veri depoların yönetiminde ve mekansal sorgulamaların gerçekleştirilmesinde uzun yıllardır kullanılmaktadır. Fakat, bu tür geleneksel sistemlerde semantik kavramlar ele alınmadığından dolayı, akıllı (semantik) mekansal sorgulamaların elverişli bir şekilde gerçekleştirilmesi mümkün olmamaktadır. Bu tez çalışmasında, geleneksel CBS uygulamalarında karşılaşılan bu kısıtların giderilmesi ve mevcut uygulama yaklaşımlarının zenginleştirilmesi amacı ile semantik teknolojinin uygulanarak akıllı sorgulamaların gerçekleştirilmesi ve mekansal veri standartlarının modernizasyonu için de INSPIRE Ulaşım veri modelinin uygulanması üzerine çalışmalar yürütülmüştür.

Bu araştırmanın metodolojik yaklaşımında ilk adım olarak uygulamada kullanılan mevcut ulaşım veri seti INSPIRE Ulaşım veri modeli standardına göre revize edilmiştir. Böylelikle, mevcut veri seti Avrupa SDI standardı olarak kullanılan INSPIRE'a göre modernizasyonun sağlanması hedeflenmiştir. Veri standartları genel olarak üst düzey bir soyutlama ile kavramsal şeması ve alt düzey detay bir soyutlama ile uygulama şeması olarak sunulmaktadır. Uygulama şeması, veri yapısı ve veri setinin veri içeriği hakkında bilgileri sağlamaktadır. Bu standartlara uyularak her düzeyde (veri formatı, veri yapısı ve veri içeriği) birlikte çalışabilirliği yaratmak mümkün olabilmektedir. Bu bağlamda, INSPIRE'ın sunmuş olduğu ulaşım ağı veri modeli standartları dikkate alınarak uygulamada kullanılacak olan toplu taşıma mekansal veritabanı şeması oluşturulmuştur.

Ulaşım veritabanı tasarımı Microsoft Visio aracı kullanılarak nesneye yönelimli Unified Modeling Language (UML) dili kullanılarak yapılmıştır. Özellikle, veri tabanı hiyerarşik yapısını temsil edebilecek yapıda veri modeli dizaynı oluşturulmaya çalışılmıştır. Veritabanı şeması hiyeraşik yapı ile detay verisetleri (feature dataset),

detay verilerin içerisinde detay sınıfları (feature classes) ve devamında da detay sınıfların içerisinde özniteliklerin yer alması gibi tanımlamalar ifade edilmeye çalışılmıştır. Tüm öznitelikler diğer detay sınıflar tarafından da görülebilmesi için "public" anahtar sözcüğü ile tanımlanmıştır. Her bir detay veriseti (feature dataset) UML notasyonunda "package" olarak tanımlanmıştır. Detay verisetleri içerisinde de detay sınıfları UML modelleri ile oluşturulmuş ve ilgili öznitelikler detay snıfları içerisinde yaratılmıştır. Öznitelikler yaygın olarak kullanılan integer, string, double gibi veri formatlarında tanımlanırken, bazıları da domain yapılarında tanımlanmıştır. Domain tanımlamaları "CoddedValueDomain" stereotype özelliğinde sayısal veya text bazlı olarak yapılmıştır.

Bu araştırma kapsamında uygulanacak toplu taşıma ağı geoportalı için, idari sınırların yer aldığı ilçe ve mahalle katmanların tutulacağı "Administrative Units" ve ulaşım katmanların yer alacağı "Transportation Networks" detay verisetleri UML de yaratılmıştır. Transportation Network detay veriseti içerisinde karayolu ve deniz yoluna ilişkin detay veri sınıflarını barındırmaktadır. Karayolu ve Denizyolu detay veri bağlantı sınıfı INSPIRE veri modeli özelliklerinde tanımlanan "CommonTransportLink" soyut sınıfından türetilerek oluşturulmuştur. Karayolu ve Denizyolu ile ilgili tüm öznitelik bilgileri, INSPIRE veri modeli içerik bilgilerinden alınarak oluşturulmuştur. Karayolu ve Denziyolu ağ düğüm noktaları, benzer şekilde INSPIRE "CommonTransportNode" soyut sınıfından türetilerek oluşturulmuş ve ilgili öznitelik bilgileri bu sınıflarda tanımlanmıştır.

Benzer şekilde, toplu taşıma ağı verileri otobüs, dolmuş ve deniz otobüsü ulaşım verileri de INSPIRE ulaşım veri temasından faydalınarak yaratılmıştır. INSPIRE Transport Network Link Sequence Soyut Sınıf tanımına göre: "Herhangi bir kola ayrılmadan şebekede sürekli bir yolu temsil eden ağ elemanı Link Sequence olarak tanımlanabilir. Şebekede başlangıç ve bitiş noktası ile tanımlanabilen 'link sequence' için ayırt edici özellik uzunluk bilgisidir. Bir 'link sequence' ulaşım şebekesinde güzergah olarak tanımlanabilir." Bu tanımlamadan yola çıkarak, uygulamada kullanılacak otobüs güzergahı, dolmuş güzergahı ve deniz otobüsü güzergahı detay sınıfları INSPIRE "CommonTransportLinkSequence" soyut sınıfından türetilerek oluşturulmuştur. Otobüs, dolmuş ve deniz otobüsü durak noktaları ise INSPIRE "CommonTransportPoint" soyut sınıfından türetilerek oluşturulmuştur. Ulaşım ağı için gerekli diğer öznitelik bilgileri de stereotype "CoddedValueDomain" yapısında oluşturularak UML ulaşım veri modeline eklenmiştir. Bu çalışmalar sonrasında INSPIRE ulaşım veri modeli dikkate alınarak oluşturulmuş UML veri tabanı tasarımı elde edilmiştir. Sonrasında, bu veri modeli şemasından mekansal veritabanın yaratılması için Visio da bulunan "XMI export add-on" aracı ile şemanın XML Metadata Interchange (XMI) formatına dönüşümü gerçekleştirilmiştir. Bu XMI dosyası ArcCatalog yazılımında bulunan "Schema Wizard" aracı ile kullanılarak UMLde tasarlanan veri modeli yapısında ulaşım ağı mekansal veritabanı oluşturulmuştur.

Veri standartları üzerine yapılan bu çalışmalara ek olarak, ulaşım veriseti üzerinde akıllı mekansal sorgulamaların, kavramsal çıkarımların yapılabilmesi için semantik teknoloji (ontoloji) de bu araştırma kapsamında uygulanmıştır. Geleneksel Coğrafi Bilgi Sistemleri uygulamaları mekansal sorgulamaları RDBMS üzerinden yapmaktadır, fakat semantik heterojen yapılarda bu sorgulamalar kavramsal farklılıkları göz ardı etmekte ve sorgulama sonuçlarında yetersiz kalabilmektedir. Bu gibi benzeri problemlerin aşılmasında semantik teknolojinin uygulanması ile veriler üzerinden kavramsal çıkarımların yapılması ve daha akıllı mekansal sorgulamaların gerçekleştirilmesi sağlanabilir. Ontology semantik teknolojinin asıl omurgasını oluşturmaktadır. Bu araştırma kasamında da semantik heterojenliğin (kavramsal benzerliklerin) giderilmesi amacı ile ontoloji konsepti uygulanmıştır ve toplu taşım ağı için bir ontology domain'i oluşturulmuştur. Sorgulanan mekansal veri için ontology domain üzerinde tanımlı kavram ve ilişkiler dikkate alınarak sonuçların elde edilmesi ve sonrasında da tanımlanan SDI içerisinde ilgili verilerin bulunması sağlanmıştır. Toplu Taşıma Ağı Ontology Domain'i oluşturulan ulaşım ağı mekansal veritabanı dikkate alınarak geliştirilmiştir.

Toplu Taşıma Ağı Ontology sınıf hiyeraşisi aşağıda örneklendiği biçimde yaratılmıştır:

1) Toplu Taşıma Güzergah (Public Transportation Route), Toplu Taşıma Bağlantı (Public Transportation Link) ve Toplu Taşıma Durağı (Public Transportation Stop) sınıfları Toplu Taşıma Ağı (Public Transportation Network) alt sınıfları olarak yaratılmıştır.
2) Otobüs Güzergahı (BusRoute), Deniz Otobüsü Güzergahı (FerryRoute) ve Dolmuş Güzergahı (SharedTaxiRoute) sınıfları, Toplu Taşıma Güzergah (Public Transportation Route) alt sınıfları olarak oluşturulmuştur.
3) Otobüs Bağlantı (BusLink), Deniz Otobüsü Bağlantı (FerryLink) ve Dolmuş Bağlantı (SharedTaxiLink) sınıfları, Toplu Taşıma Bağlantı (Public Transportation Link) sınıfının alt sınıfları olarak oluşturulmuştur.
4) Benzer hiyeraşi, Otobüs, Deniz Otobüsü ve Dolmuş durak noktaları sınıflar için Toplu Taşıma Durağı (Public Transportation Stop) sınıfının alt sınıfları olarak oluşturulmuştur.

Sonrasında bu ontoloji sınıfları arasında ilişkileri tanımlamaya yarayan obje özellikleri (object properties) yaratılmıştır. Bu obje özellikleri güzergah bağlantı ilişkilerini tanımlaması için "hasLink" ve durak ilişkilerinin tanımlanması için "hasStop" olarak yaratılmıştır. Bu ilişkiler üzerinden kavramsal çıkarımların yapılabilmesi için her bir obje özelliğinin ters işlevi (inverse object properties) olarak "isLinkOf" ve "isStopOf" özellikler yaratılmıştır. Ontology sınıfları arasında ilişkilerin kurulması ontology editor yazılımları tarafından belirli anahtar sözcükler kullanılarak tanımlanabilmektedir. Örneğin, Toplu Taşıma Durak sınıfının en az bir durak noktasına sahip olması (some) "hasStop" ilişkisi ile tanımlanmaktadır. Ontology Domain'nin oluşturulması için bu çalışma kapsamında Protégé Ontology Editor yazılımı kullanılmıştır ve en az bir ilişkisi bu yazılımda (some) anahtar sözcüğü ile tanımlanmaktadır.

Toplu Taşıma Ağı Ontology Domain'i içerisinde en az bir (some) ilişkisi kurulmuş sınıflar aşağıda listelenmektedir.

- Toplu Taşıma Güzergah hasLink (some) Toplu Taşıma Bağlantı
- Toplu Taşıma Bağlantı isLinkOf (some) Toplu Taşıma Güzergah
- Toplu Taşıma Bağlantı hasStop (some) Toplu Taşıma Durağı
- Benzer şekilde ilşkiler Toplu Taşıma Güzergah (Otobüs, Deniz Otobüsü ve Dolmuş) alt sınıfları içinde tanımlanmıştır.

Toplu Taşıma Ağı Ontology'si içerisinde sınıfların, obje ilişkilerin ve kısıtlamaların tanımlanmasından sonra uygulama veri seti içerisindeki gerekli bilgiler ontology domain de "individual" olarak yaratılmıştır. Bu bilgiler "individuals" veri setindeki kavramsal farklılıklarda dikkate alınarak ontology'e aktarılmıştır. Örneğin, otobüs durağı ismi "Taksim_Meydan" ile veritabanında yer alırken aynı durak noktası "Taksim" ismi ile dolmuş durağı olarak verisetinde yer almaktadır. Bu iki durak noktası lokasyon olarak birbirlerine yakın noktalarda bulunmaktadır ve farklı kelime yazılımında verisetinde yer almaktadır. Fakat, kullanıcı Taksim noktasından güzergah sorgulaması yapmak istediği zaman, uygulamanın her iki durak noktasını da göz önüne alarak bir sorgu vermesi beklenecektir. Dolayısıyla, Bu iki durak arasında "synonym – same individual" tanımlaması kurularak kavramsal tanımlama yapılmıştır. Bu gibi semantik heterojenliğe sebep olabilecek kavramlar ontology de dikkate alınarak gerekli modellemeler yapılmıştır. Ayrıca, tanımlanan veriler (individual) arasında da sınıflar arasında olduğu gibi ontology'e girilmiş veriler "individual" arasında da ilişkiler tanımlamak mümkündür. Bu tanımlamaların yapılabilmesi için veritipi özellikleri (datatype properties) ontology de tanımlanmıştır. Tanımlanan veritipi özellikleri; "hasLinkGeoId", "hasLinkOrderId", "hasStartStopGeoId", "hasEndStopGeoId", "hasRouteName", "hasRouteNumber" isimleri ile güzergahlar ve duraklar ile ilgili verileri tanımlamakta kullanılmaktadır. Tüm bu tanımlamaların sonucunda oluşturulan ulaşım ağı veritabanı için toplu taşıma ağı ontology domain'i yaratılmıştır. Geliştirilen bu ontology domain'i üzerinde semantik akıllı sorgulamaların yapılabilmesi için new Racerpro Query Language (nRQL) dili kullanılarak RecerPro Reasoning Engine üzerinde sorgulama algortimaları geliştirilmiştir.

Sonuç olarak, aynı ulaşım sistemi içerisinde semantik heterojenlikten arındırılmış otobüs, dolmuş ve deniz otobüsü güzergahlarını ve duraklarını içeren homojen bir toplu taşıma ağı kurulmuştur. Bu ağ için kullanılan veriseti aynı zamanda INSPIRE ulaşım veri standartları dikkate alınarak revize edilmiş ve modernize edilmiş bir ulaşım mekansal veritabanı sistemi bu çalışma çerçevesinde oluşturulmuştur. Oluşturulan sistemin sonuçları semantik tabanlı (ontology) arama ve CBS hizmetleri sağlayan bir Geoportal üzerinde sunulmaktadır. Geliştirilen bu Geoportal üzerinden kullanıcılar tarafından istenilen güzergah için başlangıç ve bitiş durak noktaları alınmaktadır. Belirtilen bu durak noktaları bilgisi oluşturulan toplu taşıma ağı ontology'si üzerinde araştırılarak ilgili güzergah bilgileri sorgulanmaktadır. Ontology bazlı gerçekleştirilen sorgulama sonucuna göre ulaşım mekansal veritabanında ilgili güzergahlar listelenmektedir. Sorgulama sonucunda elde edilen güzergah alternatifleri oluşturulan Web Map Service (WMS) servisi üzerinden geliştirilen Geoportal üzerinde kullanıcılara sunulabilmektedir.

Gerçekleştirilen bu araştırma sonucunda, ontology domain'i içerdiği kavramsal bilgiler ile toplu taşıma ağı portal uygulamasını zenginleştirmekte ve otobüs, dolmuş, deniz otobüsü ulaşım ağı üzerinde homojen bir yapı kurarak akıllı sorgulamaların yapılmasına olanak sağlamaktadır. Örneğin, anlamsal olarak aynı olan durak noktaları, farklı kelime yazımları ile veriseti içerisinde temsil ediliyor olsa bile sorgulamalar esnasında aynı durak noktası şeklinde davranarak kullanıcılara daha doğru sonuçların elde edilmesine olanak sağlamaktadır. Bu gibi kavramsal bilgilerdeki farklılıklar, klasik CBS mekansal sorgulamalarında dikkate alınmamaktadır, dolayısıyla kullanıcılara eksik veya yetersiz sonuçların iletilmesine sebep olmaktadır. Bu çalışmalara ek olarak, mevcut ulaşım veriseti INSPIRE'a göre revize edilmiştir. Bu kapsamda, INSPIRE Ulaşım Veri Modelinin uyarlanması için gerekli araçların, tekniklerin ve metotların araştırılması ve uygulanması bu tez çalışmasında gerçekleştirilmiştir.

# 1. INTRODUCTION

## 1.1 Motivation

The wider use of information created by Geographic Information Systems (GIS) has long been held back by difficulties in sharing, using and accessing data between different technologies, different organizations and even between different functions within the same organisation. The GIS software systems have evolved largely in the absence of any widely accepted body of theory or concept, with each vendor adopting a distinct terminology and set of data format standards, and with each product forming the nucleus of its own information community (Sun et al., 2009). The geographic information is usually available as datasets stored in databases and accessible via GIS. However, these information sources are more heterogeneity (syntax, structure and semantic), each of them created for one specific purpose and shows differences in modeling, representation, and accessing of sources (Vaccari et al., 2009; Wicaksana, 2006). Achieving efficient data conversion and integration between information sources has always been crucial for extracting maximum value from database assets, and is one of the most important areas of research for GIS field (Boucher and Zimányi, 2010). To satisfy increased demands for the use, sharing, and exchange of geographic data in the common applications, methods to support interoperability are required by the GIS community. There are two major impediments for GIS interoperability:

1) Syntax (differences in data format) and structure (differences in schemas) problem. Geo-datasets are retrieved from different sources that can use different data formats and geographical features can be represented using different geometrical and data schemas. Often the same geographic feature is represented using different geometric features. When people deal with different geospatial data syntax and structure, they often face challenges.

The development of interoperability standards - notably from the Open Geospatial Consortium (OGC), International Organization for Standardization (ISO) and World

Wide Web Consortium (W3C) – is driving a new generation Spatial Data Infrastructure (SDI). SDI is a broad term for systems that are designed to enable the interoperability and harmony between geographic datasets. SDI shall facilitate the cross-border cooperative use of distributed geoinformation services, and thus promote the paradigm shift from previously dominant GIS to open interoperable systems (Wiemann and Bernard, 2010). INfrastructure for SPatial InfoRmation in Europe (INSPIRE) is an example for the European Union's (EU) international SDI standard (Lemmens, 2006).

Current standardization work (e.g. OGC, ISO, INSPIRE) is necessary to facilitate integration and interoperability. Although significant progress has been made regarding the systemic and syntactic heterogeneity of data in this context, semantic interoperability remains insufficiently addressed (Fan and Zlatanova, 2011).

2) Semantic problems deal with the consistency of meaning and understanding of the contained geospatial objects. Different semantics are caused by different reasons such as general complexity of geographic concepts, contextual and linguistic variations in vocabularies, space-scale sensitivity of geographic vocabularies. These would indicate different geospatial semantics.

Spatial relationships give more semantic information than relationships from traditional databases. In order to take advantage of this kind of additional knowledge in the spatial domain, it is necessary to represent this into the ontology in a suitable way (Zurita, 2004). Semantic heterogeneity of GIS can be undertaken by considering their ontological aspects. Ontologies offer significant benefits to the design and use of geographic information, beyond the abilities of today's approaches. Ontologies define semantics independently of data representation and reflect the relevance of data without accessing them (Lorenz et al., 2005). It empowers machines to understand and response the human queries based on their meaning and it makes searching more effective and intelligent (Sherimon et al., 2011). Researches in geospatial interoperability must go beyond geometry-related and database-structure concerns by taking into account of semantics heterogeneity (Sun et al., 2009). Ontology papers dominate many conferences such as Geographic Information Science (GIScience) or Association of Geographic Information Laboratories for Europe (AGILE) and most relevant journals have had special issues on this topic (Albrecht et al., 2008).

## 1.2 Scope and Objective

Interoperability is becoming necessity in GIS science for data access and sharing and is enforced through adoption of consistent standards for spatial data technical specifications such as data formats, database schema, object concepts and application syntax. The current interoperability capabilities are reflected in the European Unions' initiative to create a regional SDI – INSPIRE.

INSPIRE is an initiative developed with the intention of developing an European spatial information infrastructure delivering integrated information services that allow the users to identify and access spatial or geographical information in an interoperable manner. To support the establishment of a European infrastructure, the INSPIRE Directive defines 34 themes in an interoperable way of the spatial data that shall be standardized by the Member States. INSPIRE does not require collection of new data. However, after the period specified in the Directive Member States have to make their data available according to the Implementing Rules.

The guiding principles of INSPIRE are that the infrastructures for spatial information in the Member States will be designed to ensure that spatial data are stored, made available and maintained at the most appropriate level; that it is possible to combine spatial data and services from different sources across the Community in a consistent way and share them between several users and Applications (INSPIRE Directive).

INSPIRE has been an answer to meet the necessity of the sharing, access and exchange of geographic data in Member States by implementing the common data specification (INSPIRE Themes).

Considering the EU membership process of Turkey, implementation of INSPIRE SDI Standard is increasingly disseminating and becoming big necessity of interoperable, accessible and harmonized datasets based on modern international standards. The INSPIRE initiative is therefore timely and relevant for Turkey but also a major challenge given the general situation outlined above and the many stakeholder, organization interests needs to be addressed.

One of the main objectives of this research is to investigate INSPIRE SDI Standard to meet the interoperability in Member State Level and improve availability, accessibility and usability of geographic data. Thus, it is expected that users will spend less time and efforts on understanding and integrating data when they build their applications based on data delivered within INSPIRE.

INSPIRE initiative is a potential to enable the interoperability and harmony between geographic datasets, foster new dimensions in business services. However, current standardization does not deal with the real differences in representing geographic knowledge. It does not focus on the meaning of concepts (semantics), problem remains of semantic interoperability.

Semantic interoperability requires shared ontologies of precisely defined concepts. Ontologies are generally considered as semantically richer than database schemas, and thus closer to the user's cognitive model. Database (DB) schemas are built to organize what is going to be stored in a database, and then are used to document it. However, this approach falls short to fully express the user needs, while the information sources are suffer from the semantic heterogeneity. If the terms are synonyms and/or homonyms in the dataset, spatial search inherently restricted by the ambiguities of natural language. Different terms may refer to similar concepts and the same terms may refer to different concepts (Klien, 2008). The promising approach to overcome the semantic heterogeneity is that implementation of geospatial semantic by means of ontology on the geospatial dataset. Thus, the use of ontologies in advanced GIS query languages will provide better access for domain-specific interactions and improve the efficiency and accuracy of data retrieval.

Therefore, the other main objective of this research is to investigate the Semantic Technology to empower the traditional GIS approach and provide intelligent search.

In this regard, a public transportation geoportal, including Bus, SharedTaxi and Ferry features, is developed for the experimental investigation which uses a revised sample transportation data complaint with INSPIRE Transport Network Data Theme and a developed Transportation Ontology Domain, including concepts, relationships and individuals to provide semantic spatial search.

More specifically, the research objectives are:

- Development of the Public Transportation Geotabase complaint with INSPIRE Transport Network Data Theme to meet the interoperability in Member State level and overcome the schematic heterogeneity so that improve the usage, sharing and exchange of geographic data in the common applications.

- Development of a Public Transportation Ontology Domain considering the INSPIRE Transport Network Schema and overcome the semantic heterogeneity so that ambiguity of the concepts such as differentiating the same station names can be eliminated and hidden information can be discovered by reasoning compare to the capability of traditional DB schemas.
- Development of a Public Transportation Geoportal for the purpose of experimental investigation.
- Investigation of using open source software to develop such GIS applications.

## 1.3 Research Questions

Research Questions:

- What are the specifications and requirements to implement INSPIRE Transport Data Content as a SDI standards?
- What are the tools and techniques to design and build transport geodatabase schema according to INSPIRE specifications?
- What are the key problems in building transport geodatabase based on INSPIRE and what solutions are available for?
- Upto what extent can and should semantics be modeled to support intelligent spatial queries?
- What are the tools and techniques to model and create a transport ontology domain?
- How to design system architecture to perform semantic based geo-operations instead of traditional methods?
- What consideration are required for doing reasoning for geospatial data retrieval?
- What are the key problems in building semantic geoportal using ontology domain and what solutions are available for?

## 1.4 Thesis Organization

Chapter two gives the information about the recent related works on implementation of SDI-INSPIRE and the ontology implementation for the GIS applications. Chapter three explains the usage of Geographic Information Standards in GeoScience. The role and means of the SDI and commonly used SDI standard INSPIRE in EU are discussed in detail here. Lastly, INSPIRE Transportation Network Data Contents, which will be used as a standard in this thesis research, is explained in this section as well. Chapter four gives the definition of Geospatial Web Services. The commonly used web service architecture OGC and web service standards OGC, which are Web Map Service (WMS), Web Feature Service (WFS) and Web Coverage Service (WCS), are explained in this chapter. Chapter five focuses on the concepts of Geospatial Semantic Web, Ontologies and Description Logic as knowledge representation. Afterwards, it provides the information about the ontology languages Resource Description Framework (RDF) and Web Ontology Language (OWL) that is used to develop ontologies. Then it introduces the syntax and basic constructors used in ontology languages to build knowledge bases. In the last part of the chapter, the role of reasoning engine in ontology is explained. In chapter six, the proposed methodology of this thesis is introduced and detail information is given about that. This chapter starts with the building transportation geodatabase compliant with the INSPIRE transport network data content and it continues with the development of transportation ontology domain in according to the built transportation geodatabase. Then, it explains the development of semantic spatial query algorithms using the Reasoning Engine. These developed semantic queries will be used in the developed geoportal in order to perform semantic based spatial queries compare to traditional GIS methods. Chapter seven presents the prototype design and implementation of the public transportation geoportal. This portal will run the queries using the ontology domain and perform semantic queries in order to get public transportation route information based on the user requests. Implemented geoportal architecture is described in this chapter such as presentation layer, service layer and data layer. The last section of this chapter evaluates the prototype implementations. Finally, chapter eight provides the conclusion of the research.

In addition, appendices are included into the document to give a quick reference to the research work. These include sections for Transportation UML Models

(Appendix A), Feature Dataset, Feature Classes and Domain Models Views in the Created Geodatabase (Appendix B), Public Transportation Ontology Domain OWL File (Appendix C), Transportation nRQL Query Algorithm (Appendix D), Developed Application Java Codes (Appendix E) and Developed Application User Interface OpenLayer Scripts (Appendix F).

# 2. RELATED WORK

## 2.1 Implementation of INSPIRE as a SDI Standard

Schade and Lutz (2010) have stated that Linked Data has recently received a lot of attention as a means for making the tacit connections between online-resources explicit and for advanced information browsing. It refers to a best practice for exposing, sharing, and connecting resources, such as textual documents, pictures and maps, in the Semantic Web. It is based on the use of Uniform Resource Identifiers (URIs) as reference points, the Resource Description Framework (RDF) as basic structure for any form of description, and content-negotiation to allow a client to specify an acceptable representation.

While numerous systems and applications for the environmental sciences have recently been based on the Linked Data approach, most of these were missing concrete use cases. Schade and Lutz (2010) investigated the opportunities to use a Linked Data approach in the context of SDI, which provides one concept supporting interoperability within and across involved information communities. From an SDI development perspective, using Linked Data for integration purposes seems a logical next step. They specifically focus on INSPIRE, which is one of the few SDIs with a legal basis. They analyzed example use cases for INSPIRE, identify research questions that need to be addressed in order to apply Linked Data in INSPIRE, and indicate future steps.

The Croatian SDI approach is truly national. SDI building blocks have reached a significant level of operationally. The law on State Survey and Real Property Cadastre contains a separate chapter on SDI. As during the adoption procedure of the law, the draft INSPIRE directive was available, the SDI chapter of the law was attuned to some of the provisions of the directive. However, not all provisions of the directive are included in the Law yet. Croatia has geodatasets that provide a basis for contributing to the coverage of pan-Europe for the INSPIRE-selected data themes and components. The geodetic reference system and projection systems are standardized, documented and interconvertable. The Croatian Geodetic Institute in collaboration with State Geodetic Administration (SGA) performs quality control of the topographic products (e.g. maps and orthophotos). In addition, quality documents for the management systems and process management systems are produced. A working group Technical Standards has been established focusing strongly on interoperability and data harmonization issues as defined in the specifications of the INSPIRE implementing rules. Finally, the national language can be considered as the operational language of the SDI with additional information in English (Vandenbroucke and Biliouris, 2010).

de Vries et al. (2011) describes in the research that how practice of INSPIRE implementation are affecting SDI development. It contains the results of a European Spatial Data Research (EuroSDR) project ‘Atlas of INSPIRE implementation methods’. Aim of the project was to make an inventory of experiences when implementing INSPIRE, in order to share exemplary practices and solutions among national mapping agencies and national INSPIRE contact points.

This inventory formed the basis for the generation of the prototype Atlas for all national mapping agencies, policy makers and other stakeholders who have to implement INSPIRE. For SDI research the Atlas provides empirical base material for the conceptualization of SDI implementation approaches. The analytical framework to look at INSPIRE implementation drew on two theoretical notions of how implementation actions can lead to information infrastructure development: a cultivation approach and a design approach.

A qualitative data collection process, through a survey and two workshops, tested the extent to which either of the two approaches were prevalent for the INSPIRE implementation. The survey and the workshops provided primary data on INSPIRE implementation experiences of representatives from the European countries. Comparing the national experiences showed that both types of approaches of INSPIRE implementation are present the EU countries. The cultivation approach is more prevalent in countries which established SDI organizational structures outside the National Mapping Agencies (NMAs), and the design approach is more prevalent in countries relying solely on NMAs for INSPIRE implementation. Embedding INSPIRE implementation in national SDI activities seems furthermore to relate to cultivation approaches, consisting of a gradually flatter inter-organizational working relations, and a scaling up strategy which iteratively links the national implementation plans of INSPIRE to the local implementation plans in national and sub national organizations, and vice versa. The variety in approaches imply that a uniform, best practice, INSPIRE implementation approach for all countries does exist, but that the choices for certain practices strongly relate to the local contextual conditions and windows of opportunities. The implication of these findings for research in SDI development is that more emphasis should be placed on the mechanism of interaction between the slowly changing socio-organizational context and rapidly technologies.

Hanzlova et al. (2011) states in this research that there has been a strong need linking nature conservation with geoinformation in Europe. The INSPIRE Directive (2007/2/EC) has been an answer to the necessity of interoperable, accessible and harmonized datasets. It also addresses EU SDI to support environmental policies. NATURE-SDIplus is an eContentplus Best Practice Network aiming at SDI for nature conservation. The approach of the NATURE-SDIplus network is to share data experiences and good practices, to improve exploitation of datasets, and to re-use information on nature conservation.

NATURE-SDIplus contributes to the implementation of the INSPIRE Directive with targeted reference to the cluster of data themes of the INSPIRE Annexes, i.e. Protected sites (Annex I), Biogeographical regions (Annex III), Habitats and biotopes (Annex III), Species distribution (Annex III). The mission of the NATURE-SDIplus project is to establish a network on geographical information for nature conservation, to stimulate the community of nature conservation stakeholders improving the harmonization, the exploitation and the accessibility of their datasets.

In such a view, NATURE-SDIplus aims to support the implementation of the INSPIRE Directive at EU level through the evaluation of common metadata profiles and data models for the addressed data themes, compliant with the INSPIRE provisions and data specifications. NATURE-SDIplus analyses the usability and accessibility of data. The results of this analysis were used to develop the NATURE-SDIplus European metadata profile and data model for datasets on nature conservation.

The project defines a common multilingual and multicultural approach for a simpler and standardized access to spatial data. A demonstration infrastructure, compliant with the INSPIRE principles and supported by web services, provides the data accessibility with a dedicated geoportal: the main gateway to available datasets and services.

The final target of NATURE-SDIplus is to establish a long-term sustainable network of stakeholders dealing with geo-information for nature conservation. Our research scope is dealing the Transportation Network Data Specification of INSPIRE different from this paper, while this paper study area is the data specification related to the NATURE-SDI.

Storgaard (2012) stated that the implementation of the INSPIRE Directive is well underway in Denmark. With law on Infrastructure for Spatial Information, Denmark has a solid foundation for the further development of the national infrastructure for spatial information. A development built upon a longstanding collaboration across the Danish public sector on common agreements and solutions.

An example of a common solution is the geoportal “geodata-info” which consist and distribute metadata of the Danish spatial data and services – among those the identified INSPIRE-data and services (e.g. view and download).

The new common public digital strategy is focusing on reusing data. Public authorities detect a wide range of information about citizens, businesses, real estate, buildings, roads, etc. and a small but very important part of these data are used again and again across the entire public sector. Such data are called basic data. They are the foundation for the authorities to discharge their duties properly, for citizens and legal certainty for companies and for society as a whole performance. Therefore suggest the common public digital strategy to make certain basic data authoritative as a unique data foundation for e-government secured.

The Infrastructure for Spatial Information Act constitutes a legal basis for developing the SDI in accordance with the EU’s INSPIRE Directive.

The act implements the regulations, principles and associated guidelines of the INSPIRE Directive into Danish law. With the Infrastructure for Spatial Information Act, Denmark has strengthened the framework for the national SDI and its links with eGovernment. This aim of this study is to implement INSPIRE Directive based on the Danish law, considering the regulations, principles, etc.

All research explained above dealing with the INSPIRE Directive and studies about that. However, these researches are differentiating each other by the data themes of the INSPIRE Annexes and application area.

The aim of our research is to implement a public transportation geoportal compliant with the INSPIRE Transport Data Content and perform the intelligent spatial queries by the use of ontology technology. In this regard, we have taken the INSPIRE Transport Network Data Theme as an application schema and harmonized existing transportation network data by following the guidelines explained in the INSPIRE Transport Network Data Specification. After the creation of Transportation Geodatabase compliant with the INSPIRE, we have developed the Transportation Ontology Domain including all relations, properties, individuals and more defined into the created geodatabase.

## 2.2 Implementation of Geo-Ontologies

Ontologies have been exploited in many domains and studies, for their capacity to promote sharability of knowledge bases, knowledge organization, and interoperability among systems. Some studies can also be found in the domain of transportation with different goals.

Niaraki and Kim (2009) study on a generic ontology-based architecture using a multi-criteria decision making technique to design a personalized route planning system.

The objective of this research is to determine an impedance model of road GIS and in intelligent transportation systems for a personalized ontology-based route planning system using a multiple criteria decision-making method. In this manner, personalized and user-centric route planning services using semantic technologies and ontologies perceive user and context models to satisfy user demands and predict their requirements.

This work was accomplished using a generic ontology-based architecture employing a multi-criteria decision making technique to extend a user-centric route planning modeling technique that encompasses three stages. First, a general personalized ontology-based user-centric architecture was constructed. Second, road segment ontology was obtained to determine appropriate criteria for designing a road segment impedance model including both user preferences and context characteristics for the personalized route planning. Third, a user model and context model were developed using the obtained criteria. In order to merge these criteria to create a road segment impedance model, a multi-criteria decision-making technique employing a ‘hierarchical structure’ and ‘criteria pair wise comparison’ was utilized. Finally, after deriving the personalized impedance model, the final personalized algorithm was implemented and verified in a GIS road environment.

In this research, all of the criteria to represent all the characteristics of the road network ontology were divided into two groups comprised of context and user variables. Road commuter traffic, safety, road facilities, and weather conditions are examples of context criteria. The user criteria include factors that affect an individual tourist’s trip such as tourist attractions and user preferences to define the impedance model. These criteria were selected based on the personalized road segment ontology and their influence on finding the optimum path in the road network.

Thus, the research result illustrates the capability and accuracy of the personalized user-centric algorithm based on a tourist travel model verified in a real-world situation which is actually not the same goal we are trying to achieve in our investigation.

Our goal is to construct a public transportation ontology domain for different kinds of transportation modes in water and land so that ambiguity of the concepts such as differentiating the same station names can be eliminated and hidden information can be discovered by reasoning.

In other words, homogeneous reasoning system can be obtained for the all transportation modes in the same transportation system.

Another work in this area is the research of Houda et al. (2010) that focus on the information required by the passenger to prepare a journey and to choose the best way to move from one point to the other using the same or several transport modes.

This ontology considers different transport modes but its objectives are different from ours since it deals with the other services such as restaurants, libraries, etc. that can be available in the route and useful for the passenger. To support the passenger’s journey planning by providing such services, they organized several kinds of journey patterns in a taxonomy based on the organization of transportation. In this way a journey pattern is classified as a direct journey pattern, when the journey pattern is composed only of stop points in journey pattern; or as an indirect journey pattern, when it has some connection stop points.

Another different objective is that all these journey information come from different systems and should be presented to the user in an integrated and organized view to make possible his choice and planning.

The query system supported by the public transportation ontology uses the internet to look for all possibilities of journey patterns in the different operators' databases while our purpose is to obtain homogeneous reasoning system in the same transportation ontology domain.

Zhang et al. (2010) research aim is to propose a solution for searching, discovering, and composing semantically heterogeneous transportation spatial data at feature level from different sources over the web through providing semantic specifications of Web Feature Service (WFS).

Without a formal semantic description of WFS, it is difficult for users and applications to perform an intelligent content-based search, and users cannot automatically compose and synthesize the discovered WFS without additional human assistance or programming.

For example, WFS may advertise as a freeway-feature data provider, whereas a user may request a highway feature from the data provider even though the freeway feature and the highway feature here refer to the same things. Thus to make geospatial features more practically searchable and ubiquitously available, a semantic-based approach is needed such that applications can use a WFS capability to a level of detail that permits automatic discovery and composition geospatial features.

To overcome the aforementioned problems and facilitate automatic geospatial feature discovery and integration, this study aims to extend the existing OGC WFSs with geospatial semantic web technologies and examine the use of geospatial semantic web Technologies such as description logic (DL) reasoner, inference rules, and web ontology language (OWL) ontology for OGC WFS descriptions to enable disparate GIS to share and integrate geospatial features at the semantic level.

The difference of this research method with our investigation is that this system use OGC WFSs to publish feature-level data through the web from semantically heterogeneous databases. The OGC WFSs are connected to OWL ontologies and map the OGC WFS descriptions to OWL ontologies to provide a semantically based view of the services.

In addition, this research domain ontology is created for feature data in general and application ontology includes only transit bus route and bus stop spatial data. Particularly, users can query any composite bus roads and bus stops within a certain distance (default is 1 km) of the selected bus stop in the study area. On the other hand, our ontology domain express the rich semantic knowledge of three dimension transporting means (bus, shared taxi, ferry) in formalization and related administrative areas.

Also, Kovarova and Szalayova (2010) has carried out a case study in public transportation departure notifications using Semantics in the field of widgets. They stated that Widgets are becoming increasingly present in our everyday routines, which makes their portability and reusability desirable properties. As a particular example, they consider public transportation passengers who are extensively using the internet to make their lives simpler. In order to minimize the time spent at the bus stop, passengers use to check their bus line departures on the web before they leave their homes or offices. For this purpose, there exist several internet portals providing information on local transportation time schedules. This case study starts by presenting a better way (quicker and easier) of obtaining the same information – using an adaptive desktop widget with comfortable user interface. The second step is the utilization of the semantics of considered data in order to make the widget portable through different data sources of the same domain.

Fan and Zlatanova (2011) has proposed a concept for a solution to the semantic interoperability problem in emergency management using an ontology by presenting a case study.  Emergency response is a complex activity involving many actors and heterogeneous spatial data. Two of the major challenges are the integration and extraction of these data and their transmission to emergency management actors.

Spatial data are stored in a large diversity of models not specifically intended for emergency response. Emergency managers want to get needed information by considering those various kinds of models before taking actions. However, mapping between emergency managers' models with these models should be built every time whenever either part of the mapped pairs is changed or a new model is needed. As ontology can define the uniform definitions for model's data, it may reduce the time and complexity of model mapping and integration so as to extract the correct information that would support the task of every actor efficiently. This paper discusses the possibility of applying the ontology to resolve semantic heterogeneity in emergency response.

Sherimon et al. (2011) presents the different development phases of ontology for Oman National Transport Company (ONTC). The company operates to different destinations from its headquarters. A person who wishes to travel in the ONTC can book his/her credentials in advance. The aim of the research is to improve efficiency of data retrieval by using ontology as a knowledge base. For example for an input of destination point, the system can retrieve the best fare.

The concept of this research ontology domain is the types of the buses such as parcel coaches and passenger buses. The price of the parcel coaches will be calculated depending upon the destination, mode of delivery and weight of the parcel. Price will vary depending on the destination. If it is within the city, the price will be less and outside the city, the price will be more. All these specifications can be mentioned using the data property of the classes in ontology domain. The system can retrieve the appropriate result according to the query input.

Our research objectives are become different since the purpose of this ontology is to facilitate information retrieval for user travel planning in a city or in an entire country, and, particularly, to help the passenger to decide which itinerary and what mode of transport is preferred to reach its intended destination based on the ONTC services such as differences of ticket rates for journey of day time and night time, long and short distance.

Another similar work Shen et al. (2011) proposed a prototype about semantic network based route planner that assists the users in finding the walking route in Kanazawa City. The aim of research is to analyze the users' willing by their free text input and try to understand what they need, then generate a proper route for users to traverse all the places which are conform to the their thoughts.

This work tried to suppose a method to solve the problem that how we can understand the users' willing if the will input freely in a sentence or a phrase as search query. This work emphasized the semantic analysis for geographic information, for instance, where users want to go or what kind of place they want to travel, such as the historical buildings, museum etc. Various data such like history, culture asset, architecture, are related to the buildings, temples, shrines etc. And these data are collected and classified, then be converted to metadata saved in database as search source for semantic analysis. The relation among the places will be stored in the ontology file for relation searching.

In this study, the system allow users to input their search query freely, the system can fuzzy analyze the requirement automatically, and use ontology file to do the semantic analysis. However, the decompressor named "CJK Analyze" is utilized to split the input query into a word with two letters. Despite that, this decompressor can expand the scope of fuzzy search, but it cannot provide a more adequate word segmentation method for Japanese.

In addition, users cannot manually set the starting point and the destination, the system only contrives to set the point (pre-defined place) as starting point.

The main goal of this system is to assists the users in finding the walking route based on the users' willing while we are trying to provide different transportation route alternatives based on the user start and destination points by overcoming the lacks of semantic level inquiry.

# 3. GEOGRAPHIC INFORMATION STANDARDS

The collection, storage, management, and analysis of geospatial data are critical components of many business activities. Unfortunately, geospatial data can and have been stored in a number of ways that may not be readily accessible and usable, or easily shared with, or reported to others. The countermeasure is standardization. There are many factors that drive the need and desirability for standardization. The widespread use of geographic information is creating a need for standards. Consistent and accessible information, and associated systems and services contribute to making life simpler and increase the reliability and effectiveness of the goods and services we use (ADSIC, 2009). From the GIS perspective, the key benefits of standards are:

- Data sharing/communications: standardized data is more easily understood; standardization allows countries and agencies to exchange data freely when based upon a common set of definitions and understandings about the data and its features.
- Elimination of redundancy - standardization contributes directly to the reduction and in some cases the elimination of duplication and redundant gathering of similar data and database construction.
- With GIS standards as a major component, technology, data policy, and institutional framework are four necessary components for the realization of a spatial data infrastructure.

In order to take full advantage of rich sources of geo-information it is necessary to process and archive data products and to make them widely available. To streamline and optimize the dissemination, national and international Geographic Information Standards are being established so called Spatial Data Infrastructures (SDIs) (van Oosterom, 2005). Since the first international conference on “Emerging Global Spatial Data Infrastructure” held in Bonn in 1996, the concept of SDI has rapidly experienced a rising awareness in the geographic information community (Sun et al., 2009).

### 3.1 Spatial Data Infrastructure (SDI)

The term SDI is often used to denote the relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of and access to spatial data. The SDI provides a basis for spatial data discovery, evaluation, and application for users and providers within all levels of government, the commercial sector, the non-profit sector, and academia and by citizens in general. SDI is an initiative intended to create an environment that will enable a wide variety of users to access, retrieve and disseminate spatial data and information in an easy and secure way. However, different institutes, governments, and situations vary the definition of SDI. Wytzisk and Sliwinski (2004) had given a more broader and flexible definition of SDI as “a multi-leveled, scalable, and adaptable collection of technical and human services, which are interconnected across system, organizational, and administrative boundaries via standardized interfaces”. Those services enable users from different application domains to participate in value chains by gaining a seamless access to spatial information and geo-processing resources.

The components of SDI include the following, (1) Data are distributed among heterogeneous sources. (2) Access Network builds a basic network to access data from different sources, for example, the Internet. (3) In order to cooperate and reuse these data on the access network, Standards have to be followed for interoperability. (4) Policy is to build, maintain, access and apply standards and datasets (e.g. broker or catalogue for storing, managing and discovering the information of datasets) for (5) People to transact, process and make decisions. Moreover, the three components (access network, policy, and standards) are very dynamic due to the rapidity with which technology develops and the need for mediation of rights, restrictions and responsibilities between people and data change. The components of SDI in the era of web service would be also applied, but a little difference. First, access to network would be the Internet. Second, data would be changed by web services or GIS services, which would be reusable to provide data accessing services or data processing services. Third, Standards would be created and maintained by international consortium, for instance, International Organization for Standardization (ISO) or OGC (Open Geospatial Consortium) (Sun et al., 2009). The main components of SDIs are geographic information services providing access to geospatial data and geoprocessing capabilities. Acknowledging the role of interfaces for distributed computing in the GIS area, the OGC identified the interface of software components as the key ingredient of GIS technology that needed standardization in order to achieve interoperability. The result is a series of syntactic interface specifications, establishing protocols for components exchanging geospatial information (Klien, 2006). Also, ISO Technical Committee Number 211 (ISO/TC 211) Geographic Information/Geomatics, which is responsible for the ISO geographic information series of standards. These standards may specify, for geographic information, methods, tools and services for data management (including definition and description), acquiring, processing, analyzing, accessing, presenting and transferring such data in digital/electronic form between different users, systems and locations.

Fourth, Policy would be replaced by catalogue services to store, manage and discover the suitable datasets. Spatial data collected by various public agencies, private sector, non-profit organizations and academia are available in different scales of reference – local, national, regional, global. SDI's are also developed on different political and administrative levels. A kind of hierarchy is formed (Fig. 3.1). There are projects of SDI's from local or even corporate levels up to regional and global ones. The levels are inter-connected.

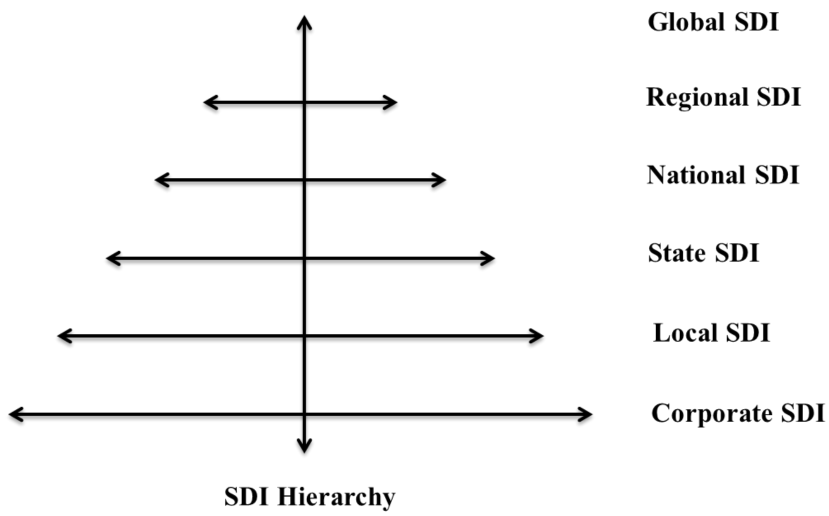



**Figure 3.1 :** SDI hierarchy (Sudra, 2010, p. 22).

The international (regional/global) level is very specific since on this level countries cooperate with each other in different fields where the spatial data is involved, including business and economic development, solving social problems, environmental management, regional/global mapping and others. One of the most prominent initiatives of international cooperation is the project of the European Commission of the European Union to develop a Geographic Information Infrastructure on the European level, known as INSPIRE. INSPIRE aims at making available relevant, harmonized and quality geographic information for the purpose of formulation, implementation, monitoring and evaluation of Community policy-making (Lemmens, 2006; Sun et al., 2009).

### 3.2 INSPIRE Directive on Spatial Data Infrastructure

Infrastructure for Spatial Information in the European Community (INSPIRE) (Fig. 3.2) has been established under the Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 which entered into force on 15 May 2007. INSPIRE Directive is a Framework Directive which lays down general rules to establish a European Union spatial infrastructure which will enable the sharing of environmental spatial information among public sector organizations and better facilitate public access to spatial information across Europe (INSPIRE Directive). The potential users of geo-information provided through INSPIRE include governmental agencies and public administration, utility and public services, commercial and professional users, research and educational institutions, nongovernmental and non-profit organizations and individual citizens.


**Figure 3.2 :** INSPIRE logo.

The primary motivation for establishment of a pan-European SDI is interoperability and shared use of spatial data and services between public authorities for the performance of public tasks, in particular in the context of environmental policies.

Several major problems with interoperability of geoinformation among member countries of the European Union can be identified. These are:

- a variety of data formats (maps in digital and analogue forms, satellite imagery and aerial photographs, databases, reports, statistical data), file formats and encodings
- lack of harmonization of spatial reference systems
- differences in data models and data semantics (understanding of terms, definitions), vocabularies and registries
- lack of consistency of map scales, dataset resolutions, and data representation levels
- differences in data quality (accuracy, precision, temporal aspects)
- geographical gaps in coverage of particular datasets with a lot of data duplication at the same time
- a variety of portrayal rules, symbologies
- insufficient documentation of data leading to problems with data identification and interpretation; differences in metadata standards
- incompatible information systems.

Normally data collected by various organizations are processed and stored locally within the organizations. When it comes to data sharing, it made the situation challenging due to the difficulties in data identification, access and manipulation of available data. This isolation of data causes gaps in availability, lack of harmonization between datasets at different geographical scales and duplication of information collection. Awareness is growing at national and at EU level about the need for quality geo-referenced information to support understanding of the complexity and interactions between human activities and environmental pressures and impacts. The INSPIRE initiative is therefore timely and relevant but also a major challenge given the general situation outlined above and the many stakeholder interests to be addressed (Weerasinghe, 2009).

The vision of INSPIRE regarding geographic information is based on several common principles:

- Data should be collected once and maintained at the level where this can be done most effectively.

- It should be possible to combine seamlessly spatial data from different sources and share it between many users and applications.
- Spatial data should be collected at one level of government and shared between all levels.
- Spatial data needed for good governance should be available on conditions that are not restricting its extensive use.
- It should be easy to discover which spatial data is available, to evaluate its fitness for purpose and to know which conditions apply for its use.
  (INSPIRE Directive)

INSPIRE is to be based on the infrastructures for geographic information established and operated by the Member States of the European Union. This complies with the concept of distributed infrastructure and the hierarchical structure of SDI's. INSPIRE does not require particular countries to collect new spatial data on particular themes. As such, even data harmonization is not an obligation. This means that data models and standards at the national level do not need to change due to INSPIRE nevertheless INSPIRE requirements and recommendations should be adhered to when the data is served from public registries at the European level.

The scope of the INSPIRE Directive are "spatial data held by or on behalf of a public authority operating down to the lowest level of government when laws or regulations require their collection or dissemination" (Recital (12); Article 4.6).

The INSPIRE Directive defines thematic scope of the spatial data that shall be standardized. In total INSPIRE covers 34 Spatial Data Themes defined in three Annexes.

Annex I

1. Coordinate reference systems
2. Geographical grid systems
3. Geographical names
4. Administrative units
5. Addresses
6. Cadastral parcels
7. Transport networks

8. Hydrography
9. Protected sites

Annex II

1. Elevation
2. Land cover
3. Ortho-imagery
4. Geology

Annex III

1. Statistical units
2. Buildings
3. Soil
4. Land use
5. Human health and safety
6. Utility and governmental services
7. Environmental monitoring facilities
8. Production and industrial facilities
9. Agricultural and aquaculture facilities
10. Population distribution –demography
11. Area management/restriction/regulation zones & reporting units
12. Natural risk zones
13. Atmospheric conditions
14. Meteorological geographical features
15. Oceanographic geographical features
16. Sea regions
17. Bio-geographical regions
18. Habitats and biotopes
19. Species distribution
20. Energy resources
21. Mineral resources

The development of INSPIRE Implementing Rules for harmonized spatial data follows a two-step approach. The first step was the development of a conceptual framework and specification methodology laid down in the following documents:

- Definition of Annex Themes and Scope (Url-4)
- Generic Conceptual Model (Url-5)
- Methodology for the development of data specifications (Url-6)
- Guidelines for the encoding of the spatial data (Url-7).

The second step comprises the development of Data Specifications for each data theme listed in the Annexes of the Directive. Other, supportive, documents include: INSPIRE Feature Concept Dictionary, INSPIRE Glossary, INSPIRE Consolidated UML Model, GML Application Schemas, INSPIRE Code List Dictionaries.

Harmonized spatial data specifications shall be developed for the data referring to all three Annexes. According to the INSPIRE Technical Architecture: "Only spatial data sets that conform to data specifications for Annex themes that are adopted INSPIRE Implementing Rules will be considered fully integrated in the infrastructure" (Url-1). However, the adoption dates of the implementation rules for data themes defined in particular Annexes are different. The data from Annex I has the priority, as it can be considered basic spatial reference data. Data Specifications (Guidelines) for particular Annex I data themes have been made available in September 2009. Furthermore, the 'Draft Commission Regulation implementing Directive 2007/2/EC of the European Parliament and of the Council as regards interoperability of spatial data sets and services' has been made available in December 2009. This document includes legally based implementing rules for INSPIRE-compliant spatial data. INSPIRE data specifications are not intended to cover all kinds of data requirements. Member States are encouraged to re-use the INSPIRE data specifications for their own usage; extend spatial object types and add new properties, specify additional constraints applicable to the own data sets and re-use of INSPIRE objects to spatially enable application data (Portele, 2009). In this research, the transportation spatial data corresponding to the INSPIRE Annex I Transport Network data theme has been implemented according to the needs of public transportation dataset.

## 3.3 INSPIRE Data Specification on Transport Networks

INSPIRE Directive defines the spatial data theme Transport Networks as: “Road, rail, air and water transport networks and related infrastructure. Includes links between different networks. Also includes the trans-European transport network as defined in Decision 1692/96/EC of the European Parliament and of the Council of 23 July 1996 on Community guidelines for the development of the trans-European transport network and future revisions of that Decision.” (INSPIRE Data Specification on Transport Networks).

The data specification on Transport networks provides:

- the basis for the development of the part of the Implementing Rules, defined in the Article 7(1) of the INSPIRE Directive, related to the spatial data theme Transport networks and
- the implementation guidelines that will accompany the Implementing Rule on the Interoperability of Spatial Data Sets and Services according to Article 7(1) of the INSPIRE Directive.

The scope of the INSPIRE Transport Networks Data Product Specification incorporates five distinct transport themes:

- Road transport
- Rail transport
- Water transport
- Air transport
- Cableways

This is summarized in the below figure 3.3 which shows how the Directive guides the Generic Conceptual Model (GNM) which contains a basic framework for any kind of network model. From the GNM a Common Transport application schema is adapted and this then is used as the basis for all five application schema. Collectively these are drawn together in this document as the Transport Networks Data Product Specification (INSPIRE Data Specification on Transport Networks).

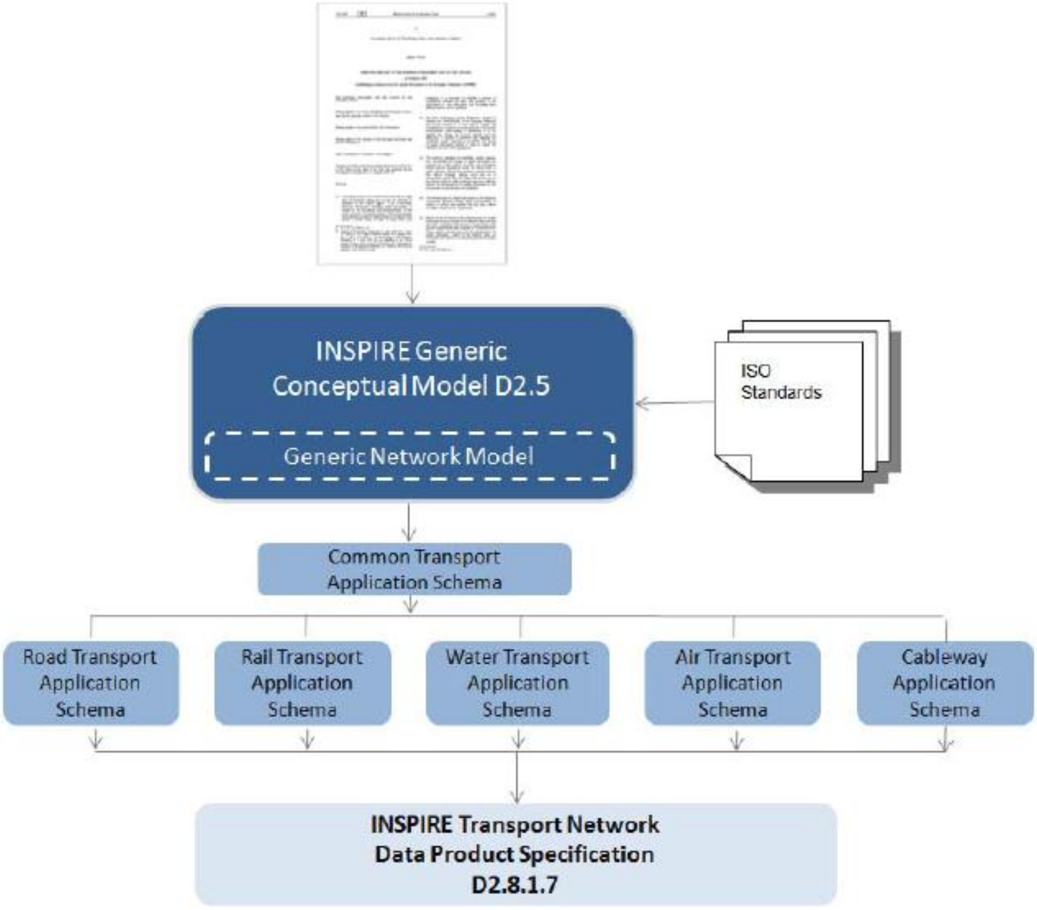



**Figure 3.3 :** Overview of the Transport Networks structure and dependencies in INSPIRE Data Specification (INSPIRE Data Specification on Transport Networks, p. 6).

The data specification has been prepared by the INSPIRE Thematic Working Group Transport Networks (TWG-TN), a multinational team of experts in the field drawn from different parts of the European Union, in the frame of the common and transparent development process.

The transport component should comprise an integrated transport network, and related features, that are seamless within each national border. In accordance with article 10.2 of the INSPIRE Directive, national transport networks may also be seamless at European level, i.e. connected at national borders. Transportation data includes topographic features related to transport by road, rail, water, and air. It is important that the features form networks where appropriate, and that links between different networks are established, i.e. multi-modal nodes, especially at the local level, in order to satisfy the requirements for intelligent transport systems such as location based services (LBS) and telematics. The transport network should also support the referencing of transport flow to enable our navigation services.

The data specification is extensive, covering major transport networks types that are defined in the five distinct transport themes (sub-themes): Road, Rail, Water, Air transport and Cableways, including the connections between those types (figure 3.4). The sub-themes are defined in a way that they can be used together to support an integrated approach to transport and they may be used with other spatial data themes. It is evident that there are a very large number of applications that can potentially use the Transport networks.

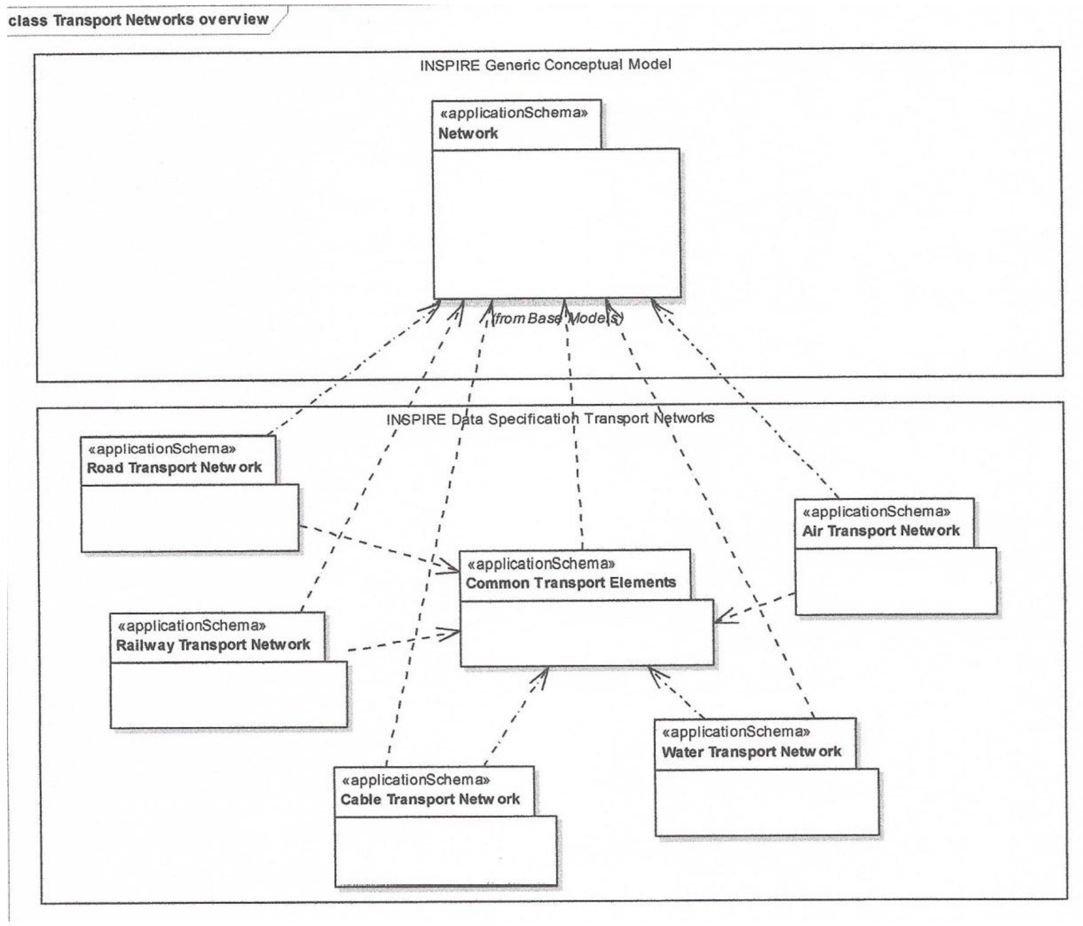



**Figure 3.4 :** UML class diagram: overview of the transport networks UML structure and dependencies (INSPIRE Data Specification on Transport Networks, p. 13).

Taking into account the variety of responsibilities in collecting, managing and using the data and different approaches in the data base management practice, from simple models to complex data arrangements, this data specification is provided as basic framework and with the purpose to maximize the reuse and sharing of the data about a network. It is mainly focused on the “widely reused – widely referenced” segments of spatial objects, supporting the loose linkage between the diverse organizational data with these spatial objects and allowing the extensibility to fit into diverse applications and user’s needs.

This approach provides a framework for users to configure and associate their own information (from surface condition surveys, to journey planning, to trans-European transport policymaking etc.) using existing transport networks information in each Member State. All the spatial data sub-themes are based on the INSPIRE Generic Conceptual Model that relies on several ISO 19100 series of geographic information standards to provide the foundations for specific aspects of interoperability.

Specific mechanisms, used in the data specification and defined in the GNM, include:

- Network connection mechanism to establish the cross-border connectivity (a simple cross-referencing system to establish cross-border connections between the transport networks) or to establish intermodal connectivity (by linking two transport network elements from different transport networks which use a different mode of transport);

- Object referencing to support the reuse of information (for example to avoid the duplication of the geometry and to link complementary feature types from different organizations);

- Linear referencing to support and link the different transport properties to the transport elements – it is used to position phenomena along a linear object, using a distance from the beginning of the linear object and

- The mechanism to combine the network elements into high-level semantic meanings.

The elements in the network are handled as nodes, links, aggregated links, areas and points. In addition, the individual transport links can be combined to form transport link sequences or further – the combination of both can be used to form the transport link sets (Fig. 3.5).

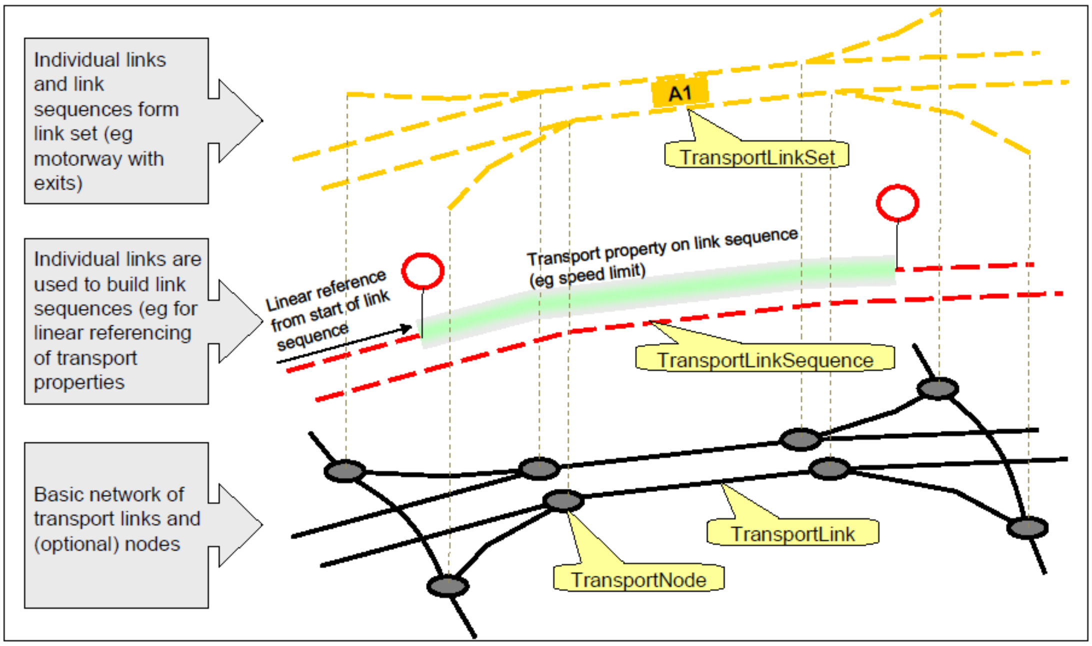



**Figure 3.5 :** Example of the use of link, node, link sequence and link set (INSPIRE Data Specification on Transport Networks, p. 24).

The data specification includes three types of geometry: (a) (topographic) area objects, (b) centerline objects and (c) point objects. The types (a) and (b) may be alternative representations of the same real world phenomena about which the user can associate their own information (objects). The type (c) is, apart from network nodes, only included in the specification for marker posts. The basic spatial representation type is 2D vector.

The purpose of this document is to specify a harmonized data specification for the spatial data theme Transport Networks as defined in Annex I of the INSPIRE Directive. However, each organization will have different responsibilities from the next and this will influence the kind of data they collect and manage and use. In turn some organizations may use simple models; others by necessity will have complex data arrangements. This data specification is provided as basic framework, which users can adopt and if required – extend for themselves.

### 3.3.1 Application schema of road transport network

The Road Transport Networks application schema (Roads Schema) employs a link and node structure to represent a road system used for the transportation of vehicles in the form of a linear network. The Roads Schema inherits classes from the Common Transport Schema and also creates its own classes to describe properties of the road network such as Ownership and traffic direction that can apply to whole sections of the network element or subsections that can be described using linear referencing. Figure 3.6 below shows the example of use of elements forming the road transport network. Unified Modeling Language (UML) class diagram of overview of the road transport networks enumerations and road transport networks are illustrated in figure 3.7 and 3.8.

The primary aspects modeled for road network elements are:

- Spatial: Geometric (point, line and area (topographic)) representation of various elements that are parts of a network. Typically, the network is handled as a network of connected linear elements (links) with points (nodes) at the ends of the lines (at junctions, road ends etc.). Also real objects with a function in a network may be represented in the dataset. Network connectivity within the roads network is essential but between elements in the other networks is an optional spatial aspect.
- Temporal: All elements in a network may have a temporal validity (i.e. description of when the network element exists in the real world) and optional information on when data was entered, modified or deleted in the dataset.
- Thematic: The road schema can be thematically displayed via several of the attributes defined within the specification such as owner authority or speed limits.

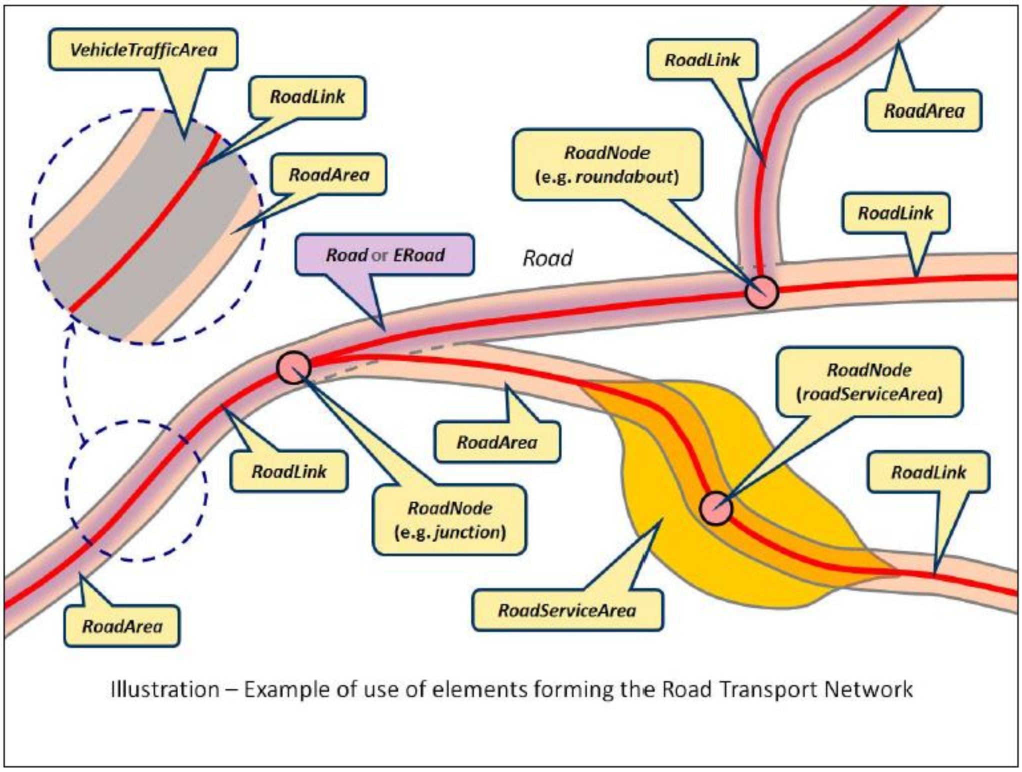



**Figure 3.6 :** Example of use of elements forming the road transport network (INSPIRE Data Specification on Transport Networks, p. 46).

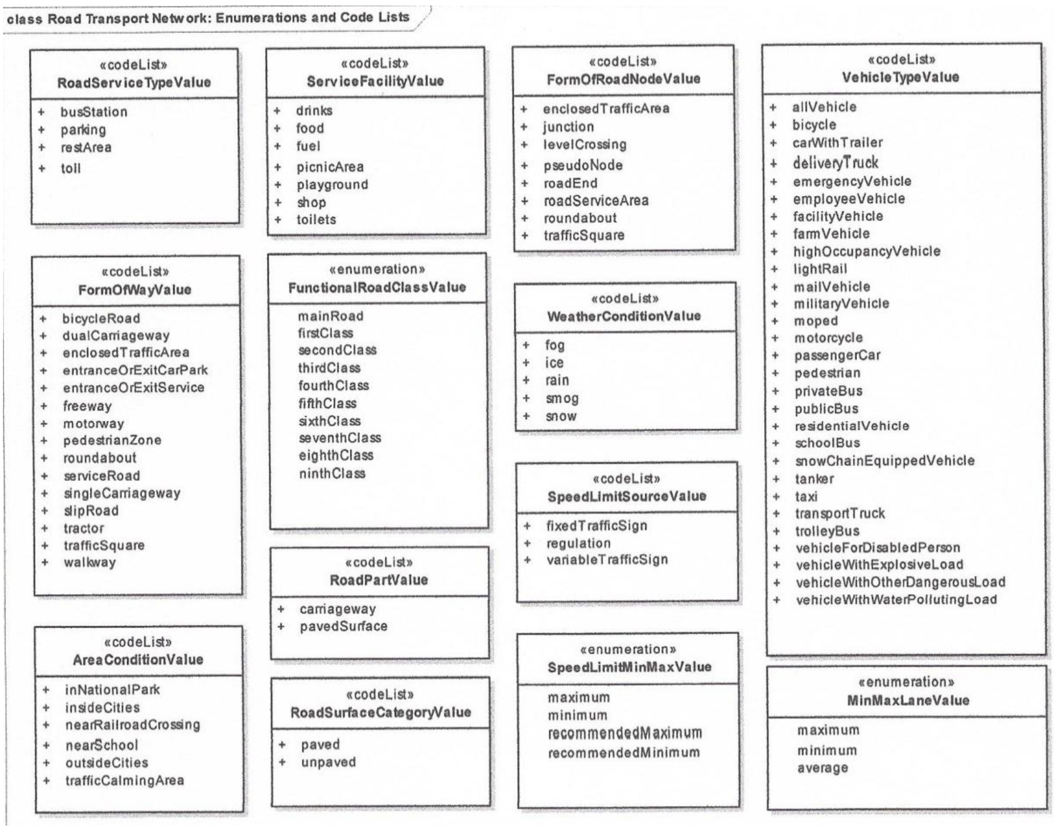



**Figure 3.7 :** UML class diagram: road transport network enumerations-code list (INSPIRE Data Specification on Transport Networks, p. 53).

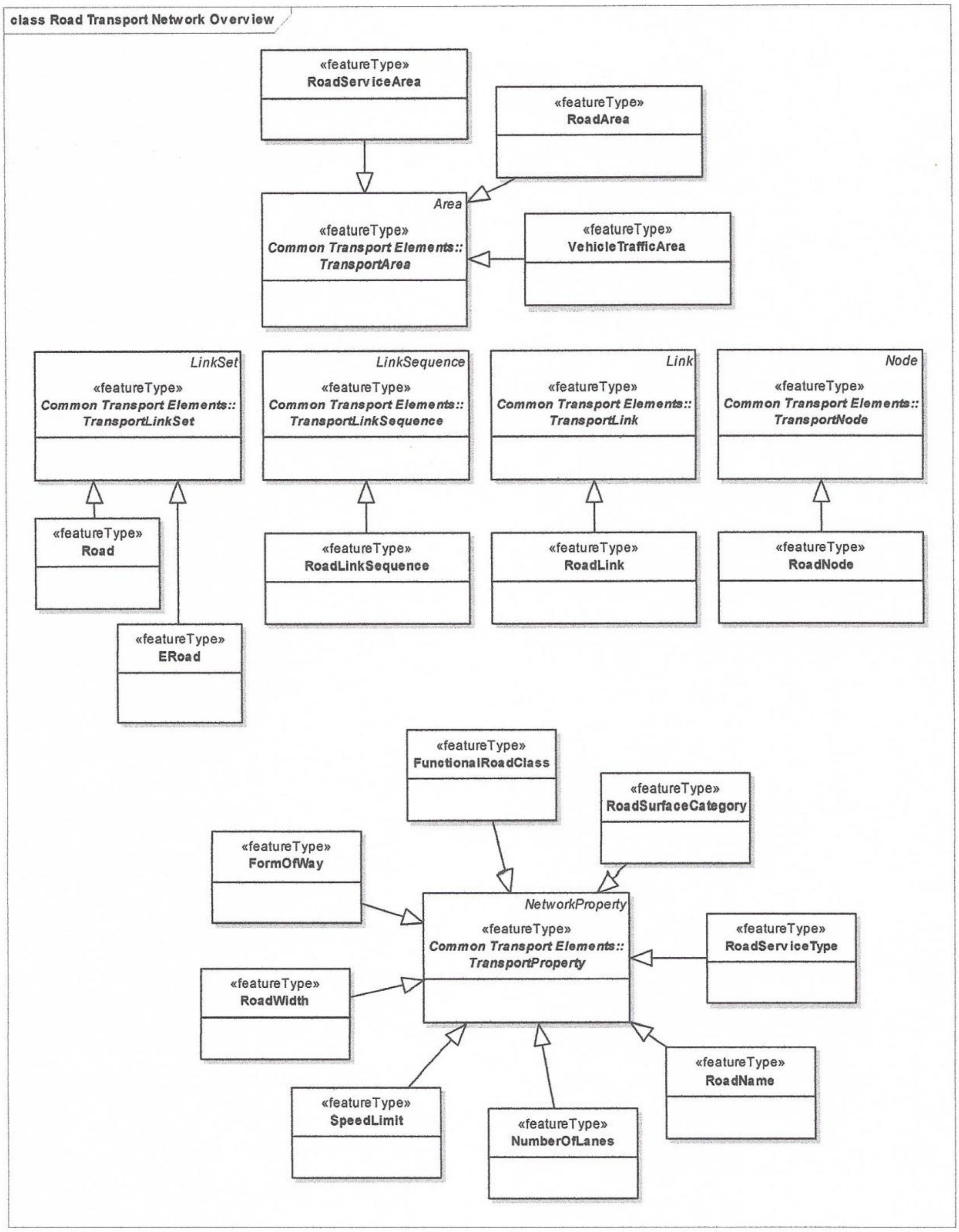



**Figure 3.8 :** UML class diagram: Overview of the road transport networks (INSPIRE Data Specification on Transport Networks, p. 41).

### 3.3.2 Application schema of water transport network

The Water transport network schema employs a link and node structure to represent navigable waterways and ports within water bodies (main marine waterways and ships routing systems – traffic separation schemes - at the sea, and inland waterways defined across rivers and canals) allowing transportation in the form of a linear network (figure 3.9 and 3.10). The water transport network reuse the centerline geometry of the Hydrography theme's network (hydrographic network) of rivers and canals as the basis for most inland water transport. However, when this is not enough to describe it properly, specific water transport network's centerlines can also be used (waterway links), in combination to the previous ones, in order to complete the network. The water schema also inherits classes from the Common Transport Schema and also creates its own classes to describe properties of the water network such as water traffic flow direction and any restrictions that can apply to whole sections of the network element or subsections that can be described using linear referencing. UML class diagram of overview of the water transport networks application schema and water transport networks enumerations and code list are illustrated in figure 3.11 and 3.12.

- Spatial: Geometric (point, line and area (topographic)) representation of various elements that are parts of a network. Typically, the network is handled as a network of connected linear elements (links) with points (nodes) at the ends of the lines (at junctions, ports etc.). Also real objects with a function in a network may be represented in the dataset. Network connectivity within the water network is essential but between elements in the other networks is an optional spatial aspect.

- Temporal: All elements in a network may have a temporal validity (i.e. description of when the network element exists in the real world) and also optional information on when data was entered, modified or deleted in the dataset.

- Thematic: The water schema can be thematically displayed via several of the attributes defined within the specification such as CEMTClass.

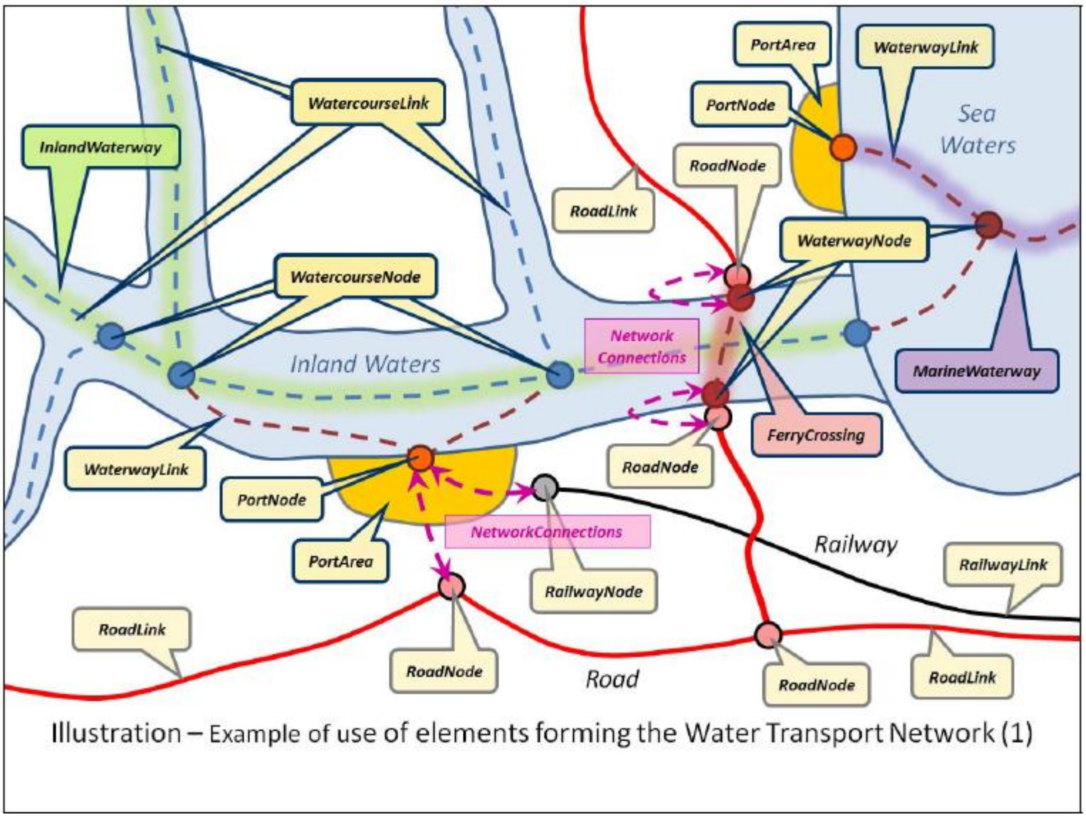



**Figure 3.9 :** Example of use of forming the water transport network (1) (INSPIRE Data Specification on Transport Networks, p. 93).

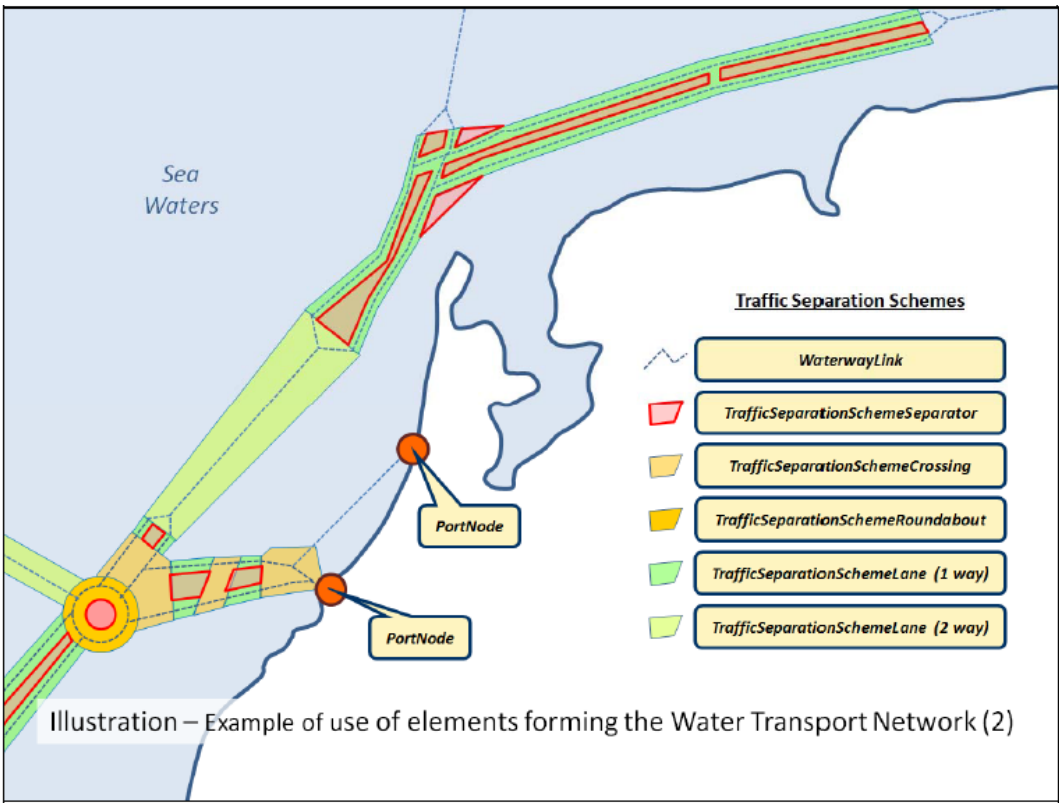



**Figure 3.10 :** Example of use of forming the water transport network (2) (INSPIRE Data Specification on Transport Networks, p. 93).

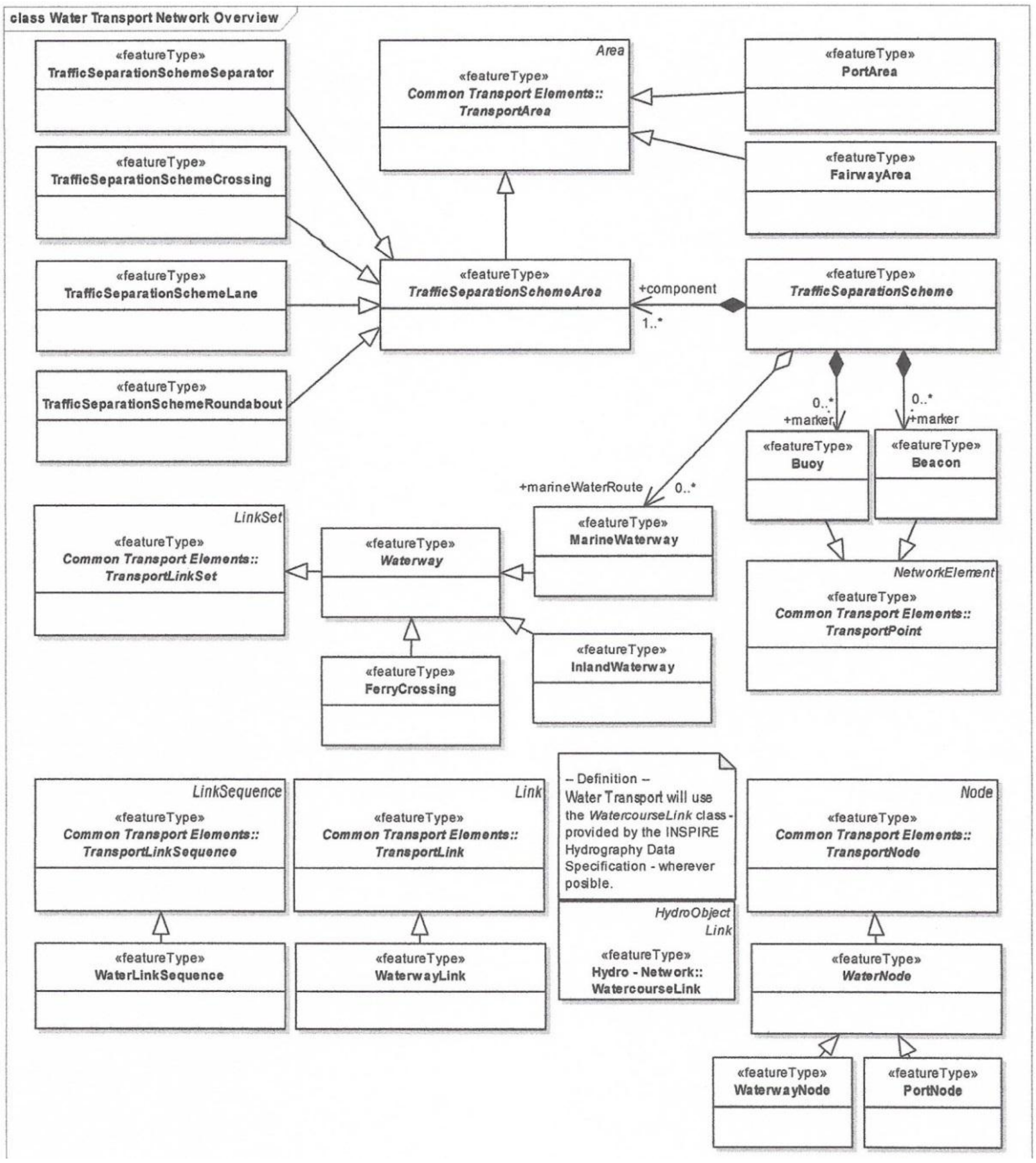



**Figure 3.11 :** UML class diagram: overview of the water transport networks application schema (INSPIRE Data Specification on Transport Networks, p. 87).

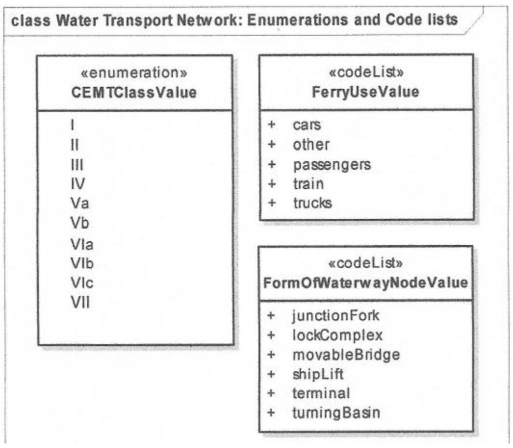



**Figure 3.12 :** UML class diagram: water transport network enumerations-code list (INSPIRE Data Specification on Transport Networks, p. 100).

# 4. GEOSPATIAL WEB SERVICES FOR SPATIAL DATA RETRIEVAL

Spatial data infrastructure (SDI) is currently being set up with regions, countries, and across national borders to facilitate management and access to geospatial information. Through geospatial web services, the components of Geographic Information Systems are decomposed and become accessible via SDI (Gone and Schade, 2008).

Web Services give us a means of interoperability between different software applications running on a variety of platforms. Web Services support interoperable machine-to-machine interaction over a network. Every Web Service has an interface described in a machine-readable format. Web Service interfaces are described in a standardized way by using Web Service Description Language (WSDL). WSDL files define input and output properties of any service and services' protocol bindings. WSDL files are written as Extensible Markup Language (XML) documents. WSDL is used for describing and locating Web Services. Web Services are defined by the four major elements of WSDL, "portType", "message", "types" and "binding". Element portType defines the operations provided by the Web Services and the messages involved for these operations. Element message defines the data elements of the operations. Element types are data types used by the Web Service. Element binding defines the communication protocols. Other systems interact with the Web Service in a manner as described in WSDL using Simple Object Access Protocol (SOAP) messages. SOAP is an XML based message protocol for exchanging the information in distributed environment. It provides standard packaging structure for transporting XML documents over a variety of network transport protocols. It is made up of three different parts. These are the envelope, the encoding rules and the Remote Procedure Call (RPC) convention. SOAP can be used in combination with some other protocols such as Hypertext Transfer Protocol (HTTP) (Sayar et al., 2005). These concepts work under the framework of Service Oriented Architecture (SOA), as represented in the Figure 4.1.

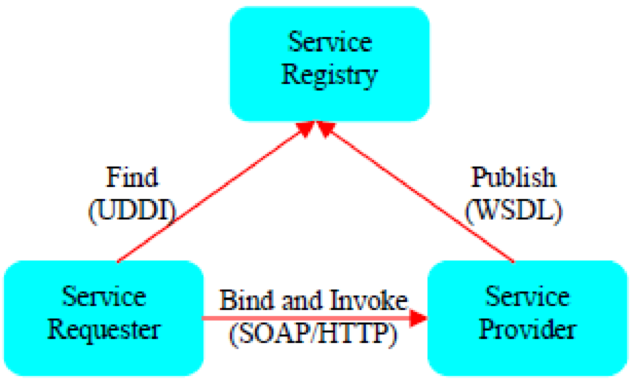



**Figure 4.1 :** OGC Web Services Architecture diagram
(Sahin and Gumusay, 2008, p. 627).

SOA concept has three components (Figure 4.1): service provider, service registry, service requester and three operations: publish, find, and bind. A SOA relates the roles of the three components with the three operations to maintain automated discovery and the use of services. The three essential components mentioned above can be described as following:

- Service provider publishes services to a registry and makes it available on the Internet for the requests of the consumers.
- Service requester (client) performs service discovery operations on the service registry in order to find the needed service; then accesses services.
- Service registry helps service providers and service requesters to find each other by acting as a registry of the services (Sahin and Gumusay, 2008).

After Web Services gain momentum and wide acceptance, entities working on GIS started to involve in Web Services area and tried to integrate and/or convert their GIS servers and applications into web services.

### 4.1 Geospatial Web Services

In GIS community, Geospatial Web Services have been introduced to overcome spatial non-interoperability problem. Geospatial Web services have been developed with the goal of sharing geospatial data and services among heterogeneous geospatial processing systems (Amirian and Alesheikh, 2008). Generally, a geospatial web services can be viewed as a modular Web application that provides services on geospatial data, information, or knowledge. It refers to the use of Web services technologies to manage, analyze, and distribute spatial information.

As a web service, a geospatial web service involves three actors: user entity (consumer), provider entity (provider) and register entity (broker). Basic operations during the life cycle of a geospatial web services include publication, discovery, binding, invoking and execution. The interfaces of some geospatial web services have been standardized. The most important players for such standardization efforts are ISO/TC 211 and OGC. Because of the influential specifications from OGC, geospatial web services have an add-on unique geospatial flavor on the message communicating other than those popular W3C Web services standards such as WSDL, SOAP and Universal Description, Discovery and Integration (UDDI). Geospatial web services focus on standard interface directly through HTTP protocol, which is now recognized as more related to representation state transfer (REST). Further details about the OGC standards will be explained in following section (Zhao et al., 2007).

## 4.2 Open Geospatial Consortium (OGC) Web Services

The Open Geospatial Consortium (OGC) is an international industry consortium of 483 companies, government agencies and universities participating in a consensus process to develop publicly available interface standards. OGC Standards support interoperable solutions that "geo-enable" the Web, wireless and location-based services and mainstream IT. The standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications. OGC’s specifications are harmonized with the ISO 19100 standards series under development by ISO/TC211 (International Organization for Standardization, Technical Committee 211, and Geographic information/Geomatics).

Within the broader context of Web Services, OGC Web Services (OWS) represent an evolutionary, standards-based framework that enable seamless integration of a variety of online geoprocessing and location services. OWS allows distributed geoprocessing systems to communicate with each other across the Web using familiar technologies such as XML and HTTP. OGC Web Services provide a vendor-neutral, interoperable framework for web-based discovery, access, integration, analysis, exploitation and visualization of multiple online geodata sources, sensor-derived information, and geoprocessing capabilities (Url-3). OGC Web Services framework consists of interface implementation specification and encodings which are openly available to be implemented by developers. The interface implementation specifications are software technology neutral details about various operations of each geospatial Web service. The encodings provide the standard glue among different parts of geospatial Web services. Each service of this framework can be implemented using various software technologies and systems (Amirian and Alesheikh, 2008). The followings are some examples of the OGC's members which are well-known: Google, ESRI, Microsoft, ORACLE, NASA, IBM, MIT and so on. In the GIS world, especially in the area of interoperable web mapping, OGC web services are very meaningful because they provide the standardization and opportunities for users to share geospatial information even though data is from different sources and/or different applications. Technically, a dataset contained in a local information system, is implemented as a web service to be usable (allowing requests) over the web. The software executing this implementation respects at least one OGC/ISO standard (Pornsamutsin, 2010).

The most important OGC specifications are:

- GML - Geography Markup Language
- WFS - Web Feature Service
- WMS - Web Map Service
- WCS - Web Coverage Service
- CS-W - Catalog Service Web
- SFS - Simple Features - SQL

The OGC Web Services (OWS) suite includes three principal types of georeferenced information access services: Web Map Server (WMS), Web Coverage Server WCS), and Web Feature Server (WFS). In addition, there are services such as GeoParser and GeoCoder that return spatially referenced results. Figure 4.2 is an architecture diagram showing conceptually how some of the OGC Web Services are related, and naming some (not all) of the operations they define (Web Map Service Implementation Specification).

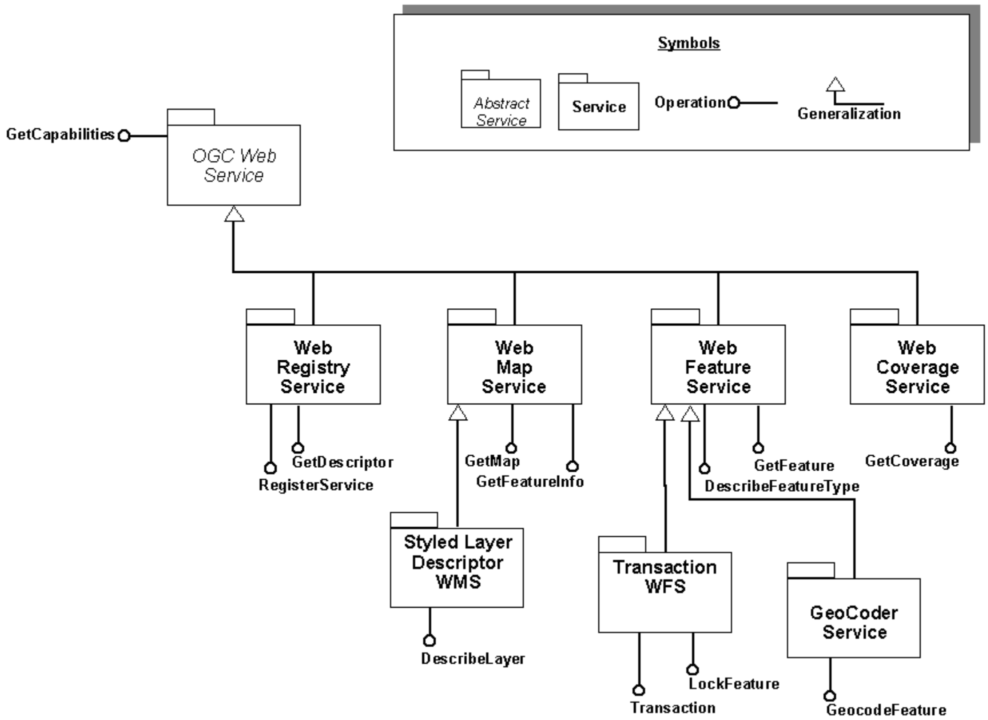



**Figure 4.2 :** OGC Web Services Architecture diagram
(Web Map Service Implementation Specification, p. 3).

### 4.2.1 Web map services (WMS)

Web Map Service (WMS) Implementation Specification is one of the most well-known Implementation Specifications by OGC. This standard specifies operations to retrieve a description of the maps offered by a service instance, to retrieve a map, and to query a server about features displayed on a map.

A Web Map Service produces maps of georeferenced data. We define a "map" as a visual representation of geodata; a map is not the data itself. These maps are generally rendered in a pictorial format such as PNG, GIF or JPEG, or occasionally as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats. This specification standardizes the way in which maps are requested by clients and the way that servers describe their data holdings. The OpenGIS Web Map Service Interface defines three operations, the first two of which are required of every WMS (figure 4.3).

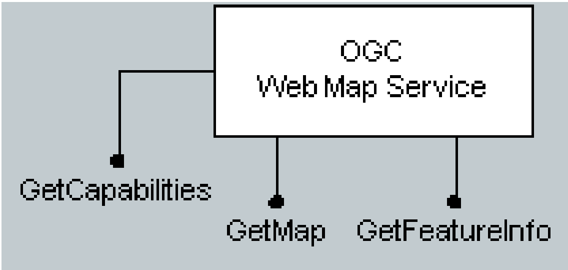



**Figure 4.3 :** The OpenGIS Web Map Service Interface
(OpenGIS WMS Cookbook, p. 10).

**GetCapabilities (required):** Obtain service-level metadata, which is a machine-readable (and human-readable) description of the WMS's information content and acceptable request parameters.

**GetMap (required):** Obtain a map image whose geospatial and dimensional parameters are well defined.

**GetFeatureInfo (optional):** Ask for information about particular features shown on a map.

A standard web browser can ask a Web Map Service to perform these operations simply by submitting requests in the form of Uniform Resource Locators (URLs). The content of such URLs depends on which of the tasks is requested. All URLs include a specification version number and a request type parameter.

In addition, when invoking GetMap a WMS Client can specify the information to be shown on the map (one or more "Layers"), possibly the "Styles" of those Layers, what portion of the Earth is to be mapped (a "Bounding Box"), the projected or geographic coordinate reference system to be used (the "Spatial Reference System," or SRS), the desired output format, the output size (Width and Height), and background transparency and color.

When two or more maps are produced with the same Bounding Box, Spatial Reference System, and output size, the results can be accurately layered to produce a composite map. The use of image formats that support transparent backgrounds (e.g., GIF or PNG) allows the lower Layers to be visible. Furthermore, individual map Layers can be requested from different Servers. The WMS GetMap operation thus enables the creation of a network of distributed Map Servers from which Clients can build customized maps.

A particular WMS provider in a distributed WMS network need only be the steward of its own data collection. This stands in contrast to vertically integrated web mapping sites that gather in one place all of the data to be made accessible by their own private interface. Because each WMS is independent, a WMS must be able to provide a machine-readable description of its capabilities. This "Service Metadata" enables Clients to formulate valid requests and enables the construction of searchable catalogs that can direct clients to particular WMSs.

A WMS may optionally allow the GetFeatureInfo operation. When invoking GetFeatureInfo the Client indicates, what map is being queried and which location on the map is of interest. If it does, its maps are said to be "queryable," and a Client can request information about features on a map by adding to the map URL additional parameters specifying a location (as an X, Y offset from the upper left corner) and the number of nearby features about which to return information.

**Cascading map servers:** A "Cascading Map Server" is a WMS that behaves like a client of other WMSs and behaves like a WMS to other clients. For example, a Cascading Map Server can aggregate the contents of several distinct map servers into one service. Furthermore, a Cascading Map Server can perform additional functions such as output format conversion or coordinate transformation on behalf of other servers.

**Styled layer descriptors:** The behavior of a Web Map Service can be extended to allow user-defined symbolization of feature data instead of named Layers and Styles. The Styled Layer Descriptor (SLD) specification describes this extension. This specification applies to a Web Map Service that publishes its ability to produce maps rather than its ability to access specific data holdings. A basic WMS classifies its georeferenced information holdings into "Layers" and offers a finite number of predefined "Styles" in which to display those layers. In brief, an SLD-enabled WMS retrieves features from a Web Feature Service and applies explicit styling information provided by the user in order to render a map (Web Map Service Implementation Specification; OpenGIS WMS Cookbook).

### 4.2.2 Web feature services (WFS)

The OGC Web Map Service allows a client to overlay map images for display served from multiple Web Map Services on the Internet. In a similar fashion, the OGC Web

Feature Service allows a client to retrieve and update geospatial data encoded in Geography Markup Language (GML) from multiple Web Feature Services. Besides operations to retrieve data, WFS describes operations to edit (insert, update, delete) data on the level of a feature. With WFS (requests for) transactions on geographic features can directly be sent to a single database. This makes WFS the only open, interoperable standard with which data can be edited over the Internet (Web Feature Service Implementation Specification, p. 10).

The requirements for a Web Feature Service are:

1) The interfaces must be defined in XML.
2) GML must be used to express features within the interface.
3) At a minimum a WFS must be able to present features using GML.
4) The predicate or filter language will be defined in XML and be derived from CQL as defined in the OpenGIS Catalogue Interface Implementation Specification.
5) The datastore used to store geographic features should be opaque to client applications and their only view of the data should be through the WFS interface.
6) The use of a subset of XPath expressions for referencing properties.

In the specification, the Web Feature Service Interface is described as illustrated in figure 4.4.

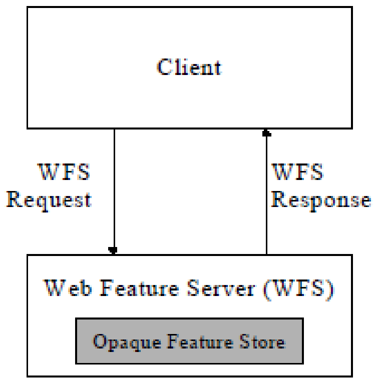



**Figure 4.4 :** The OpenGIS Web Feature Service Interface
(Web Feature Service Implementation Specification, p. 10).

The specification describes the WFS operations. WFS supports INSERT, UPDATE, DELETE (in a Transaction-request) and QUERY (GetFeature) operations on geographic features using HTTP as the distributed computing platform. It also provides operations to describe the service and the contents of the service, such as GetCapabilities and DescribeFeatureType (Brentjens, 2004).

Following is a typical scenario which describes the use of the WFS operations and the interaction between a client and a Web Feature Service. Figure 4.5 shows the WFS Interaction Steps: Client's interaction with WFS usually starts with a discovery step which involves retrieving the capabilities document. After this the client may request details about a certain feature by issuing a DescribeFeatureType request. However the most common queries used are GetFeature requests to retrieve particular features. displays these steps (Aydın, 2007).

1) GetCapabilities: The clients (Web Map Server or users) start with requesting a capabilities document from WFS. When a GetCapabilities request arrives, the server may choose to dynamically create a capabilities document and returns this, or simply return a previously created xml document.

2) DescribeFeatureType: After the client receives the capabilities document he/she can request a more detailed description for any of the features listed in the WFS capabilities document. The WFS returns an XML Schema that describes the requested feature as the response.

3) GetFeature: The client may then ask WFS to return a particular portion of any feature data. GetFeature requests contain some property names of the feature and a Filter element to describe the query. The WFS extracts the query and bounding box from the filter and queries the related database(s) that holds the actual features. The results obtained from the DB query are converted to that particular feature's GML format and returned to the client as a FeatureCollection object.

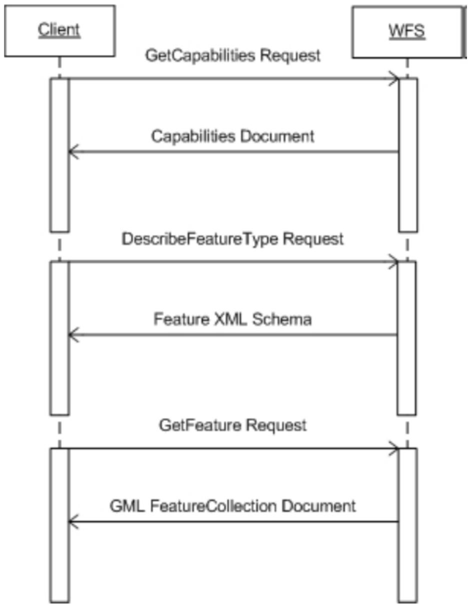



**Figure 4.5 :** WFS interaction steps (Aydın, 2007, p. 53).

### 4.2.3 Web coverage services (WCS)

The Web Coverage Service (WCS) supports electronic interchange of geospatial data as "coverage" - that is, digital geospatial information representing space-varying phenomena. The WCS provides available data together with their detailed descriptions; allows complex queries against these data; and returns data with its original semantics (instead of pictures), which can be interpreted, extrapolated, etc. and not just portrayed. Unlike WFS, which returns discrete geospatial features, the Web Coverage Service returns representations of space-varying phenomena that relate a spatio-temporal domain to a (possibly multidimensional) range of properties. The WCS consists of three operations: GetCapabilities, GetCoverage, and DescribeCoverageType. The GetCapabilities operation returns an XML document describing the service and the data collections from which clients may request coverages. The GetCoverage operation of a WCS is normally run after GetCapabilities has determined what queries are allowed and what data are available. The GetCoverage operation returns values or properties of geographic locations, bundled in a well-known coverage format. In fact, any format is allowed, but the specification provides five formats (GeoTIFF, HDF-EOS, DTED, NITF and GML) of which the server should at least support one. The GetCoverage syntax and semantics bear some resemblance to the WMS GetMap and WFS GetFeature requests, but several extensions support the retrieval of coverage rather than static maps or discrete features (Brentjens, 2004).

# 5. SEMANTIC WEB

Today's Web content is huge and not well-suited for human consumption. An alternative approach is to represent Web content in a form that is more easily machine-processable by using intelligent techniques. The machine processable Web is called the Semantic Web (Ganapathy and Sagayaraj, 2011). The Semantic Web aims for machine-understandable resources, whose information can then be shared and processed both by automated tools, such as search engines, and by human users. Resources can be classic text documents, domain specific data (such as geo-data), software applications, etc.

Semantic Web provides an ontology-based framework that allows data to be shared and reused across model, application and even community boundaries. An ontology is a collection of key concepts and their inter relationships collectively providing an abstract view of an application domain. It defines a common vocabulary across the system and forms the conceptual basis for integration. To make sure that different agents have a common understanding of these terms, one needs ontologies in which these terms are described, and which thus establish a joint terminology between the agents. Basically, an ontology is a collection of definitions of concepts and the shared understanding comes from the fact that all the agents interpret the concepts of the same ontology (Prasanna and Bakshi, 2010; Baader et al., 2003).

## 5.1 Geospatial Semantic Web

The idea of a geospatial semantic web is to capture, analyze, and tailor geospatial information beyond the purely lexical and syntactic levels of data search. Similar to other semantic web efforts, realization of a geospatial semantic web needs to (1) develop spatial ontologies for different domains and identify semantic distinctions and relations among ontological specifications; (2) represent the identified semantic concepts for computer processing and for human understanding; and (3) the processing of geospatial queries against these ontological commitments and evaluate retrieval results based on the match between the semantics of the expressed information need and the available semantics of the information resources and search systems (Jung and Sun, 2010).

Semantic technology concepts enable information systems to perform intelligent search. An intelligent approach could improve the efficiency and accuracy of data retrieval. It empowers machines to understand and response the human queries based on their meaning and it makes searching more effective. Ontology is the backbone of semantic web and using ontology as a knowledge base improves the efficiency of data retrieval. Retrieval of data and information is the vital part of any system. Spatial Semantic web approach is vital in any kind of transport system because an intelligent approach will improve the query result (Sherimon et al., 2011). Therefore, semantic approach is implemented for the purpose of build intelligent public transportation geoportal in this research study.

Main activities of the Semantic Web initiative include:

- Development of ontology representation languages such as OWL
- Development of tools for ontology building and ontology mapping
- Semantic markup of resources, including web services
- Development of knowledge-based reasoner
- Application and demonstration development for information retrieval and reasoning

### 5.2 Ontology as Knowledge Representation

A key role is played by Ontologies for the Semantic Applications. The term ontology has its origin in philosophy. The modern analytical approach to ontologies concerns basic categories and terms like ‘things’, ‘attributes’ or ‘events’. An important question is that how these different categories are related to each other, and how they interact. An ontology can be understood in this context “as a machine readable vocabulary that is specified with enough precision to allow differing terms to be precisely related” (Bode, 2006). In ontologies, the semantic description of data, i.e., the logical relationships between data elements and other formal statements are made explicit through ontology languages such as OWL (Ontology Web Language). This not only makes it easier for other human users of models to understand the specifically intended meaning, but also means that other tools can use the definitions transparently (Malik et al., 2010).

Ontology is an outcome of the conceptual analysis of a given problem area (domain). It represents simulated reality by formal specification in such a way so the possibilities of its sharing exceeds the group of closely interested subjects. Ontologies are used to share domain information and include computer usable definitions of basic concepts in the domain and the relationships among them. They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable. The development of ontologies is the first step towards the automatic reasoning based on knowledge since the definition of its elements (concepts, relations among concepts, axioms and the rest) allows knowledge that is not explicitly indicated, to be inferred (Samper et al., 2006).

The outcome of ontological analysis is mostly conceptual hierarchy which capture the most crucial entities of respective domain, their attributes and generally valid relations among them. The most important observed relations are requisite relation among properties and the whole-part relation among entities included in the respective concept (Číhalová et al., 2008). The main focus of ontologies are the classes, which are describing the concepts in the domain. A knowledge base covers classes and instances of classes. For example, a class of wines represents all wines. Specific wines are instances of this class. The Bordeaux wine you can order in a restaurant is an instance of the class of Bordeaux wines. A class can have subclasses which represent more specific concepts than the superclass. For example, we can divide the class of all wines into red, white, and ros´e wines. Alternatively, we can divide a class of all wines into sparkling and non-sparkling wines. Depending on the used logic-based language the ontology author can define very granular differences between the classes (Bode, 2006).

Slots describe properties of classes and instances: Château Lafite Rothschild Pauillac wine has a full body; it is produced by the Château Lafite Rothschild winery. We have two slots describing the wine in this example: the slot body with the value full and the slot maker with the value Château Lafite Rothschild winery. At the class level, we can say that instances of the class Wine will have slots describing their flavor, body, sugar level, the maker of the wine and so on (Noy and McGuinness, 2001).

All instances of the class Wine, and its subclass Pauillac, have a slot maker the value of which is an instance of the class Winery shown in figure 5.1 as an example.

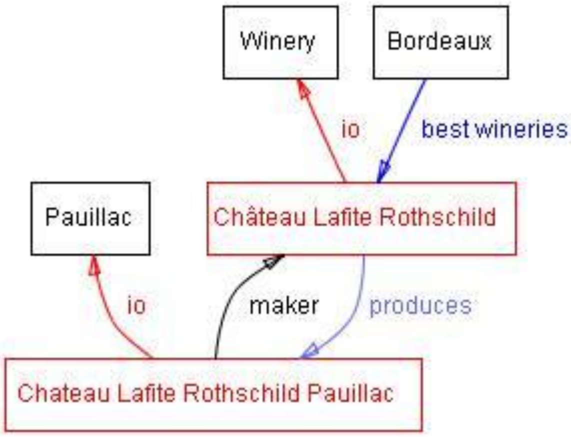



**Figure 5.1 :** Wine Ontology (Noy and McGuinness, 2001, p. 4).

An ontology consists of a formal description of classes (general things), relations between them and properties (of these things). This formal description is normally written in a logic-based language like RDF or OWL, so that "detailed, accurate, sound, and meaningful distinctions can be made among the classes, properties, and relations". The used logic-based language and definitions are described in following sections 5.2.2 and 5.2.3. The ontology language for the semantic web OWL provides means to describe entities of an application domain in an ontology in a well-structured way. The underlying formalism for OWL are Description Logics (DLs) (Turhan, 2011).

### 5.2.1 Description logics

The formal underpinnings of Description Logics (DL) are a cornerstone of Semantic Web technology such as the Web Ontology Language (OWL), which are a family of knowledge representation formalisms that have formal semantics. This family of logics is tailored towards representing terminological knowledge of an application domain in a structured and formally well-understood way.

The other important feature of DL is that providing reasoning service, which allows one to infer implicitly, represented knowledge from the knowledge that is explicitly contained in the knowledge base. DL reasoners can help build and maintain sharable ontologies by revealing inconsistencies, hidden dependencies, redundancies, and misclassifications (Turhan, 2011).

A knowledge representation based on Description Logics provides facilities to set up knowledge bases, to reason about their content.  A DL knowledge base consists of two components, a set of terminological axioms called TBOX and a set of assertional axioms called ABOX. The TBOX introduces the terminology (vocabulary of an application domain), and the ABOX contains assertions about named individuals in terms of this vocabulary. The vocabulary contains concepts, which represent sets of individuals, and roles, which indicate binary relationships between individuals (Yilma, 2006).

These concepts (unary predicates) and roles (binary predicates) then restrict the way these classes and relations are interpreted. Based on these definitions, implicitly captured knowledge can be inferred from the given descriptions of concepts and roles. These inferences are defined based on the formal semantics of DLs. A great range of these inferences has been defined and investigated for a variety of DLs with differing expressivity, where the expressivity of a DL is determined by the means it allows describing concepts and roles. The inferences devised for DLs are, for instance, sub-class or instance relationships (Turhan, 2011).

The knowledge base contains information expressed in explicit way. Using it, the knowledge base can obtain implicit, new information. The kind of reasoning we can do using Description Logics is:

- Reasoning with the terminology (TBox): TBox contains the concepts that conform the vocabulary of the application in a given domain
- Reasoning with the assertions (ABox): ABox contains statements about individuals using the terms declared in the vocabulary. ABox introduces the assertional knowledge (world description).

The idea is to have the ability of reasoning that make possible to infer implicitly represented knowledge from the explicitly knowledge that is contained in the knowledge base (Leon, 2006).

While the formal terrain of DLs is now well mapped out and covered by efficient algorithms, a new generation of end-user tools is necessary to put this technology into the spotlight of industrial routine. Some of the technologies explored and refined by these tools can be applied to DL. DL ontologies are potentially very complex and large. An ontology editor should simplify navigation and help users to maintain large ontologies through mechanisms.

A good example of such a tool is Protege, a knowledge-modeling platform developed at Stanford Medical Informatics with support from a community of thousands of users over almost two decades, which can be used to define classes and properties, to edit logical class expressions, to invoke reasoners, and to link ontologies into the Semantic Web (Knublauch et al., 2004).

Recent work within the Semantic Web community has resulted in W3C recommendations of two knowledge representation languages, RDF and OWL. Most knowledge representation languages such as RDF schema and OWL have their foundation on Description Logic. These formalisms are represented by First Order logic. DL and these derived mark-up languages is a sort of defacto standard for representing ontologies based on their well-founded semantics and available reasoning services.

### 5.2.2 Resource description framework (RDF)

The RDF is a language for representing information about resources in the World Wide Web. RDF is a framework for representing information about resources in a graph form. Since it was primarily intended for representing metadata about WWW resources, it is built around resources with URI. RDF is a data model that can be represented in XML to describe objects and relationships between them. It has a set of triples: property, subject, and object. RDF has formal semantics that provide a dependable basis for reasoning about the meaning of an RDF expression (Zhang et al., 2008). An example of a triple is shown in the figure 5.2 below. It says that "Joe Smith has homepage http://www.example.org/~joe". All elements of this triple are resources defined by URI.

The first resource http://www.example.org/~joe/contact.rdf#joesmith (subject) is intended to identify Joe Smith. Note that it precisely defines how to get to a RDF document as well as how to get the joesmith RDF node in it. The second resource http://xmlns.com/foaf/0.1/homepage (predicate) is the predicate homepage from a FOAF (Friend-of-a-friend) vocabulary. The last resource (object) is Joe's homepage http://www.example.org/~joe/.

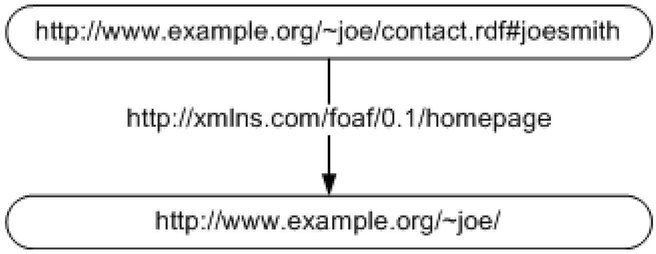



**Figure 5.2 :** RDF triple in graph representation (Url-2).

RDF uses XML syntax for exchanging its graphs, called RDF/XML. RDF Schema (RDFS) is a separate language, which can be used to specify a domain specific vocabulary for usage in RDF. It can be seen as a type system for RDF. RDF Schema provides constructs such as ‘Class’ and ‘subClassOf’ to establish such vocabulary and is considered to be an ontology language that enables a user to create a classification hierarchy with typing of properties (Lemmens, 2006).

For example, rdfs:subclassof property is used to build class hierarchies and rdfs:subpropertiesof used to allow properties to be specialized. The following RDF triple statement shows rdfs:subclassof property used to express that Grid_FeatureType is subclass of FeatureType (5.1).

<http://geoserver.itc.nl/aster/Ontology/EnvironmentalGeospatialServices.owl#Grid_FeatureType> <http://www.w3.org/2000/01/rdf-schema#subClassOf>

<http://geoserver.itc.nl/aster/Ontology/EnvironmentalGeospatialServices.owl#Feature_Type> **(5.1)**

**rdfs:Class:** A class defines a group of individuals that belong together because they share some properties. A class in RDF Schema corresponds to the generic concept of a Type or Category, in other languages. In RDF Schema, a class is any resource having an rdf:type property whose value is the resource rdfs:Class. Classes can be organized in a specialization hierarchy using subClassOf.

**rdfs:subClassOf:** Class hierarchies may be created by making one or more statements that a class is a subclass of another class. For example, Vector_FeatureType is subclass of FeatureType. Reasoners can thus infer an instance, which is Vector_FeatureType, is also a Feature_Type. The corresponding code is shown below (5.2).

<rdfs:Class rdf:about="Vector_FeatureType">

<rdfs:subClassOf>

<rdfs:Class rdf:ID="Feature_Type" />

</rdfs:subClassOf>

</rdfs:Class> **(5.2)**

**rdf:Property:** Properties can be used to state relationships between individuals or from individuals to data values. For Example: FeatureType class has a property hasAttribute, which relates it with the Attribute class; Attribute class has a property hasName, which relates it with an instance of the datatype: string.

**rdfs:domain:** A domain of a property limits the individuals to which the property can be applied.

**rdfs:range:** The range of a property limits the individuals that the property may have as its value (Yilma, 2006).

### 5.2.3 Web ontology language (OWL)

The OWL is another language for defining ontologies. OWL builds on the flexible approach to data representation of the RDF but goes beyond the basic semantics of RDF Schema (Zhang et al., 2008). Compared to RDF Schema, however, OWL has more facilities for expressing meaning and semantics. OWL extends the basic statements facility of RDF and the possibilities how classes and properties are defined in RDF Schema.

OWL can specify classes by using different logical operations like intersection, union or complement of other classes. Properties can also be defined and, if necessary, organized in a hierarchy of sub properties. In addition, domain and range restrictions can be declared. Domains are OWL classes. Ranges can be either OWL classes or externally defined data types such as String or Integer. Instances of classes can also be represented in OWL together with the values of their properties (Bode, 2006).

The most important extension compared to RDF and RDF Schema is the ability to define restrictions for properties or classes.

OWL comes in three different versions:

- OWL Lite has the lowest expressiveness. It is mainly designed for applications with simple taxonomic hierarchies and simple constraints.
- OWL DL has already the maximum expressiveness of OWL, "while retaining computational completeness and decidability". All language constructs of OWL can be used, but they underlie certain restrictions.
- OWL Full is designed "for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees".

### 5.2.3.1 Components of an OWL ontology

An OWL ontology consists of three components called Individuals, Properties, and Classes.

- Individuals also known as instances, represent objects in the domain we are interested in. They can be referred as being instances of classes.
- Properties are binary relations between individuals. Properties link two individuals of two classes. They are also known as roles in description logic.
- Classes are a concrete representation of concepts. OWL classes are sets that contain individuals.

**Subclass, disjoint and equivalence**

**SubclassOf:** Subclasses are specializations of their super classes. In OWL there is a built-in most general class named Thing that is the class of all individuals and is a super class of all OWL classes. An empty ontology contains the named class owl:Thing. All classes created are subclass of owl:Thing.

OWL allows general expressions to be used in axioms. "B is a subclass of A" written as $B^2$ A means "B implies A" emphasizes the meaning of subsumption, 'B is subsumed by A' written A ⊒ B. Subsumption in OWL means necessary implication.

**Disjoint:** In OWL, classes are assumed to be overlapping until disjointness axioms are entered. In order to separate the classes we must make them disjoint from one another. In this way we ensure that an individual which has been asserted to be a member of one class cannot be a member of another class.

**EquivalentClass:** Two classes C and D are said to be equivalent classes if all the statement that define C also defines D. Concept C is equivalent with concept D if C is subsumed by D and D is subsumed by C; i.e. C $\propto$ D means C $\supseteq$ D and D $\supseteq$ C

**Properties and restrictions**

**OWL properties:** OWL Properties represent relationships between two individuals of different classes. There are two main types of properties; Object properties and Data type properties.

- Object properties: Object properties link individuals of one class (which is domain to the property) to individuals of another class (which is range of the property).

  This can be expressed in DL as:

  ! R(a,b)Where R is a role, and a and b are individuals

  rdfs:range C(b) Where C is a class (which is the range of the property)

  rdfs:domain D(a)Where D is a class (which is the domain of the property)

- Datatype properties: Data type properties link an individual of one class to a value of an XML Schema Data type or rdf literal.

  The expression in DL is:

  ! R(a,b)Where R is a role, and a and b are individuals

  rdfs:range C(b) Where C is an XML Schema data type or rdf literal

  rdfs:domain D(a)Where D is a class (which is the domain of the property)

For example, Attribute class has a data type property, called hasName which associates its individuals to the XML data type: string. Both owl:ObjectProperty and owl:DatatypeProperty are subclasses of the RDF class rdf:Property.

**OWL restrictions:** The purpose of OWL is not just to create a concept hierarchy but to describe and define concepts. Restrictions on properties define the concepts clearly. OWL allows concepts to be defined by sets of necessary conditions.

In OWL properties are used to create restrictions, to restrict the individuals that belong to a class. There are three main categories of restrictions in OWL; Quantifier restrictions, Cardinality restrictions, and hasValue restrictions.

- Quantifier restrictions: use two quantifiers the existential quantifier and the universal quantifier. An existential restriction also known as ‘SomeValueFrom’ denoted by (∃) specifies the existence of at least one relationship along a given property to an individual that is a member of a specific class. A However, existential restrictions do not mandate that the only relationships for the given property that can exist must be to individuals that are members of the specified class. Universal restrictions also known as “AllValueFrom’ and denoted by (!), describes the set of individuals that, for a given property, only have relationships to other individuals that are members of a specific class.
- Cardinality restrictions: used to describe the class of individuals that have at least, at most or exactly a specified number of relationships with other individuals or datatype values.
    - Owl:minCardinality: A minimum cardinality restriction specifies the minimum number of relationships that an individual must participate in. The symbol used for the minimum cardinality restriction is “>”.
    - Owl:maxCardinality: A maximum cardinality restriction specifies the maximum number of relationships that the individual can participate. The symbol used for maximum cardinality restriction is “<”.
    - Owl:Cardinality: A cardinality restriction specifies the exact number of relationships that an individual participate and uses the equal sign “=”.
- hasValue restrictions: A hasValue restriction, denoted by the symbol , describes the set of individuals that have at least one relationship along the specified property to a specific individual.

**Domain and range:** In OWL domain and range are not constraints to be checked, rather they are axioms and can be used in reasoning (Yilma, 2006).

Once we have the model and structure of the ontology in a logic-based language like RDF or OWL, we can add the resources we decided as instances of the ontology and relate those using properties. Protégé Ontology Editor Software can be used to do these operations with a friendly graphical interface. Additionally Reasoning Engine like RacerPro Software needs to be used to realize reasoning operations on these ontologies and on the data themselves (Vila, 2011).

### 5.3 Role of Reasoning Engine in Ontology

One important part of the ontology management are the reasoners or inference engines, programs that generate deductions following logical rules, what allows us to deduce information from the ontology model (Vila, 2011). Geo-ontology reasoning and the query based on it are essential to GIS data sharing and interoperating. The function of reasoning is to discover potential relationship with the known relationship and acquire connotative knowledge with given knowledge by certain logic and rule. Ontology reasoning is a typical instance that the computer comprehends the ontology knowledge. Computer can interpret the information described by ontology and complete task intelligently (Wang et al., 2007).

The inference engine tries to derive answers from a knowledge base. This knowledge base is represented by a set of language specific rules of the ontology languages. These rules are then used to answer the queries, when the answer is not directly available. With the help of rules the answer can be inferred from available facts. For example, if the application requests the childrenOf an individual, but the working memory only contains parentOf relations, the inference engine can use the inverse property statements about childrenOf and parentOf to identify the correct response (Bode, 2006).

One of the main services offered by a reasoner is to test whether or not one class is a subclass of another class. By performing such tests on all of the classes in an ontology it is possible for a reasoner to compute the inferred ontology class hierarchy.

Another typical inference on concept expressions is concept satisfiability, which is the problem of checking whether a concept expression does not necessarily denote the empty concept (Leon, 2006).

Some of the services offered by the reasoner are:

- To derive new information on via automated inference (fundamental role)
- To ensure integrity of instances in a knowledge base

- To ensure the logical consistency of the ontology itself
- To provide reasoning across ontologies, and hence integrate data from multiple nodes
- To build and maintain sharable ontologies by revealing inconsistencies, hidden dependencies, redundancies, and misclassifications

The RacerPro system is particularly tailored for supporting this kind of applications, which mainly build on the exploitation of assertional reasoning (Abox reasoning). The main idea is that Aboxes are not static parts of the ontology, but are efficiently generated on the fly (referring to a shared Tbox, which is “processed” only once). Tboxes (ontologies) and Aboxes are maintained using the Racer-Pro server system, which communicates with remote application programs using well-defined axiom manipulation languages or entailment query languages.

The backbone of RacerPro is a description logic reasoner. It provides inference services for terminological knowledge as well as for representations of knowledge about individuals. Based on new optimization techniques and techniques that have been developed in the research field of description logics throughout the years, a mature architecture for typical-case reasoning tasks is provided. With its query and rule languages, RacerPro goes well beyond standard inference services provided by other OWL reasoners (Haarslev et al., 2011).

# 6. PROPOSED METHODOLOGY: SEMANTIC PUBLIC TRANSPORTATION GEOPORTAL

The methodology of this research firstly applied for the existing transportation dataset to harmonized based on the INSPIRE Transport Data Specification to meet the interoperability in European SDI Level. Data standards are typically represented by conceptual schemas on a high abstraction level, together with application schemas on lower levels of abstraction. They serve the purpose of creating interoperability at all levels (data format, data structure and data content) between the resources that comply with these standards. Application schema holds information about data structure and data content of the data set. In this regard, INSPIRE Transport Network Data Content Specification is used as an application schema to build the public transportation geodatabase during this research.

In addition to this, geospatial semantic method by means of ontologies is implemented public transportation dataset, which allows for automated reasoning and intelligent spatial queries during this research. Traditional GIS Systems are not able to handle intelligent queries (semantic) just by seeking of required data as defined in RDBMS, while the information sources are suffer from the semantic heterogeneity. In order to solve the problems of semantic heterogeneity, caused by synonyms – homonyms, ontology as a concept model is implemented in this research. Using ontology as a knowledge base improves the efficiency of data retrieval and overcome the deficiencies in the keyword-base traditional spatial query approach.

Lastly, Semantic spatial query algorithms are developed by the use of nRQL (new Racerpro Query Language) of RacerPro Reasoning Engine. This allows performing semantic queries in transportation ontology domain in order to retrieve transportation route alternatives considering the semantic heterogeneity.

The following sections explain the implemented in detail for this research.

### 6.1 Building Public Transportation Geodatabase Compliant With INSPIRE Transport Networks Data Content

In this thesis research, The Unified Modeling Language (UML) is an object-oriented schematic language with standardized meanings of symbols and relations is used to design Transport Network schema compliant with INSPIRE Data Specification. The concept of the UML is based on object-oriented technology, which means that it tries to identify objects of the real world and describe them as such, including their functionalities, data and attributes. One of the possibilities is to have UML-schemas translated into geodatabases. During this study, the designed UML schema is converted into the transportation geodatabase by a combination of Microsoft (MS) Visio CASE Tool and ArcCatalog from ESRI. The CASE tool will read the model and allow you to create the geodatabase schema and, optionally, generate code to define custom behavior.

The general strategy for using UML and CASE tools to design and create your geodatabase involves using UML to define the entire schema for the geodatabase, generating that schema, and then populating the schema with data is shown below figure 6.1.

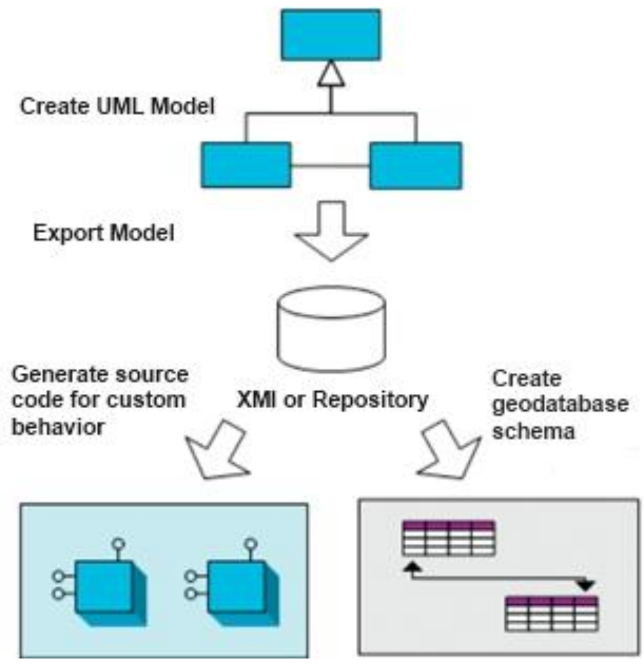



**Figure 6.1 :** The general strategy for using UML and CASE tools (Introduction to CASE tool, p. 2).

Transportation Feature Models of sample dataset are created in UML object models using the ESRI template classes diagram during this work. Then, designed Transportation Network Schema exported to a XMI file. XMI is a metadata format that enables interchange of data and metadata between tools, middleware and repositories. It links modeling-formats such as UML with metadata formats, such as XML. Directly following the UML to XMI conversion, it is necessary to check the XMI-file with the 'Semantics Checker'. This functionality checks the model for errors. If errors are found, a report is created, that points to the errors involved. When no errors are found, the model is semantically correct and ready to be converted to a geodatabase schema. Afterwards, ArcGIS Schema Generator Tool is used to read this XMI file and generate Transportation Network Geodatabase based on the UML design. The steps for accomplishing this are explained in the following sections.

### 6.1.1 Modelling of transport network geodatabase in UML

The design of the Geodatabase has been done using the UML formalisms, based on the object-oriented paradigm. In particular, in the case of the Geodatabase, several diagrams able to represent the hierarchical structure of the database are proposed. The term "hierarchical" means that the Geodatabase is constituted by "feature datasets" containing in turn "feature classes", the latter being characterized by attributes. All the attributes are set as public, in order to allow a total vision among the classes. The UML class notation and inheritance "is-a" generalization notation are described in figure 6.2, which is used during the design of transportation feature models.

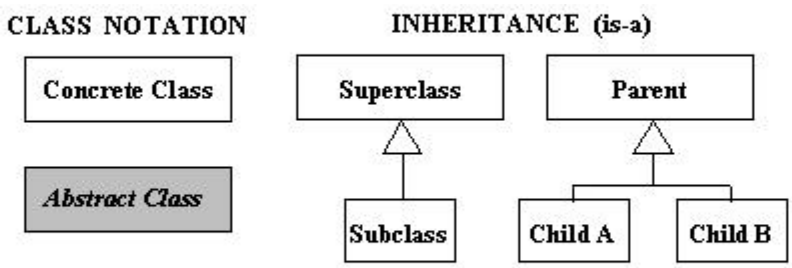



**Figure 6.2 :** UML class and inheritance notations.

Transportation Network geodatabase schema is designed as two feature dataset as "Administrative_Units", "Transport_Networks" UML package notation using MS Visio Interface (figure 6.3).

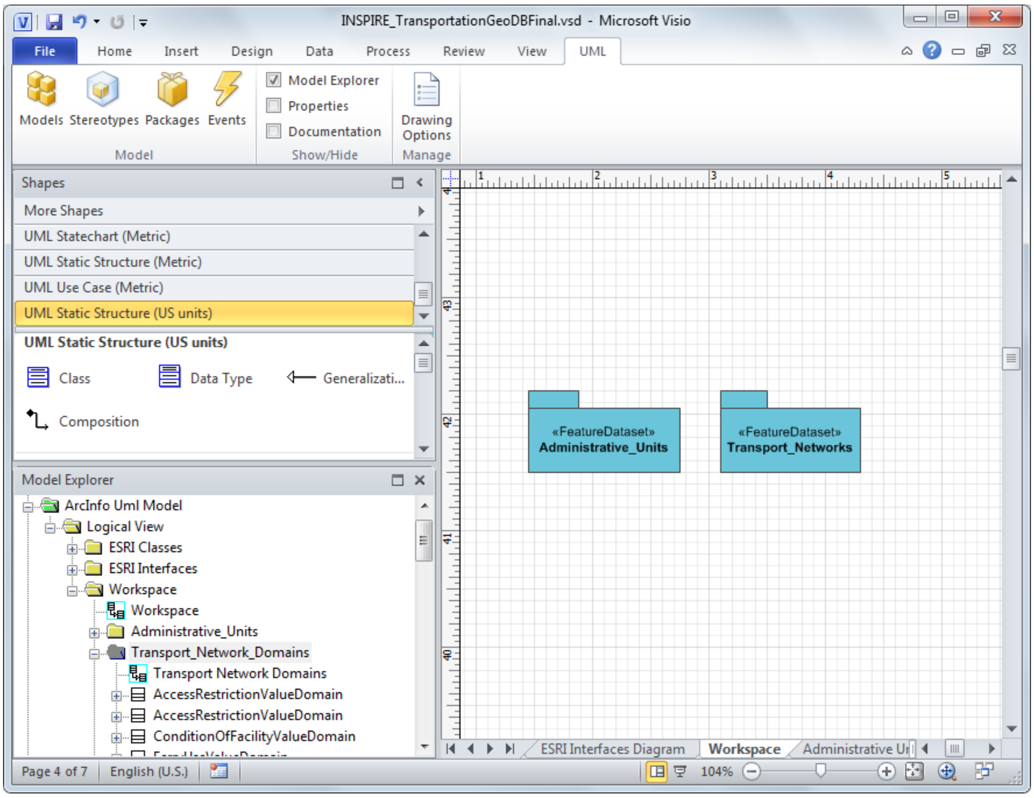



**Figure 6.3 :** Representation of the feature datasets as packages in MS Visio.

Inside the packages are present the UML classes, with the relevant attributes, modeling the feature classes. Some of the attributes are of common type (Integer, String, Date, etc.), while other refer to values belonging to defined domains. In order to properly model this case, stereotype domains are created as <<CodedValueDomain>> (numerical or textual) and refer the attributes of interest to these domains.

The Administrative Unit package contains the Province and District feature classes as shown figure 6.4. Transportation feature classes including, RoadLink, WaterwayLink, Road and Waterway Nodes and also Public Transportation Bus, SharedTaxi, Ferry Route and Stop classes are created inside of the Transport Network package.

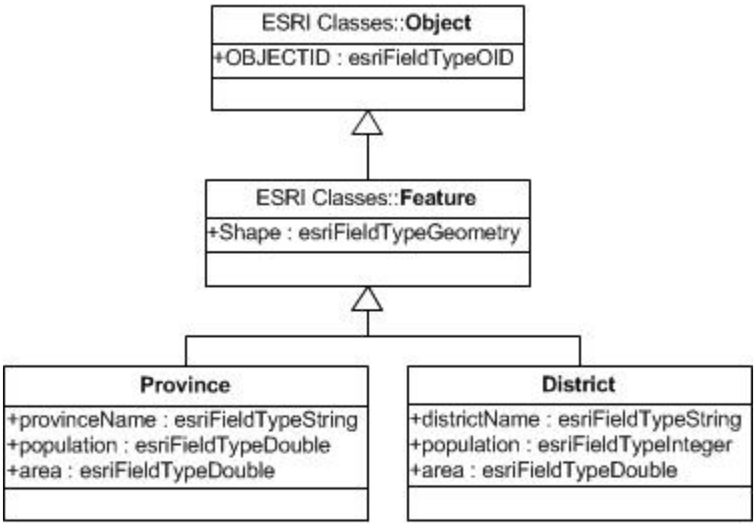



**Figure 6.4 :** Feature classes “Administrative_Unit - Province and District” UML diagram.

Road and WaterwayLink feature clases are derived from the CommonTransportLink UML object model as defined in the INSPIRE Transport Network Model. All the attribute and necessary information of these feature models are created in according to the INSPIRE Transport Network Data Specification. The designed UML classes of RoadLink and WaterwayLink feature models illustrated figure 6.5.

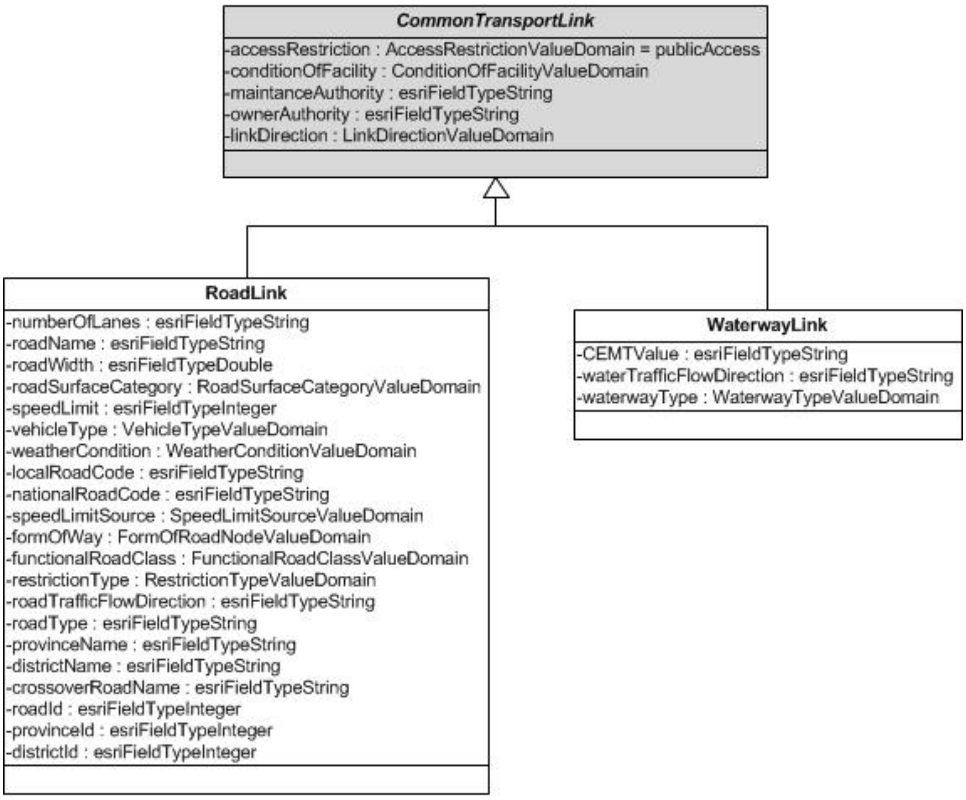



**Figure 6.5 :** Feature classes “Transport_Network Link - Road and Waterway Link” UML diagram.

Transportation Node feature classes are derived from the CommonTransportNode abstract class as Road Node and Water Node compliant with the INSPIRE. INSPIRE Transport Network Data Specification decribes the Water Node models as super class of Waterway Node and Port Node, so we do the same way in our geodatabase schema (figure 6.6.).

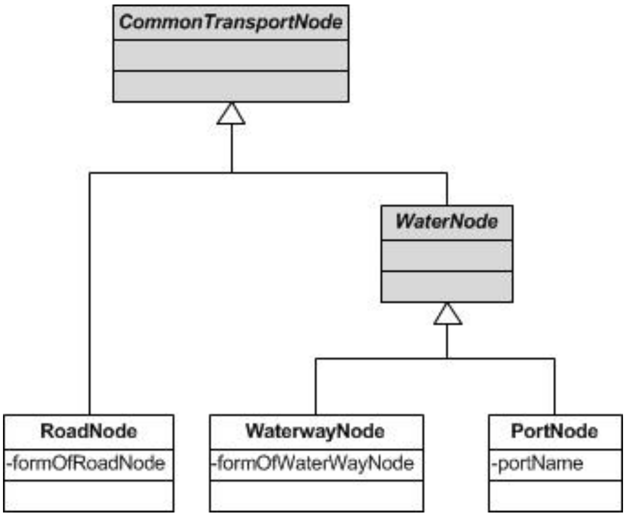



**Figure 6.6 :** Feature classes “Transport_Network - Road and Water Node” UML diagram.

Also, Public Transportation Route Feature Classes (Bus, SharedTaxi and Ferry) are derived from the Common TransportLink Sequence Abstract Class. According to definition of INSPIRE Transport Network Link Sequence Abstract Class: “A network element which represents a continuous path in the network without any branches. The element has a defined beginning and end and every position on the link sequence is identifiable with one single parameter such as length. A link sequence may represent a route” (D2.8.I.7.). Starting from this definition, Public Transportation Routes are designed as PublicBusLink, PublicSharedTaxiLink and PublicFerryLink Classes derived from the Common TransportLink Sequence Class in our UML Transportation Schema (figure 6.7).

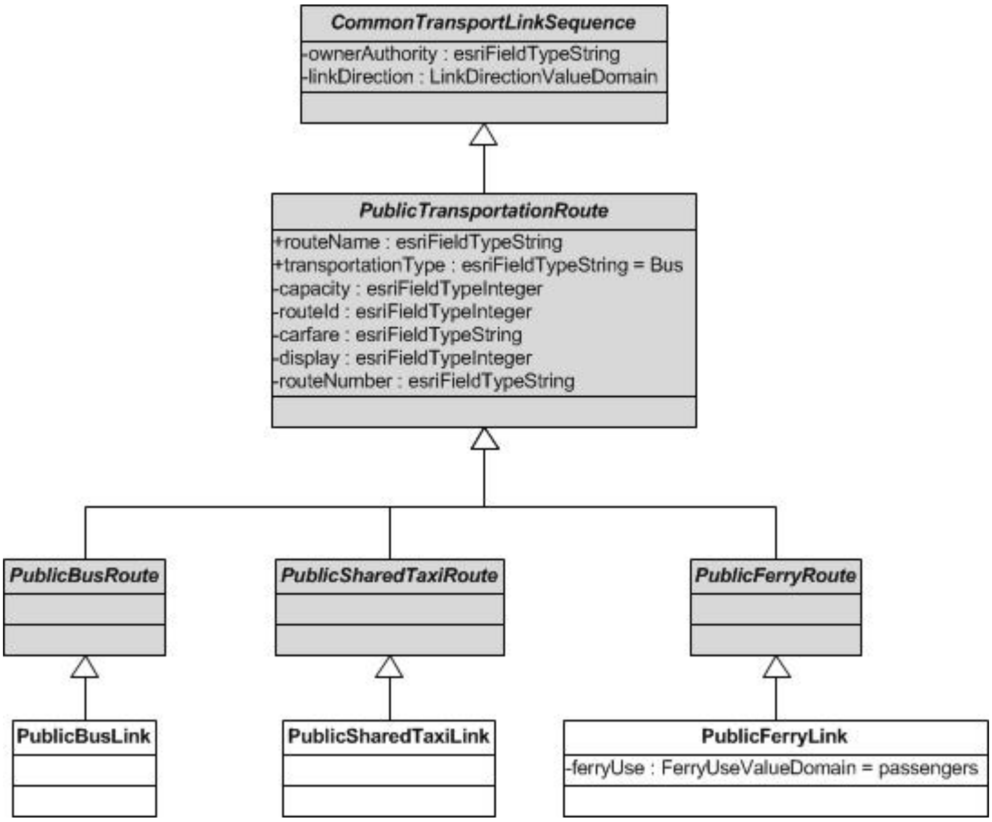



**Figure 6.7 :** Feature classes “Transport_Network – Public Transportation Route” UML diagram.

Transportation Stops for Bus, SharedTaxi and Ferry are derived from the CommonTransportPoint Class as referred to the INSPIRE Transport Network Data Specification as well. The UML feature class models of Transportation Stops are shown below figure 6.8.

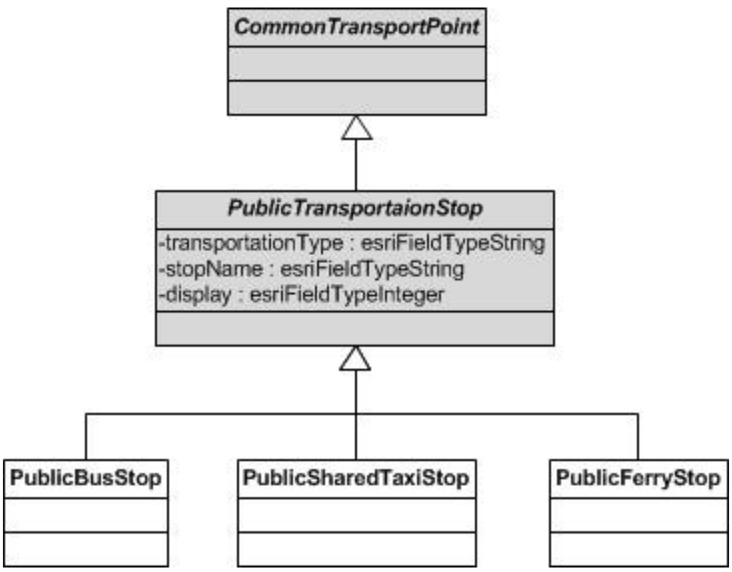



**Figure 6.8 :** Feature dataset “Transport_Network – Public Transportation Stop” UML diagram.

Also, other attributes refer to range of values. These have been modeled as <<coded and range>> stereotype domains. The designed UML model of the Transportation Network and the Transportation Domain Values can be seen Appendix A.

### 6.1.2 Export transport network UML model to XMI file

The modeled Transport Network Schema in UML is exported to a XMI file. Afterwards, this exported file will be used in order to generate Transportation Geodatabase based on the UML models.

Inside Microsoft Visio is it possible to insert a new “additional component” (Add-Ons) or register a macro inside the project, allowing to export in XMI format the diagram created (figure 6.9).

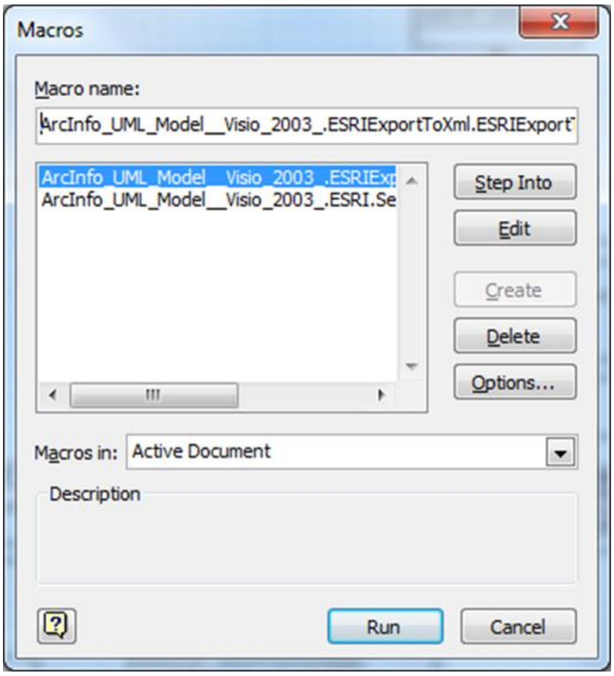



**Figure 6.9 :** ExportToXML macro in Microsoft Visio.

The Semantics Checker (macro registered in the ArcInfo UML Model) can be used to verify that a model stored in XMI has been correctly defined. It will produce a report with the list of errors encountered in the model. It is highly recommended to use the semantics checker before running the CASE tools wizards. The semantics checker can be run from within Visio when the template diagram is loaded (figure 6.10).

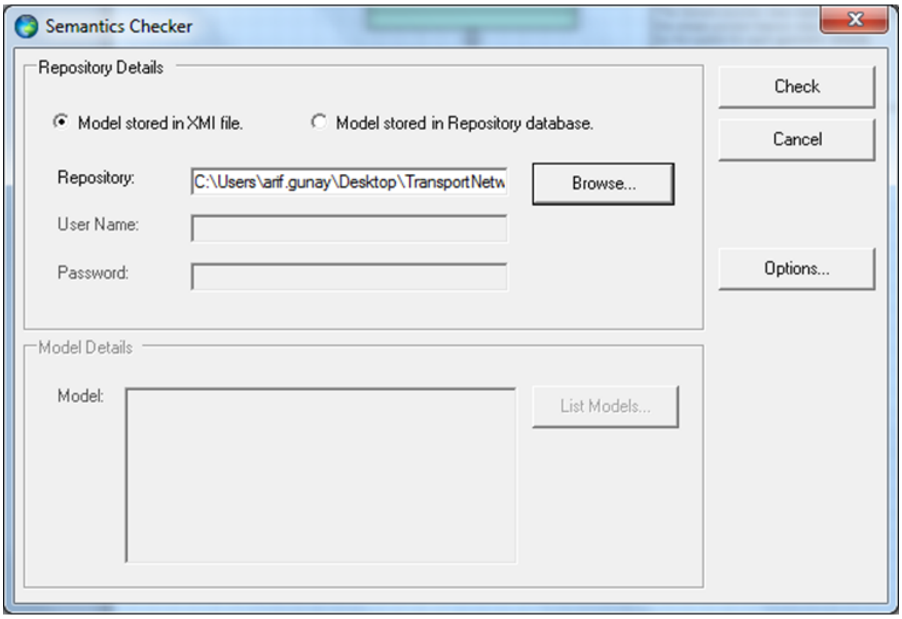



**Figure 6.10 :** Semantics checker.

### 6.1.3 Creation the transport network geodatabase from XMI file

Transport Network Geodatabase is generated with the use of exported XMI file. In ArcCatalog (ArcEditor and ArcInfo versions), it is possible to add in the toolbar the command "Schema Wizard", which allows to import the geodatabase schema modeled in Microsoft Visio, shown in figure 6.11.

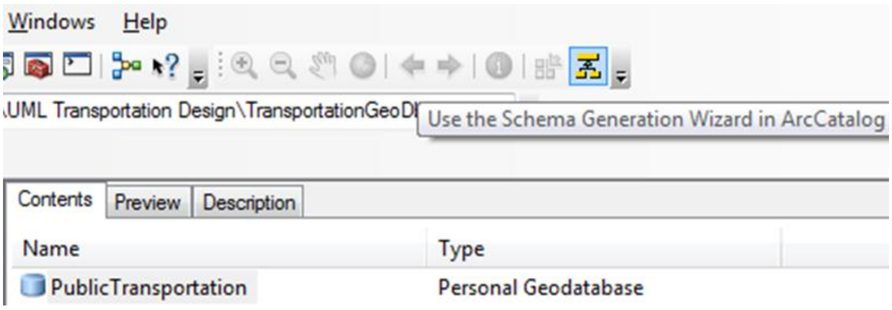



**Figure 6.11 :** Geodatabase schema wizard tool.

In this 'Schema Wizard' the geodatabase and the XMI-file have to be specified. Selecting an existing geodatabase or creating a new one (File or Personal) it is possible to start the wizard to import the XMI file (figure 6.12).

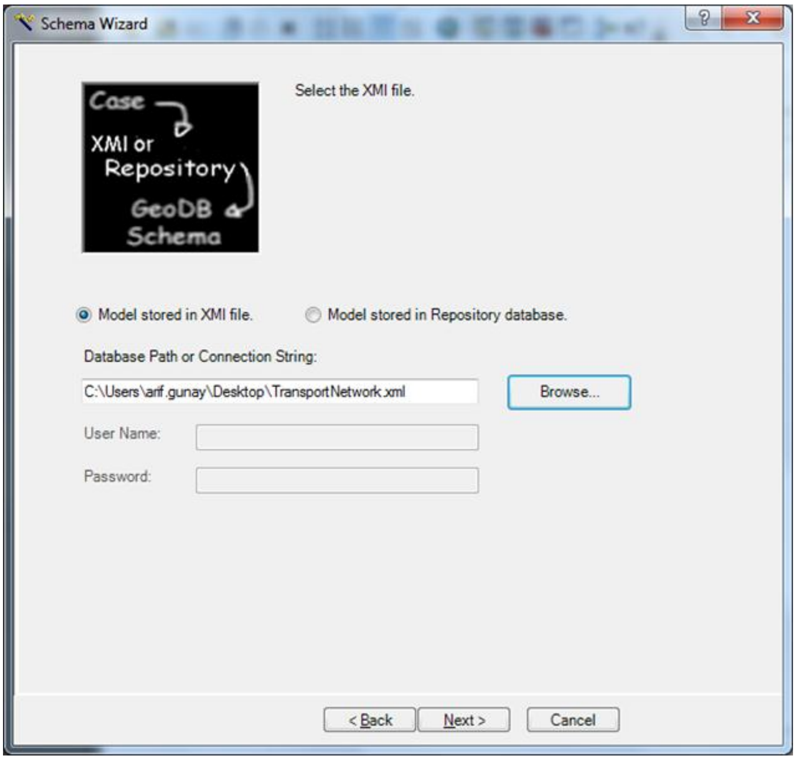



**Figure 6.12 :** Geodatabase schema wizard using XMI file.

After having selected the XMI file exported by Visio the wizard starts importing the objects. The Schema Wizard form shows the objects in a tree view. All of the datasets in the object model are selected and have check marks beside them by default. It is possible to uncheck the check boxes to unselect any dataset that you do not want to add to the geodatabase. Clicking the plus and minus signs located next to dataset allow to expand and contract the tree view. At this stage, it is required to select a dataset and set the Spatial Reference for each dataset in the Catalog tree. All of the feature classes in a feature dataset must share the same spatial reference. In the Schema Wizard form, it is also possible to highlight an object and modify the object's class properties. Once all the eventual modifications have been completed, a list of all the feature and the object classes created is visualized and it is now possible to import the schema in the geodatabase (figure 6.13).

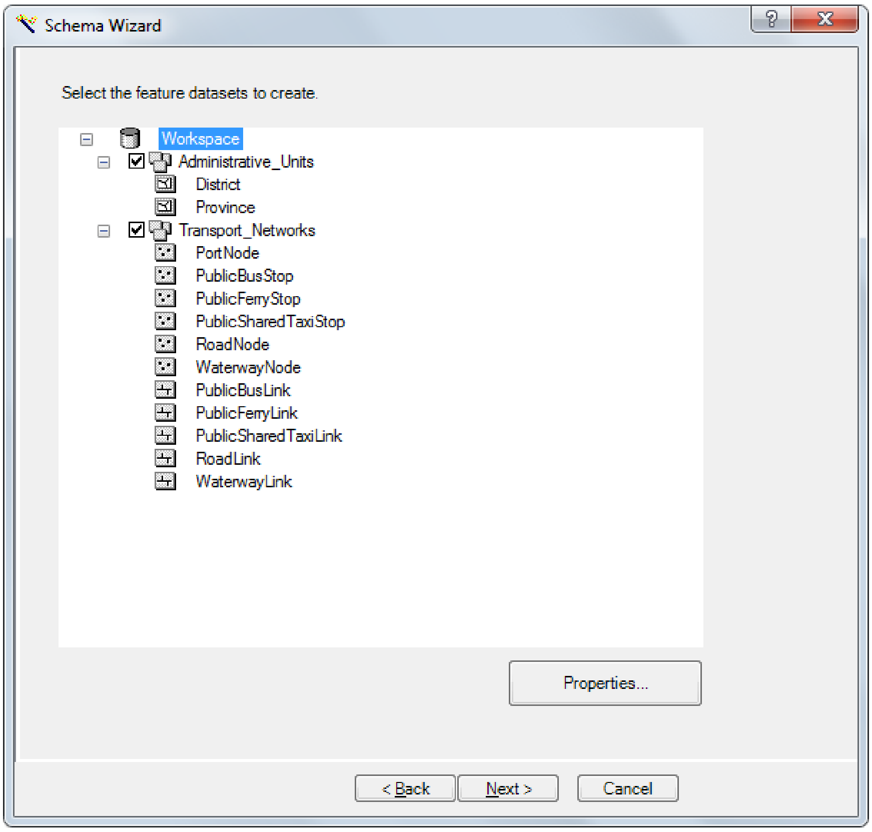



**Figure 6.13 :** Selecting feature datasets to create.

The geodatabase now contains the objects (feature datasets, feature classes, tables, and relationships) that were defined in the object model and generated by the CASE Schema Creation Wizard. Created geodatabase; feature datasets, feature classes and domains are shown Appendix B.

## 6.2 Development of Public Transportation Ontology Domain

Spatial ontology is an explicit specification of spatial conceptualization which describes spatial concepts, relationships, and individuals for public transportation. Therefore, a spatial term can be inferred to different spatial concepts based on spatial ontology. These spatial concepts are further divided into different spatial concepts. For example, transportation route are divided into bus, sharedtaxi, ferry route, and transportation stops are divided into bus, sharedtaxi and ferry stops, which is used to discover GIS route features in geodatabase.

In this study, OWL ontology consists of Concepts, Properties and Individuals (Classes) are developed for the Public Transportation Network. In order to make use of these knowledge effectively, we have designed the knowledge base system based on the OWL ontologies and rules with the help of Protégé Ontolgy Editor (Protégé).

The carried out steps for the creation of Public Transportation Ontology Domain are explained in following sections.

### 6.2.1 Class hierarchy and taxonomies

We have defined the Public Transportation Network class hierarchy considering the pre-built Transport Network Schema. The ontology classes mainly involve Transportation Route, Transportation Link and Transportation Stop in the ontology domain. These concept are further divided into sub concepts with "is-a" relation. Modeled class hierarchy described as follows (ontology view shown in figure 6.14).

- Public Transportation Route, Public Transportation Link and Public Transportation Stop are subclass of Public Transportation Network
- Bus Route, Ferry Route and SharedTaxi Route are subclass of Public Transportation Route
- Bus Link, Ferry Link and SharedTaxi Link are subclass of under the Public Transportation Link
- Likewise, Bus Stop, Ferry Stop and Shared Taxi Stop are created as subclass of Public Transportation Stop

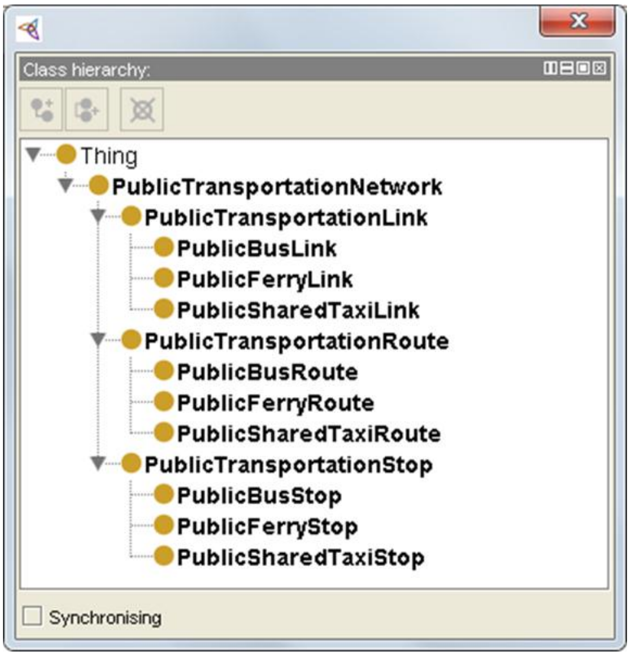



**Figure 6.14 :** Class hierarchy of public transportation network.

### 6.2.2 Object property hierarchy

Once we have model the concepts of the ontology, relationships are defined between these concepts and restrict their interpretation using properties. Object properties are relationships between two individuals. Four main object property relations are created in this ontology domain: hasLink, hasStop, hasStartStop, and hasEndStop. Each object property have also a corresponding inverse property, which enables to infer implicit taxonomic relationships and equalities automatically. If some property links individual a to individual b then its inverse property will link individual b to individual a. Therefore, in our ontology domain, isStopOf object property is created as inverse property of hasStop and isLinkOf object property is created as inverse property of hasLink. In this investigation, we have defined object properties hierarchy as follow figure 6.15:

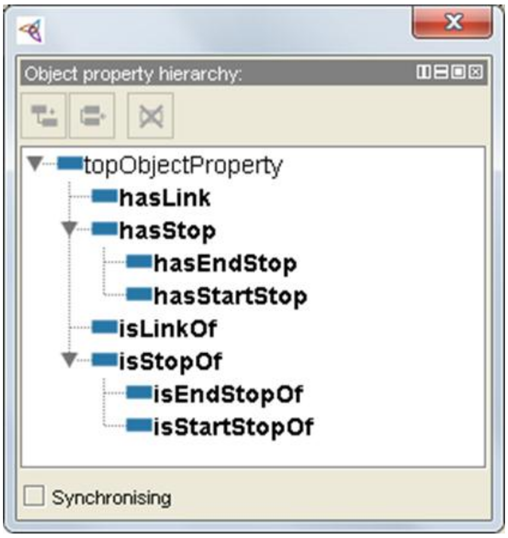



**Figure 6.15 :** Object property hierarchy.

In order to define relationships between concepts in ontology restrictions needs to be declared via the object property such as existential restriction and cardinality restriction.

### 6.2.3 Existential and universal restrictions (quantifier restrictions)

**Existential restrictions:** describe classes of individuals that participate in at least one relationship along a specified property to individuals that are members of a specified class. For example, the class of individuals that have at least one (some) hasStop relationship to members of Public Transportation Stop. In Protege the keyword, `some' is used to denote existential restrictions. According to this, following object property definitions through "some" existential restriction is created in the public transportation ontology domain.

- Public Transportation Route hasLink (some) Public Transportation Link (shown in figure 6.16)
- Public Transportation Link isLinkOf (some) Public Transportation Route (shown in figure 6.17)
- Public Transportation Link hasStop (some) Public Transportation Stop (shown in figure 6.17)
- Same relations are defined for the Transportation Route Sub Classes (Bus, SharedTaxi and Ferry). Figure 6.18 shows the hasLink (some) existential restriction between PublicBusRoute and PublicBusLink.

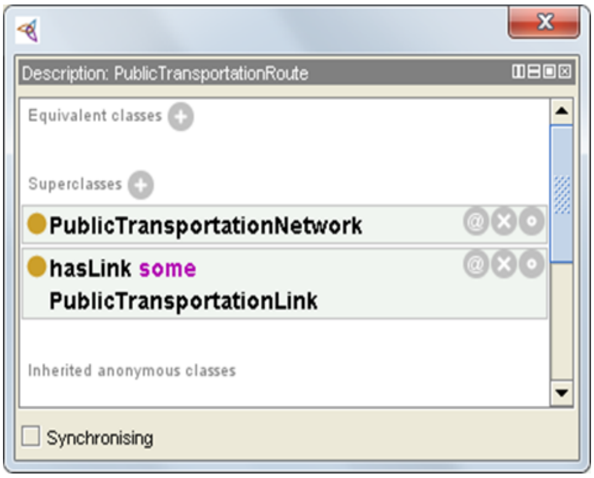



**Figure 6.16 :** Object property hasLink for Transportation Route.

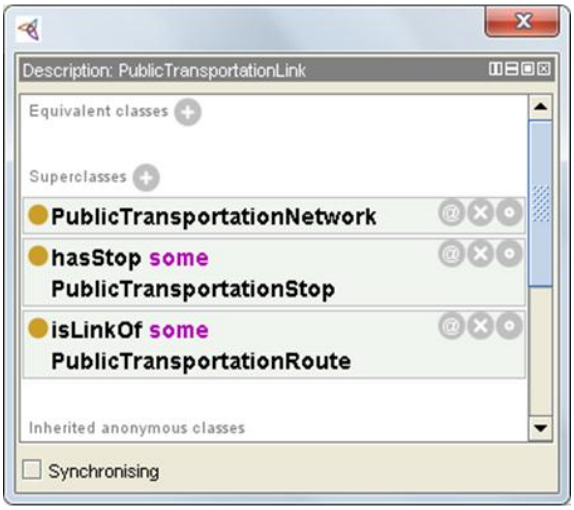



**Figure 6.17 :** Object property isLinkOf and hasStop for Transportation Link.


**Figure 6.18 :** Object property hasLink for Bus Route and Link.

**Cardinality restrictions:** we can describe the class of individuals that have at least, at most or exactly a specified number of relationships with other individuals or datatype values. The restrictions describe these classes are known as Cardinality Restrictions. In our ontology, we have modeled the Bus Link, which must have exactly one Start Stop and one EndStop. In this regard, we use the "exactly=1" cardinality restriction through the object property to define this relationship. Below figure 6.19 shows the description for Public BusLink hasStartStop and hasEndStop properties related with Public BusStop using (exactly =1) cardinality restriction.

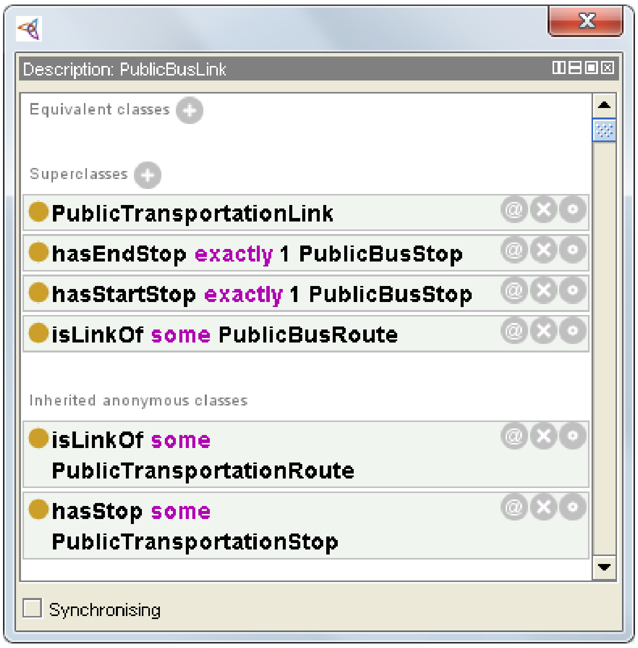



**Figure 6.19 :** Cardinality restriction hasStartStop and hasEndStop properties related with Public BusStop using (exactly =1).

### 6.2.4 Definition of individuals – member lists

After defining classes, object properties and relations using with restrictions, all individuals related to our public transportation dataset are created in ontology domain.

During the creation Route and Link individuals, we have considered the round-travel direction. Round-travel directions (From-To) can be different in the transportation system. For example, from-to direction of the bus routes are not exactly the same line with to-from direction of the bus route and they have even different stop names along the round- travel. Therefore, transportation routes are modeled for representing both directions separately (From-To and To-From) in the ontology domain.

We have defined the naming notations for Route and Link individuals as first letter of classname_route number_from/to direction. For example, Public Bus Route individuals can be seen as B_110_FT naming convention. Figure 6.20 shows the created Bus Route and Link individuals considering the round-travel directions, and Bus Stop individuals are illustrated in the figure as well.

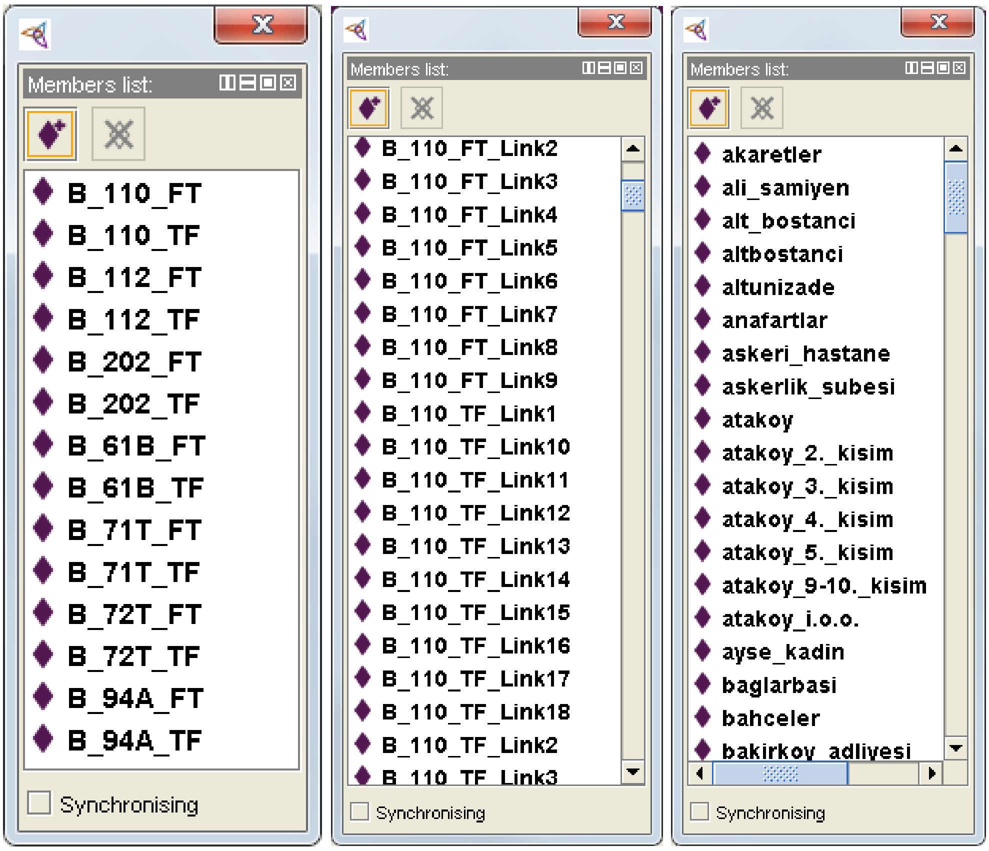



**Figure 6.20 :** Bus route, bus link and bus stop individuals.

Another thing we need to deal is that definition of common stop name for the transportation types during the individual creation. The several transportation means may pass by the same place but with different stop names or same stop name may appear at several different places. It is possible to declare these stops as **Same Individual** in ontology to overcome this sort of ambiguities. For example, Taksim_Meydan is the Stop Name of the Public Shared Taxi Stop. On the other hand, Taksim is the Stop Name of Public Bus Stop. Actually, both of them represent the same or near location but different names. Therefore, these kind of Stops are defined as Same Individual in the ontology domain. Below figure 6.21 shows the same individual definitions between Taksim and Taksim_Meydan, also Kadikoy and Kadikoy_Meydan, Kadikoy_Iskele as an example.

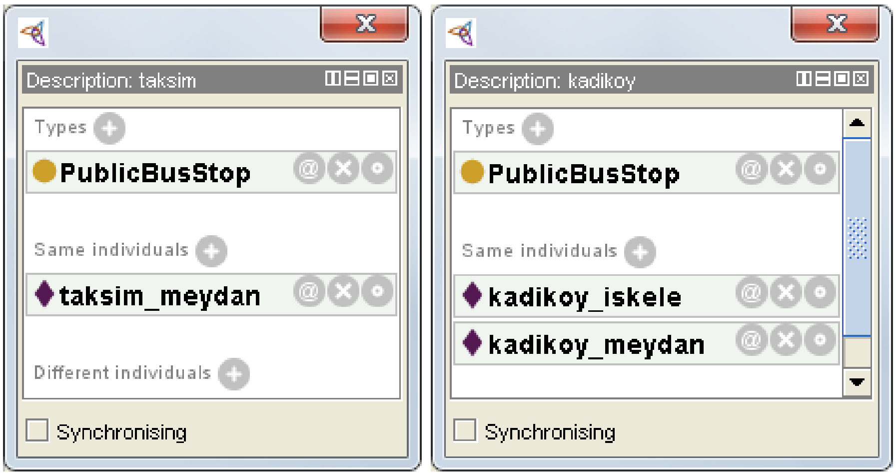



**Figure 6.21 :** Same individual definitions for taksim and kadikoy stop points.

The other Routes, Link and Stop individuals of SharedTaxi and Ferry Concepts are also created in the ontology domain as the same way described above for Bus Concept. Developed Public Transportation Network Ontology Domain OWL file can be seen in Appendix C.

### 6.2.5 Datatype property hierarchy

After the creation of instances, which we decided as individuals of the ontology, these individuals are related using the datatype properties. Datatype properties describe the relationships between an individual and data values in OWL ontology. Datatype properties link an individual to an XML Schema Datatype value or an rdf literal. Below figure 6.22 shows the created data properties in this transportation ontology.

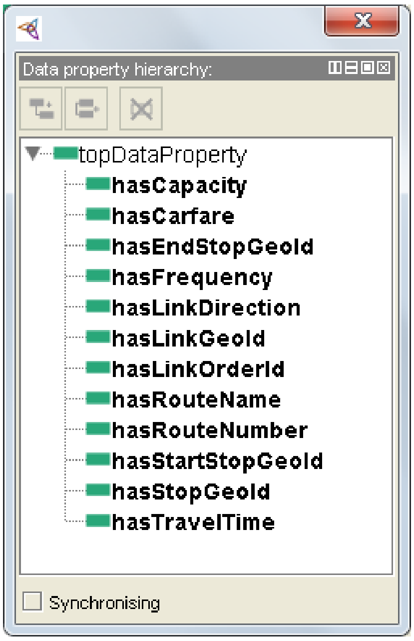



**Figure 6.22 :** Datatype property hierarchy.

“hasLinkGeoId”, “hasOrderId” properties are used to link ontology results with the transportation link features in the geodatabase. “hasStartStopGeoId” and “hasEndStopGeoId” properties store the stop feature Ids in the geodatabase so that user requested stop points can linked in the feature dataset via ontology result.

A datatype property can also be used in a restriction to relate individuals to members of a given datatype through the domain and range definitions. Built in datatypes are specified in the XML schema vocabulary and include integers, floats, strings, booleans etc. Domain and Range definitions for “hasLinkOrderID datatype properties of Transportation Link are depicted in figure 6.23. In this case, “hasLinkOrderId” property is restricted to use only integer values, other value are not allowed.


**Figure 6.23 :** Domain and range definition for hasLinkOrderId datatype property.

Datatype property values are assigned using the `datatype properties assertions view' for the individuals. Object property and datatype property definitions for B_110_FT_Link 1 individual is exampled below figure 6.24.

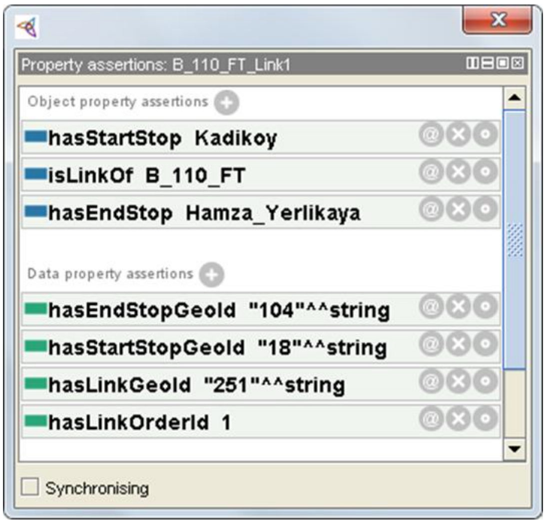



**Figure 6.24 :** Object property and datatype propertiy definitions of transportation bus links.

Below figure 6.25 shows object and datatype property definitions of sharedtaxi links and ferry links for ST_11_FT Link 1 and F_11_FT_Link 1 individuals.

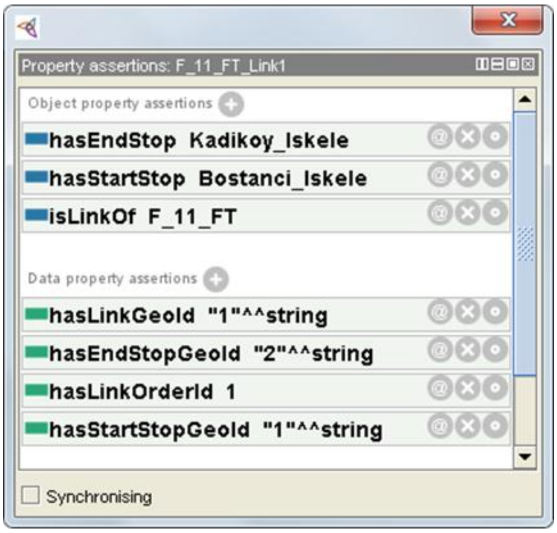




**Figure 6.25 :** Object property and datatype propertiy definitions of transportation shared taxi links and ferry links.

In addition to this, “hasCarfare”, “hasFrequency” and “hasTravelTime” datatype properties are used to describe some facet of the concept or entity such as the price for travel ticket, time frequency for departure, estimated time of the travel and so on.

Below figure 6.26 shows "hasCarfare", "hasFrequency" and "hasTravelTime" and other datatype property definitions of Bus Route B_110_FT individual.

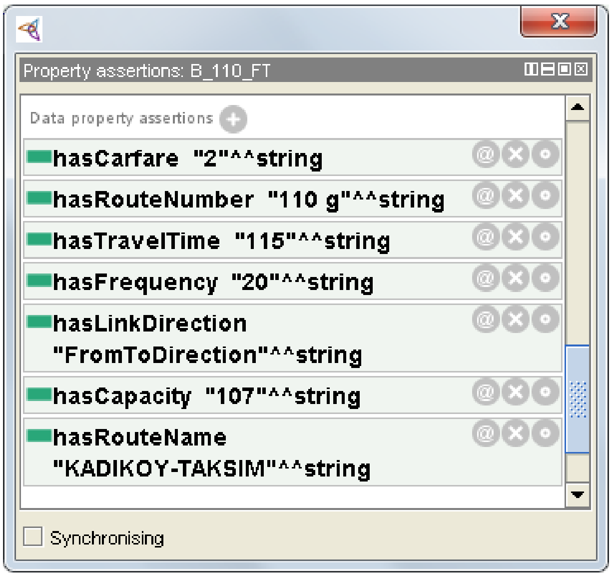



**Figure 6.26 :** hasCarfare, hasFrequency and hasTravelTime and other datatype property definitions of Bus Route B_110_FT.

### 6.2.6 Asserted and inferred models of achieved public transportation ontology

After overall ontology definitions explained above, inferences join all the expressions and derive new knowledge. But when learning process, it is not desired to see all the inferences at once, it is better to see only the ones the user anticipates (the ones user expects to find), and only the ones which are related to the concepts just defined and being analyzed at the moment of running the reasoner. OntoViz and OntoGraf add-on graphic tool of Protege are very useful in such process, which helped a lot in the discovery of new knowledge. OntoViz help us the show the set of asserted and inferred hierarchies (fig. 6.27) and OntoGraf gives support for interactively navigating the relationships of the OWL ontologies (fig. 6.28). Public Transportation Network Ontology Domain OWL file can be seen in Appendix C.

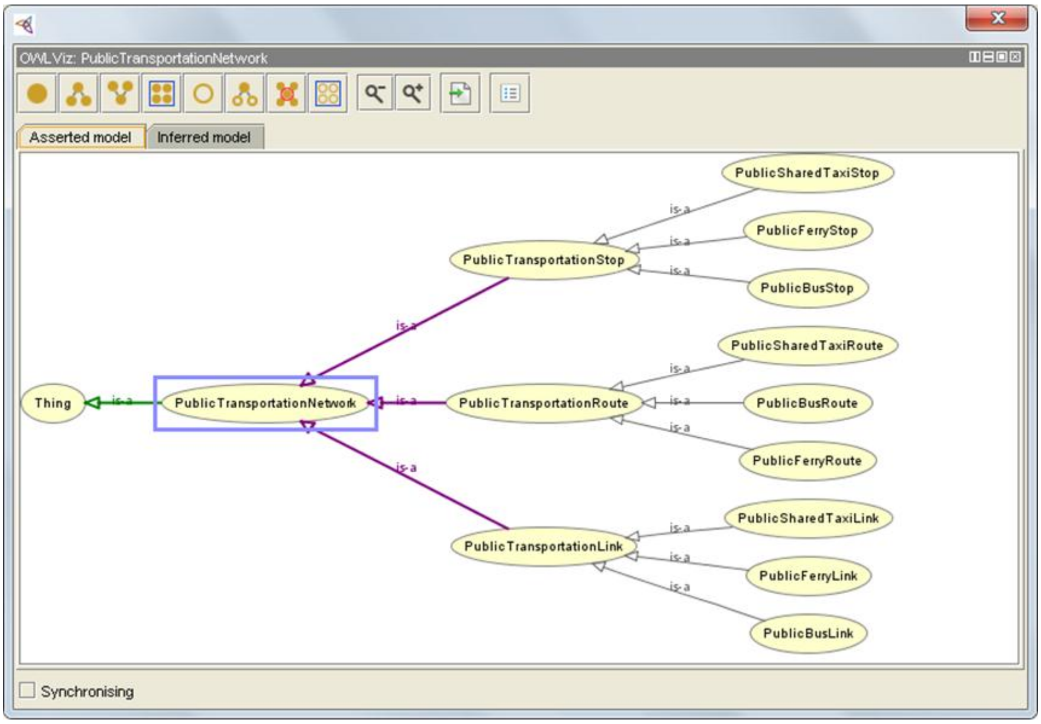



**Figure 6.27 :** Asserted and inferred models of the public transportation ontology domain view with the OntoViz tool.

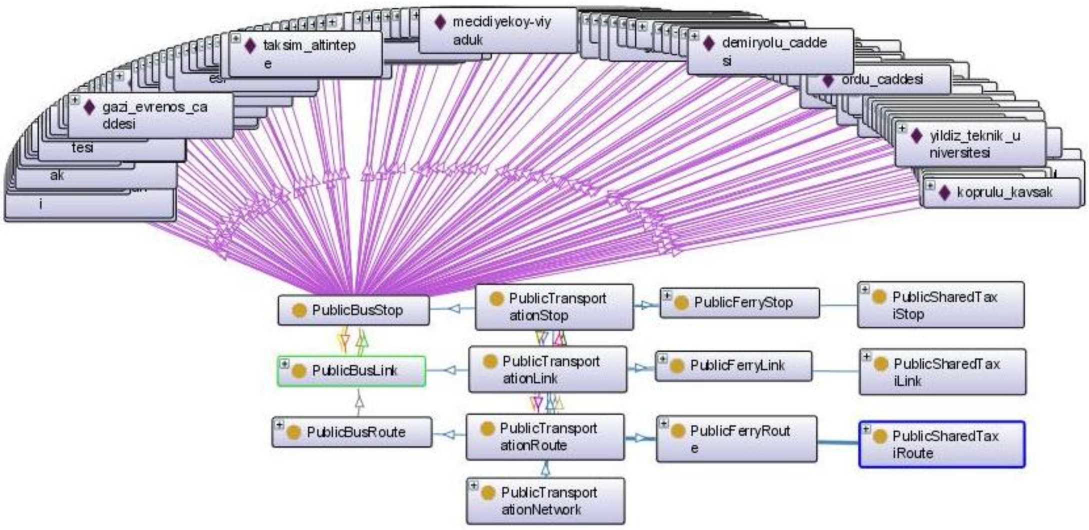



**Figure 6.28 :** Interactively navigating the relationships of the OWL ontologies with the OntoGraf tool.

## 6.3 Development of Semantic Spatial Query Algorithms Using the Reasoning Engine

In order to perform semantic queries in transportation ontology domain, nRQL (new Racerpro Query Language) of RacerPro Reasoning Engine is used (Racer Systems). Development of semantic query concept shown in figure 6.29.

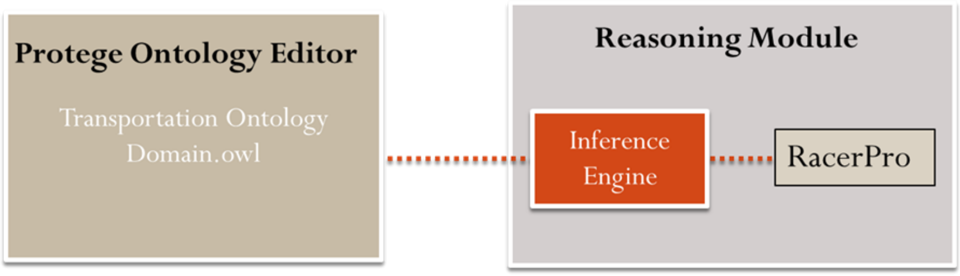



**Figure 6.29 :** Development of semantic queries using ontology domain concept.

In order to retrieve Transportation Route alternatives based on the indicated stop locations, relevant GeoId of Links should be discovered in Geodatabase based on the concepts that are inferred in spatial and transportation domain knowledge ontologies. The object properties (hasLinkGeoId, hasLinkOrderId) are used to retrieve Transportation Route GeoIds and link to the Transportation Geodatabase in order to publish discovered alternatives routes on the Geoportal. For example in figure 6.30, the public transportation concepts (BusLink, SharedTaxiLink, FerryLink and so on) and relations (hasLink, hasStartStop, hasEndStopOf and inverse object properties) are shown which is used to discover alternative routes in the domain ontology. Figure 6.31 shows the arc type definitions for the represented figure 6.30.

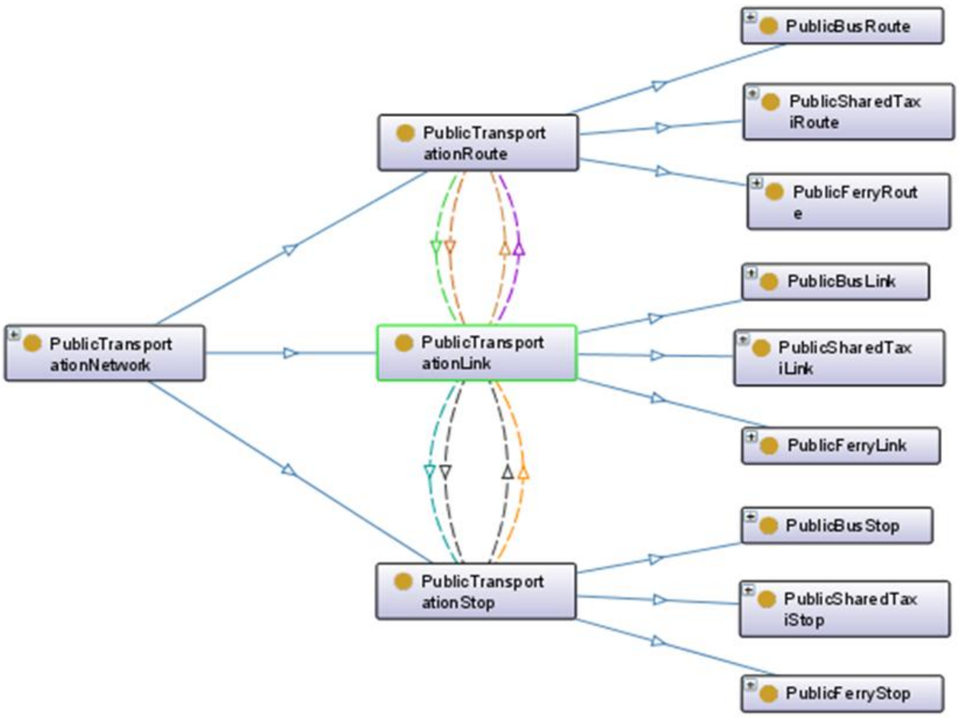



**Figure 6.30 :** Ontology domain concepts and object property relationships in OntoGraf view.

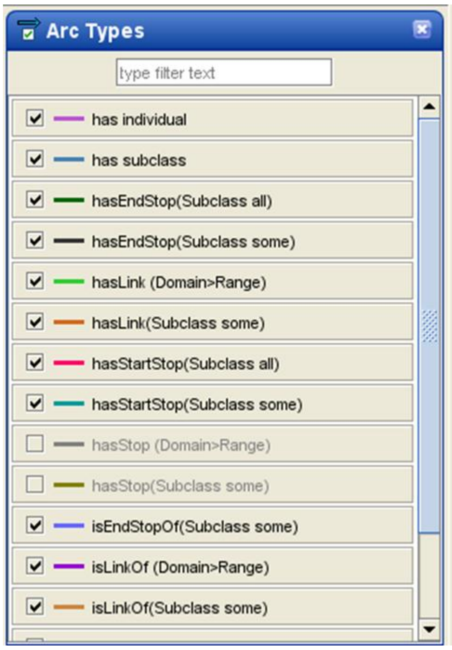



**Figure 6.31 :** Arc type definitions for the represented ontology in OntoGraf view.

**Defining semantic queries:** Semantic spatial query algorithms are developed in order to retrieve transportation route alternatives based on the user requested start and end stop points. Route alternatives retrieved from the transportation ontology domain via the RacerPro Reasoning Module. The algorithm first attempts to search for no-transfer travel. If no zero transfer travel is found, it seeks for potential one-transfer travel in the transportation domain.

The developed transportation algorithm considers the synonym and/or homonym heterogeneity for querying between two stop, since the stop names are differentiating in the ontology domain with same individual definitions. The developed transportation algorithm is also considers the condition that round-travel directions are different in the transportation system. It is also possible to retrieve data values with the use of developed algorithms such as carfare, time frequency of the travel, estimated travel time and so on.

The below nRQL algorithm returns bus route alternatives based on the given start and end stop names. The query steps are explained below which needs to be performed in order to get bus route alternatives:

1) Add role assertions to work with constraint query atoms as well for datatype properties (6.1).

Add-role-assertions-for-datatype-properties; **(6.1)**

2) Execute query to find bus route, start link and end link considering From and To link direction based on the given stop names (6.2).

(retrieve (?linkRoute ?linkStart ?linkEnd) (and
(#!Ontology1325784716005:busStop1 ?linkStart
#!Ontology1325784716005:isStartStopOf)
(#!Ontology1325784716005:busStop2 ?linkEnd
#!Ontology1325784716005:isEndStopOf) (?linkStart
#!Ontology1325784716005:PublicBusLink) (?linkEnd
#!Ontology1325784716005:PublicBusLink) (?linkStart ?linkRoute
#!Ontology1325784716005:isLinkOf) (?linkEnd ?linkRoute
#!Ontology1325784716005:isLinkOf) (?linkStart ?linkEnd (constraint
(#!Ontology1325784716005:hasLinkOrderId)
(#!Ontology1325784716005:hasLinkOrderId)<= )))); **(6.2)**

3) Execute query to retrieve linkGeoId values of found route alternatives based on the result of previous query result (6.3).

(retrieve (datatype-fillers (#!Ontology1325784716005:hasLinkGeoId ?y))
(and (?y #!Ontology1325784716005:linkRoute
#!Ontology1325784716005:isLinkOf) (?y (and (>=
#!Ontology1325784716005:hasLinkOrderId
#!Ontology1325784716005:linkStart)
(<=#!Ontology1325784716005:hasLinkOrderId
#!Ontology1325784716005:linkEnd))); **(6.3)**

4) Execute query to retrieve Start Stop GeoId value of found route alternatives based on the result of previous query result (6.4).

(retrieve (datatype-fillers (#!Ontology1325784716005:hasStartStopGeoId
?y)) (and (?y #!Ontology1325784716005:PublicBusLink) (= ?y
#!Ontology1325784716005:linkStart))); **(6.4)**

5) Execute query to retrieve End Stop GeoId value of found route alternatives based on the result of previous query result (6.5).

(retrieve (datatype-fillers (#!Ontology1325784716005:hasStartStopGeoId
?y)) (and (?y #!Ontology1325784716005:PublicBusLink) (= ?y
#!Ontology1325784716005:linkEnd))); **(6.5)**

6) Execute query to retrieve linkRoute’s datatype properties; route name, route number, car fare, travel frequency, estimated travel time defined in ontology domain (6.6).

(retrieve ((datatype-fillers (#!Ontology1325784716005:hasRouteName ?y)) (datatype-fillers (#!Ontology1325784716005:hasRouteNumber ?y)) (datatype-fillers (#!Ontology1325784716005:hasCarfare ?y)) (datatype-fillers (#!Ontology1325784716005:hasFrequency ?y)) (datatype-fillers (#!Ontology1325784716005:hasTravelTime ?y))) (and (?y #!Ontology1325784716005:PublicBusRoute) (= ?y #!Ontology1325784716005:linkRoute))); **(6.6)**

7) Execute transfer query to retrieve route alternatives if there is no direct travel route. Algorithm implemented for the one transfer option (6.7).

(retrieve (?linkRoute1 ?linkStart1 ?linkEnd1 ?linkRoute2 ?linkStart2 ?linkEnd2) (and (#!Ontology1325784716005:busStop1 ?linkStart1 #!Ontology1325784716005:isStartStopOf) (?x ?linkEnd1 #!Ontology1325784716005:isEndStopOf) (?linkStart1 #!Ontology1325784716005:PublicBusLink) (?linkEnd1 #!Ontology1325784716005:PublicBusLink) (?linkStart1 ?linkRoute1 #!Ontology1325784716005:isLinkOf) (?linkEnd1 ?linkRoute1 #!Ontology1325784716005:isLinkOf) (?linkStart1 ?linkEnd1 (constraint (#!Ontology1325784716005:hasLinkOrderId) (#!Ontology1325784716005:hasLinkOrderId) <= ))

(?x ?linkStart2 #!Ontology1325784716005:isStartStopOf) (#!Ontology1325784716005:busStop2 ?linkEnd2 #!Ontology1325784716005:isEndStopOf) (?linkStart2 #!Ontology1325784716005:PublicBusLink) (?linkEnd2 #!Ontology1325784716005:PublicBusLink) (?linkStart2 ?linkRoute2 #!Ontology1325784716005:isLinkOf) (?linkEnd2 ?linkRoute2 #!Ontology1325784716005:isLinkOf) (?linkStart2 ?linkEnd2 (constraint (#!Ontology1325784716005:hasLinkOrderId) (#!Ontology1325784716005:hasLinkOrderId)<= )))); **(6.7)**

Given nRQL codes are developed to determine Bus Route based on the given stop names. Likewise, these nRQL codes are implemented to retrieve SharedTaxi, Ferry Routes alternatives and other required information like carfare, estimated time and so on. The developed Transportation nRQL query algorithms can be found in the Appendix D.

# 7. IMPLEMENTATION: SEMANTIC PUBLIC TRANSPORTATION GEOPORTAL

In this thesis research, tree-tier system architecture (shown figure 7.1) is implemented to build geoportal and perform semantic queries. Data Layer shows the used dataset for this study. Previously created Transportation Geodatabase Schema in section 6.1 is transferred in to the used database management system (PostGIS) during this implementation. Application layer consist of Ontology Server and Reasoning Engine in addition to GIS Server. All the relationships of public transportation data is defined with in the Ontology Domain (in section 6.2) and semantic query algorithms are coded using the Reasoning Engine (in section 6.3). Thus, concepts of the dataset can be retrieved and intelligent spatial queries can be performed compare to traditional GIS keyword-base spatial search method. Lastly, presentation layer is build using the open source Open Layer software to provide access to the users via the web interface. Working principle of the system is shown below figure 7.1.

Overview of the system components:

- Database Management System
    - PostgreSQL for RDBMS and PostGIS extension for managing the geo-db
    - PostGIS Shp file and dbf loader to import INSPIRE Transport Schema
- Semantic Layer
    - Protégé Ontology Editor to create transportation ontology domain
    - RacerPro Reasoning Module to run ontology queries
- GIS Web Server
    - Geoserver provides the web platform to publish dataset. Support: Implementation of WFS, WMS OGC Standards

- OpenLayers JavaScript API to put a dynamic map into web page
- Web Server to establish client browser connection
    - Apache Tomcat Web Server
- Java programming language to integrate the software and build the architecture

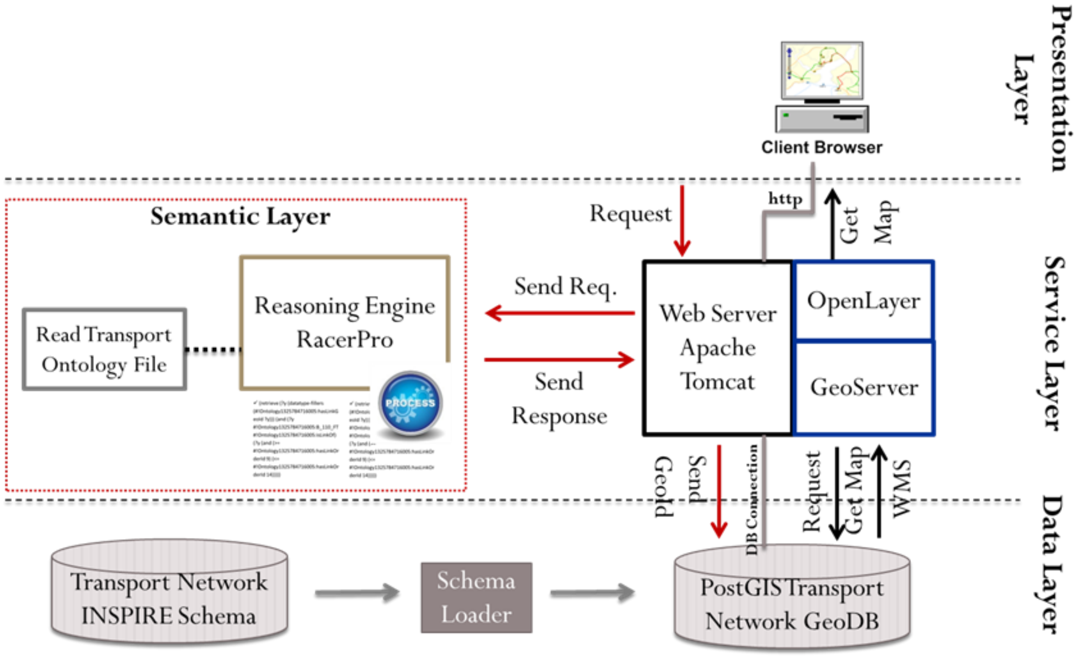



**Figure 7.1 :** Working principle of the system.

## 7.1 Data Layer

During this thesis research following transportation dataset with the given numbers of features are used covering the sample public transportation area of Istanbul.

- Provinces
- Road
- Public Transportation Link
    - Bus Link (3658 Links)
    - SharedTaxi Link (12 Links)
    - Ferry Link (3 Links)
- Public Transportation Stop

- Bus Stop (244 Stops)
- SharedTaxi Stop (15 Stops)
- Ferry Stop (4 Stops)

In this thesis research, we have planned to use open source PostgreSQL/PostGIS database system and implement geoportal application on the top of that. Therefore, a database is created in the PostGIS System and previously created Transportation Geodatabase Schema compliant with INSPIRE is imported into the PostGIS. In order to perform this import operation, we used the Shape File to PostGIS Importer tool as shown below figure 7.2.

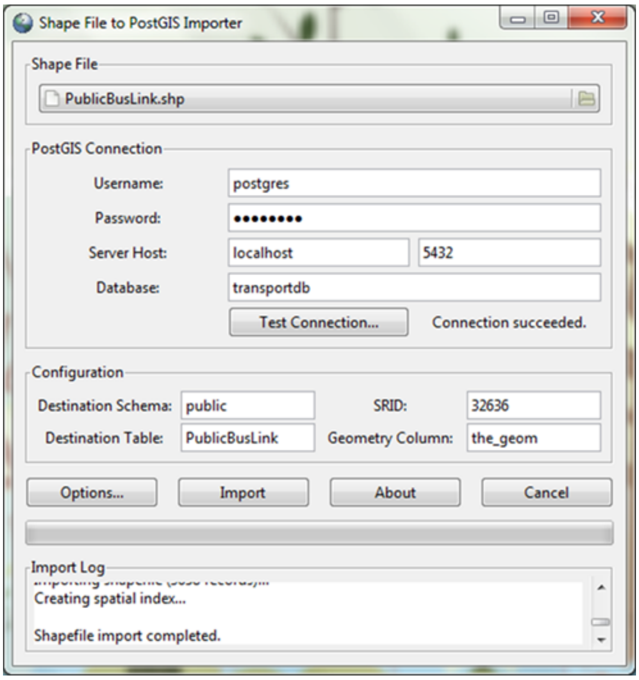



**Figure 7.2 :** Example of public bus link data into PostGIS transportdb by using shape file to PostGIS importer.

Thus, the migration of transportation geodatabase into the PostGIS open source database system is carried out.

## 7.2 Service Layer

### 7.2.1 Semantic service layer: Retrieve route data from ontology domain

To retrieve transportation route alternatives based on the indicated stop locations, relevant GeoId of Links should be discovered in Geodatabase based on the concepts that are inferred in spatial and transportation domain knowledge ontologies.

The object properties (hasLinkGeoId, hasLinkOrderId, hasStopID) are used to retrieve GIS Route GeoId and link to the Transportation Geodatabase in order to publish discovered alternatives routes on the Geoportal.

In order to perform semantic queries in transportation ontology domain, nRQL (new Racerpro Query Language) of RacerPro Reasoning Engine is used. Therefore, a connection to the RacerPro Engine must be established before any queries can be executed. The following java code example shows how a connection to an ontology source can be opened via the Java RacerPro API (7.1).

```
String ip = "localhost";
int port = 8088;
RacerClient racer = new RacerClient(ip, port);
racer.openConnection();
racer.loggingOn();
racer.sendRaw("(logging-on)");
String publicTransportation="C:/RacerPro-Preview/publicTransportation.owl";
racer.owlReadFile(publicTransportation);
```

**(7.1)**

The below java codes shows how to run nRQL semantic queries to retrieve bus route alternatives based on the given start and end stop names. In the following steps are shows how to run nRQL queries with the java codes.

1) Open RacerPro connection and add role assertions to work with constraint query atoms as well for datatype properties (7.2).

   ```
   racer.openConnection();
   racer.addRoleAssertionsForDatatypeProperties();
   ```

   **(7.2)**

2) Execute query to find route, start link and end link considering From and To link direction based on the given stop names (7.3).

   ```
   racer.racerAnswerQuery("(?linkRoute ?linkStart ?linkEnd)", "(and
   (#!Ontology1325784716005:" + busStop1 + " ?linkStart
   #!Ontology1325784716005:isStartStopOf) (#!Ontology1325784716005:" +
   busStop2 + " ?linkEnd #!Ontology1325784716005:isEndStopOf) (?linkStart
   ```

```
#!Ontology1325784716005:PublicBusLink) (?linkEnd
#!Ontology1325784716005:PublicBusLink) (?linkStart ?linkRoute
#!Ontology1325784716005:isLinkOf) (?linkEnd ?linkRoute
#!Ontology1325784716005:isLinkOf) (?linkStart ?linkEnd (constraint
(#!Ontology1325784716005:hasLinkOrderId)
(#!Ontology1325784716005:hasLinkOrderId) <= )))");
```
**(7.3)**

In the developed application, previously developed nRQL semantic queries can be run by the use of java programming language in order to determine Bus Route, SharedTaxi and Ferry Route alternatives based on the given stop names. The java code itself can be found in the Appendix E.

Likewise, we are able to retrieve all necessary information (such as LinkGeoIds, StopGeoIds, Route Name, Route Number, Car Fare, etc.) from the created Transportation Domain Ontology without running SQL queries inside of the java application.

### 7.2.2 GIS service layer: Linking semantic query result with the geodatabase

Discovered route alternatives based on the semantic query result are sent to the PostGIS Geodatabase to link the Route spatial data. Thus, obtained route alternatives and other information using ontology domain can be processed into the Geodatabase.

To link retrieved GeoIds of route alternatives with the equivalent GIS route link and stop data in Geodatabase, we have created a look up field in route link and stop geo tables, named as “display”. Accordingly, we have set the “display” fields as “1” for the equivalent GIS data in Geodatabase based on the obtained GeoIds of Route Link and Stops from semantic query result.

In the following code samples illustrate the linking GIS Bus Route with the use of achieved GeoIds from Transportation Domain Ontology.

**Creating db connection and sending semantic query result:** Before any queries can be executed a connection to the repository must be established. Therefore the DriverManager returns a reference to the connection object, which is then used to create a statement. After asserting the statement into the working memory, a simple query is created and executed. An example shows the following code fragment.

1) Create and Open PostgreSQL database connection (7.4).

```
Class.forName("org.postgresql.Driver");

db = DriverManager.getConnection
("jdbc:postgresql://localhost:5432/transportdb", "postgres", "Admin123");

sql = db.createStatement();                                        (7.4)
```

2) Execute the java code to send retrieved query result: Link GeoIds to related GIS RouteLink data Geodatabase. Below code (7.5), busQueryMembers store the semantic query results which is explained section 7.3.1.

```
for (int i=0; i < routeCounter; i++){

    for (int j=1; j < query2Members[i].length; j+=2){

        sqlBus = "UPDATE \"PublicBusLink\" SET display = 1 WHERE gid
= "+    busQueryMembers[i][j] +" ";

        sql.executeUpdate(sqlBus);  }  }                           (7.5)
```

3) Execute the code to send retrieved query result: Start Stop GeoIds to related GIS Stop data Geodatabase. Below code (7.6), busStartStopGeoId, busTransferStopGeoId, busEndStopGeoId, store the semantic query results which is explained section 7.3.1.

```
sqlBus = "UPDATE \"PublicBusStop\" SET display = 1 WHERE gid = "+
busStartStopGeoId +" ";

sql.executeUpdate(sqlBus);                                         (7.6)
```

4) Execute the code to send retrieved query result: Transfer Stop GeoIds to related GIS Stop data Geodatabase (7.7).

```
sqlBus = "UPDATE \"PublicBusStop\" SET display = 1 WHERE gid = "+
busTransferStopGeoId +" ";

sql.executeUpdate(sqlBus);                                         (7.7)
```

5) Execute the code to send retrieved query result: End Stop GeoIds to related GIS Stop data Geodatabase (7.8).

```
sqlBus = "UPDATE \"PublicBusStop\" SET display = 1 WHERE gid = "+
busEndStopGeoId +" ";

sql.executeUpdate(sqlBus);                                         (7.8)
```

In the same way, achieved semantic query results of SharedTaxi and Ferry are sent to the related geodatabase layers. The developed java codes itself can be found in the Appendix E.

### 7.3 Presentation Layer:

#### 7.3.1 GIS server: GeoServer configuration

User Geoportal is developed by using GeoServer and Openlayer opensource software. It provides an interface between the users and the geosemantic web services.

Geoserver is an open source software server written in Java that allows users to share and edit geospatial data. Designed for interoperability, it publishes data from any major spatial data source using open standards.

GeoServer is the reference implementation of the Open Geospatial Consortium (OGC) Web Feature Service (WFS) and Web Coverage Service (WCS) standards, as well as a high performance certified compliant Web Map Service (WMS). GeoServer forms a core component of the Geospatial Web.

GeoServer allows you to display your spatial information to the world. Implementing the Web Map Service (WMS) standard, GeoServer can create maps in a variety of output formats.

Geoserver Interface which allow us to manage spatial dataset, including adding data, defining coordinate system, style files, publishing geodata and so on. Layer management interface shown in figure 7.3 and 7.4.

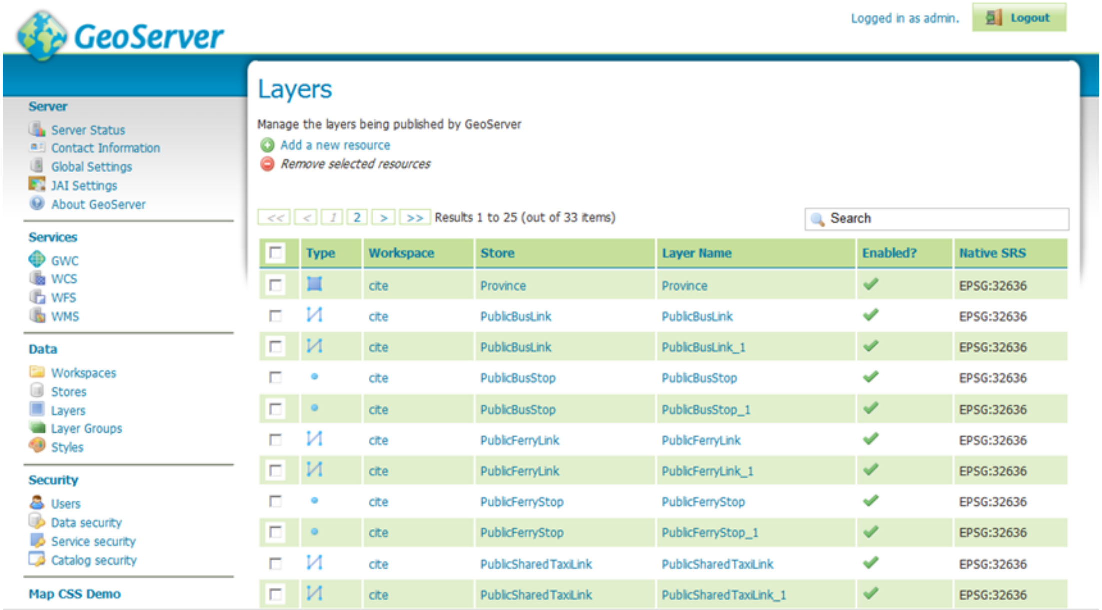



**Figure 7.3 :** Layer management interface.

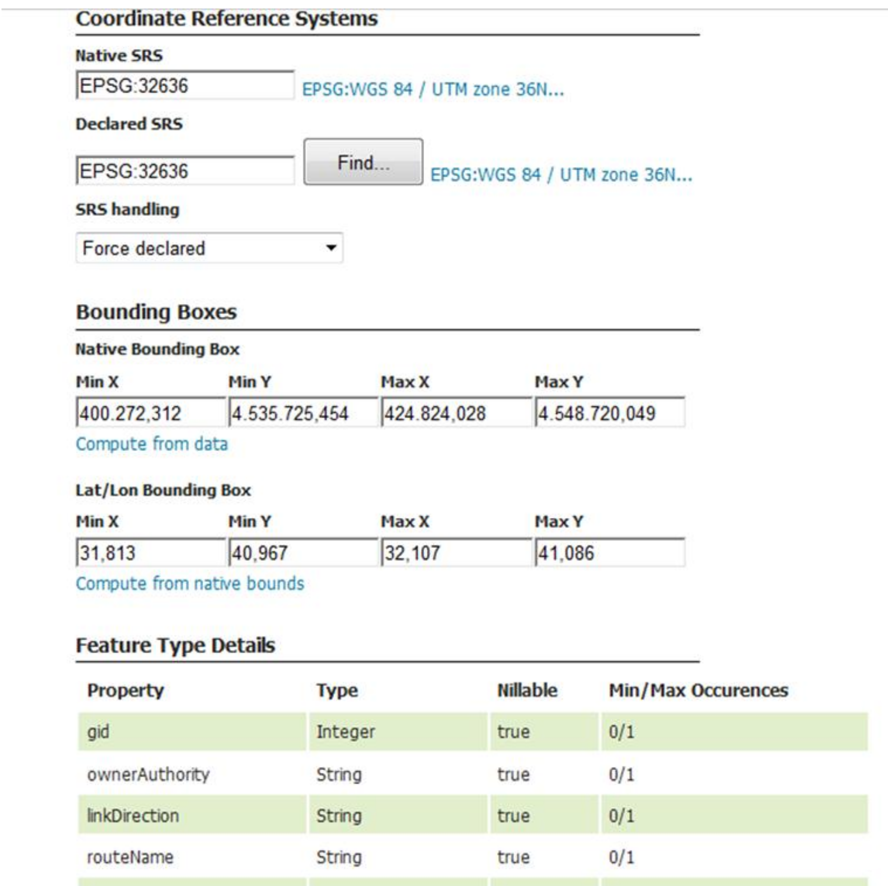



**Figure 7.4 :** Definition of layer properties like coordinate system, bounding box and feature details.

In order to understand determined route alternatives in geodatabase, we have checked the look up field “display” in Route Link and Stop dataset whether it has “1” or “0” value. If it has “1” value, we can publish them on User Geoportal as a result of our semantic queries.

We have prepared a rule in Geoserver Style file to display GIS Route Link and Stop data different than others if it includes “1” value in “display” field (7.9).

```
<sld:FeatureTypeStyle>
  <sld:Rule>
  <Name>highletedlink</Name>
    <ogc:Filter>
        <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>display</ogc:PropertyName>
        <ogc:Literal>1</ogc:Literal>
        </ogc:PropertyIsEqualTo>
    </ogc:Filter>
    <sld:LineSymbolizer>
        <sld:Stroke>
        <sld:CssParameter name="stroke">#FF0000</sld:CssParameter>
        <sld:CssParameter name="stroke-width">1.5</sld:CssParameter>
        </sld:Stroke>
     </sld:LineSymbolizer>
  </sld:Rule>
</sld:FeatureTypeStyle>                                    (7.9)
```

After setting prepared Style files for the all GIS data layers (Bus, SharedTaxi, Ferry Route and Stops), OpenLayer software used to publish the spatial dataset as WMS.

### 7.3.2 OpenLayer: Application user interface development

Our sample Transportation Dataset is already migrated to PostGIS database (explained in section 7.1) and previous section 7.3.1 explains the GIS Server configuration in order to publish these geodata using OpenLayer software.

With the use of OpenLayer software, user would be able to see all GIS layers stored in Geodatabase via the web browser. OpenLayer is completely free, Open Source JavaScript mapping library, integrated into GeoServer, making map generation quick and easy. Used OpenLayer script codes to publish GIS data on the Geoportal given below as an example.

1) Create a map object by defining boundry for map view (bounds), giving same coordinate systems as GIS dataset (EPSG:32636), map format as ‘image/png’ and so on (7.10).

```
<script src="http://openlayers.org/api/OpenLayers.js"></script>

<script defer="defer" type="text/javascript">

var zoom = 5; var map;

function init(){format = 'image/png'; var bounds = new OpenLayers.Bounds(

390002.312, 4529446.454,  430824.028, 4558720.049 );

var options = {controls:[], maxExtent: bounds, maxResolution:
95.90514062500006, projection: "EPSG:32636", units: 'm' };

map = new OpenLayers.Map($('map'), options);
```

**(7.10)**

2) After creating a map object, creating the WMS for publishing GIS dataset. Below example is given for WMS creation for province layer (7.11).

```
province = new OpenLayers.Layer.WMS("Province",
"http://localhost:8080/geoserver/wms",

{layers: 'cite:Province', srs: 'EPSG:32636',

format: format, tiled: true,

tilesOrigin : map.maxExtent.left + ',' + map.maxExtent.bottom,

isBaseLayer: true,

displayInLayerSwitcher: 'FALSE', visibility: true   },

{buffer: 0, displayOutsideMaxExtent: true});
```

**(7.11)**

3) Below example is given for WMS creation for bus link layer (7.12).

```
buslink = new OpenLayers.Layer.WMS(
```

"Bus Route", "http://localhost:8080/geoserver/wms",

{transparent: 'TRUE', layers: 'cite:PublicBusLink', srs: 'EPSG:32636',

format: format, isBaseLayer: false, visibility: true },

{displayInLayerSwitcher: 'TRUE'}); **(7.12)**

4) Below example is given for WMS creation for bus stop layer (7.13).

busstop = new OpenLayers.Layer.WMS(

"Bus Stop", "http://localhost:8080/geoserver/wms",

{transparent: 'TRUE', layers: 'cite:PublicBusStop', srs: 'EPSG:32636',

format: format, isBaseLayer: false, visibility: true },

{displayInLayerSwitcher: 'TRUE'}); **(7.13)**

Likewise, WMS' are created for SharedTaxi Link, SharedTaxi Stop, Ferry Link and Ferry Stop.

5) WMS' added to map object, previously created, to display GIS layers on the Geoportal (7.14).

map.addLayers ([sharedtaxilink, buslink, province, ferrylink, busstop, sharedtaxistop, ferrystop]);

map.zoomToExtent (bounds); **(7.14)**

Coded OpenLayer scripts itself can be found in the appendix F.

Below map in figure 7.5 shows the developed user interface of Geoportal by OpenLayer on GeoServer. It is also possible to add some more functionality to the Geoportal such as info, layer management and etc. by OpenLayer script coding. Different zoom scales are defined for layers. When the user zoom in a certain scale public bus stops will be display, so that readability of map can be increased. Below map shows the example for bus stop zoom scale (figure 7.6).

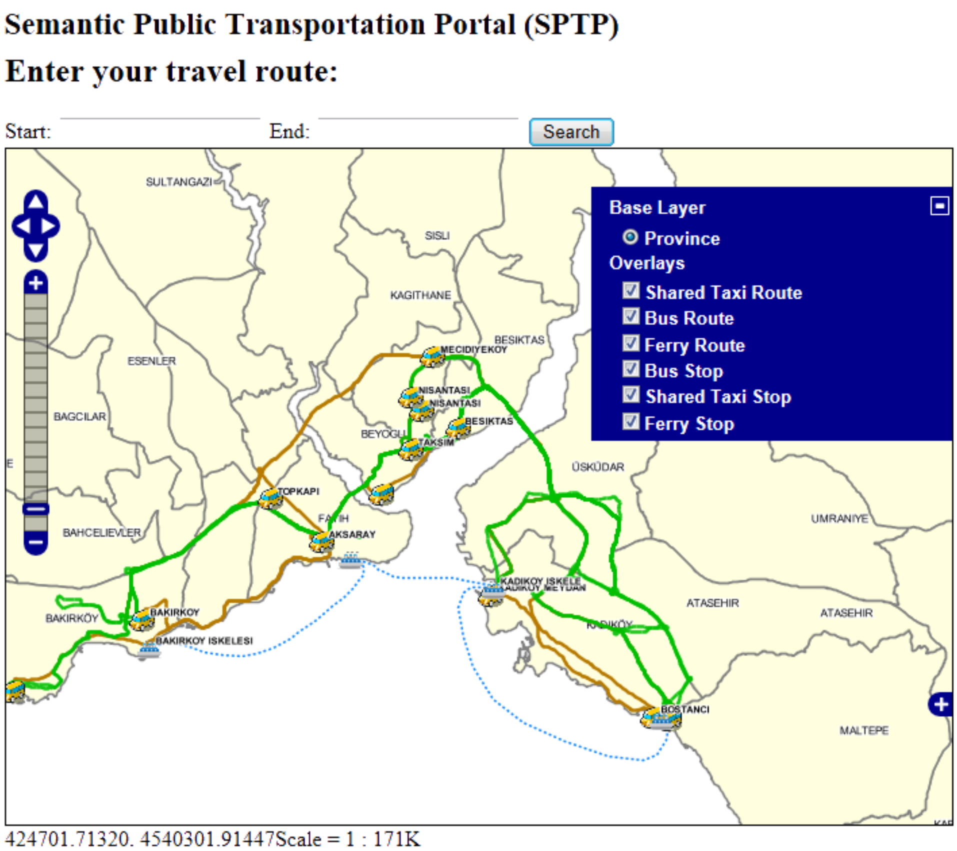



**Figure 7.5 :** User interface of Geoportal: Map view of the dataset by using OpenLayer and Geoserver.

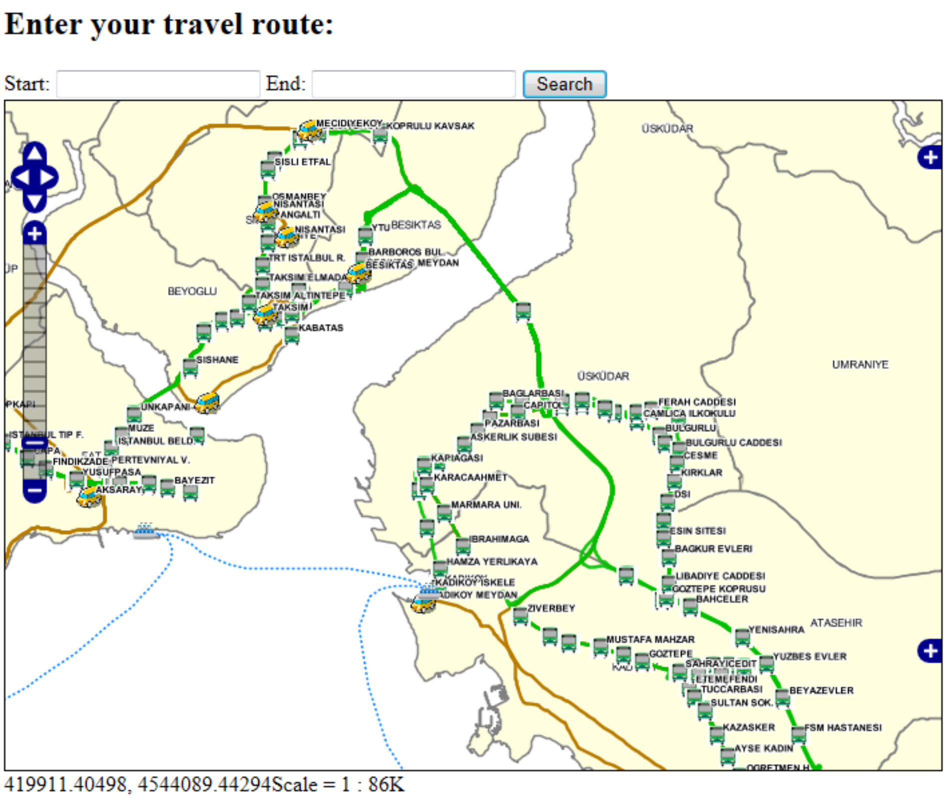



**Figure 7.6 :** Bus Stops appers when the user zoom in acertain scale.

## 7.4 Experimental Investigation

After all, developed Geoportal will allow users to retrieve Bus, SharedTaxi and Ferry Route alternatives based on the entered Start and End Stop Names.

The Geoportal application contains the following main parts to perform semantic queries in order to get public transportation route alternatives

1) The Geoportal provided an interface between the users and the geospatial semantic services to understand spatial and domain knowledge that users needed, and to provide the results of the query for the users.
2) Spatial and domain ontologies (including concepts, relations, and individuals) are organized based on different public transportation alternatives.
3) Reasoning Engine is used to perform semantic queries. Thus, transportation route alternatives are retrieved from the ontology domain.
4) Retrieved transportation routes from Ontology queries, linked to the Geodatabase and published as WMS' on the Geoportal.

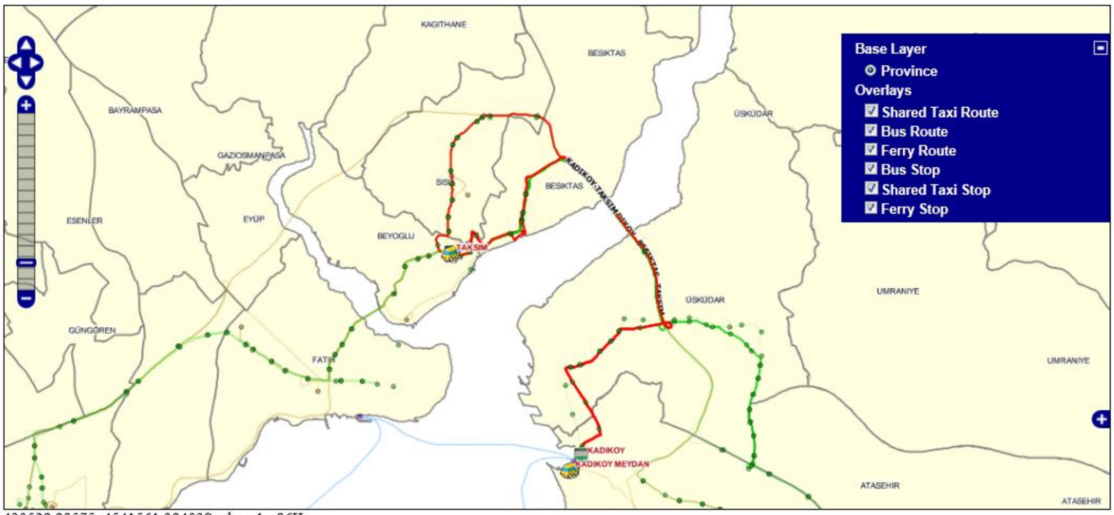
To investigate the thesis approach semantic based search, several cases performed on the developed geoportal.

**Case - 1:** Users indicate the Start Stop: KADIKOY and End Stop: TAKSİM

In according to the given stop names, transportation route alternatives are searched in the transportation ontology domain and retrieved considering the defined concepts and relationships. The result map view shows the transportation routes as highlighted in red color, including bus, sharedtaxi and ferry routes, since the homogeneous transportation system is built. The result also includes relevant spatial concepts and domain knowledge of Routes such as route name, route number, car fare, travel frequency, estimated travel time obtained by the semantic queries, shown directly on the map and listed in an information window to tell the users the knowledge that they are interested in (figure 7.7).

## Semantic Public Transportation Geoportal

Below map shows Bus, Shared Taxi and Ferry Transportation Route alternatives based on your request.




430528.99575, 4541561.39403Scale = 1 : 86K

| **Transportation Route Alternatives** | **Route Information** |
|---|---|
| Bus Route | Route Name: KADIKOY-TAKSIM \|\| Route Number: 110 g \|\| Fare: 2 TL \|\| Frequency: 20 Min. \|\| Estimated Travel Time: 115 Min. |
| Shared Taxi Route | First Route Name: KADIKOY-MECIDIYEKOY-TAKSIM \|\| Route Number: D_6 g \|\| Fare: 6 TL \|\| Frequency: 25 Min. \|\| Estimated Travel Time: 50 Min. Second Route Name: KADIKOY-BESIKTAS-TAKSIM \|\| Route Number: D_5A g \|\| Fare: 6 TL \|\| Frequency: 25 Min. \|\| Estimated Travel Time: 50 Min. |
| Ferry Route | Ferry Route is not found based on your request! |

**Route Alternatives are highlighted (in red color) on the map view**

**Note: Estimated Travel Time and Frequency info calculated based on the time table of transportation systems.**

Search New Travel Route

**Figure 7.7 :** Showing the alternative routes (including Bus, SharedTaxi, Ferry) from Start Stop: KADIKOY to End Stop: TAKSIM.

The user can managed the display option of layers if needed. Below figure 7.8 shows the only Bus Route and Bus Stop layers by using the check box display option of layers.

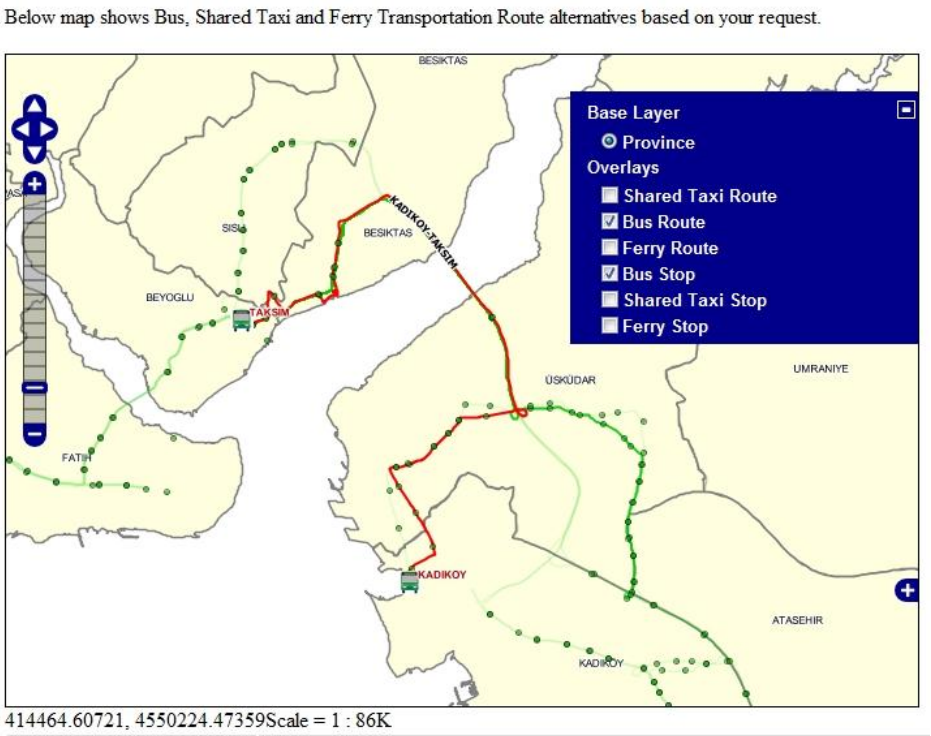



**Figure 7.8 :** Map shows the only Bus Route and Bus Stop layers by using the check box display option.

Likewise, Sharedtaxi Route and Stops can be displayed by itself (shown figure 7.9), so that search results can be seen layer by layer on the map view if needed.

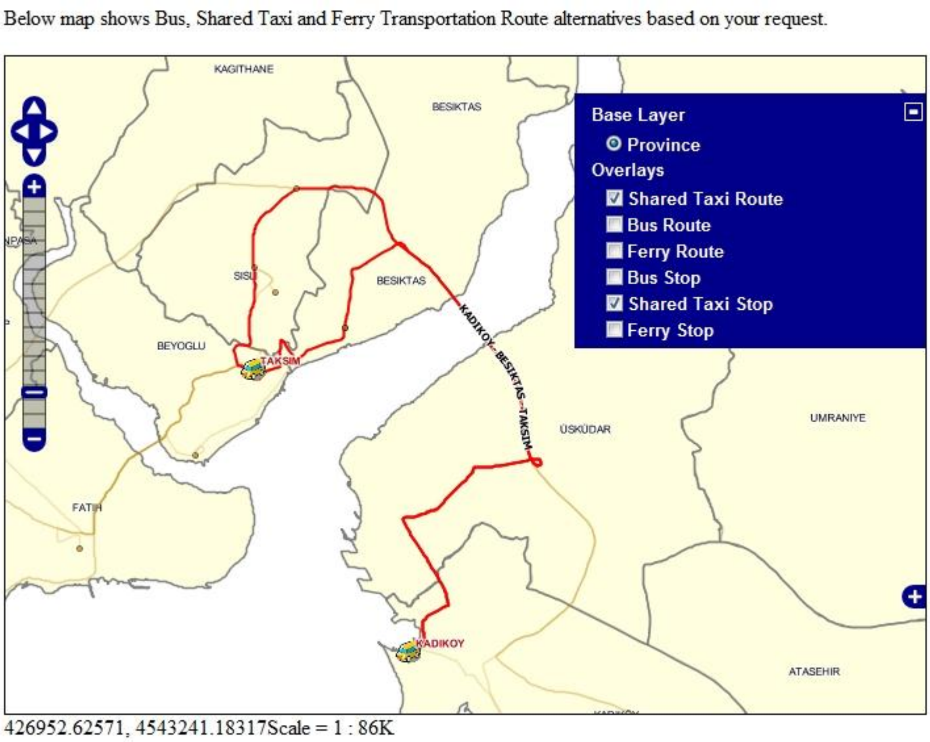



**Figure 7.9 :** Map shows the only Sharedtaxi Route and Stop layers by using the check box display option.

**Case - 2:** Users indicate the Start Stop: KADIKOY ISKELE and End Stop: TAKSİM

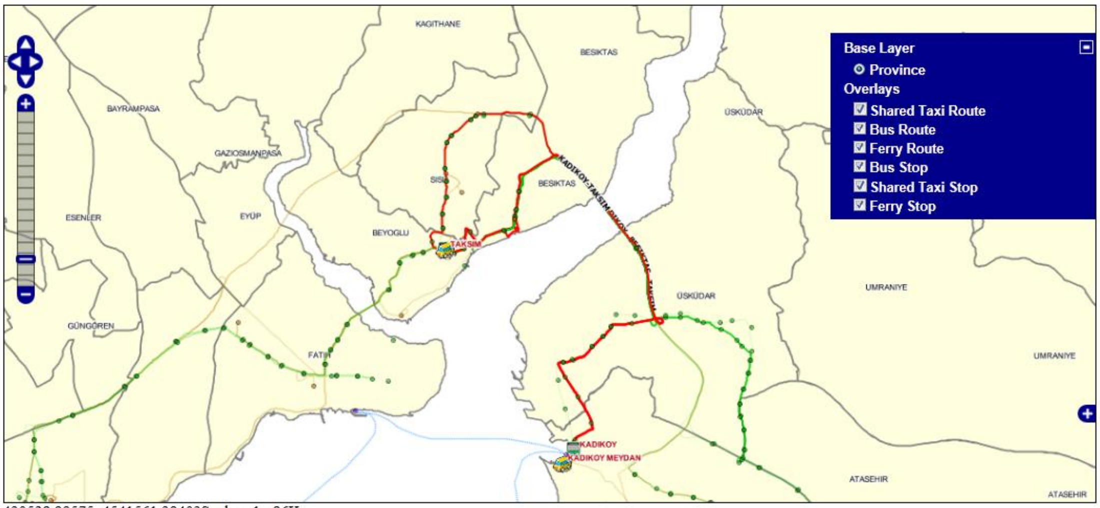
In this case, semantic heterogeneity caused by Synonyms are investigated for the transportation stops. Means of transport may pass the same place but with different stop names. Moreover, same station name may appear at different locations. The developed system query with more semantic information to overcome such synonymic problems occurred in stop names.

As for this case result, when the user enter even KADIKOY ISKELE or KADIKOY stop name, ontology search will be performed for both transportation routes sharedtaxi and bus. Because, KADIKOY ISKELE is the stop of SharedTaxi and KADIKOY is the stop of Bus are defined as **Synonyms** in the Ontology Domain.

Below figure 7.10 showing the alternative routes (including Bus, SharedTaxi, and Ferry) from Start Stop: KADIKOY ISKELE to End Stop: TAKSIM by considering Synonyms.

## Semantic Public Transportation Geoportal

Below map shows Bus, Shared Taxi and Ferry Transportation Route alternatives based on your request.




430528.99575, 4541561.39403Scale = 1 : 86K

| Transportation Route Alternatives | Route Information |
|---|---|
| Bus Route | Route Name: KADIKOY-TAKSIM \|\| Route Number: 110 g \|\| Fare: 2 TL \|\| Frequency: 20 Min. \|\| Estimated Travel Time: 115 Min. |
| Shared Taxi Route | First Route Name: KADIKOY-MECIDIYEKOY-TAKSIM \|\| Route Number: D_6 g \|\| Fare: 6 TL \|\| Frequency: 25 Min. \|\| Estimated Travel Time: 50 Min. Second Route Name: KADIKOY-BESIKTAS-TAKSIM \|\| Route Number: D_5A g \|\| Fare: 6 TL \|\| Frequency: 25 Min. \|\| Estimated Travel Time: 50 Min. |
| Ferry Route | Ferry Route is not found based on your request! |

**Route Alternatives are highlighted (in red color) on the map view**

**Note: Estimated Travel Time and Frequency info calculated based on the time table of transportation systems.**

Search New Travel Route

**Figure 7.10 :** Showing the alternative routes (including Bus, SharedTaxi, Ferry) from Start Stop: KADIKOY ISKELE to End Stop: TAKSIM by considering Synonyms based on the query request.

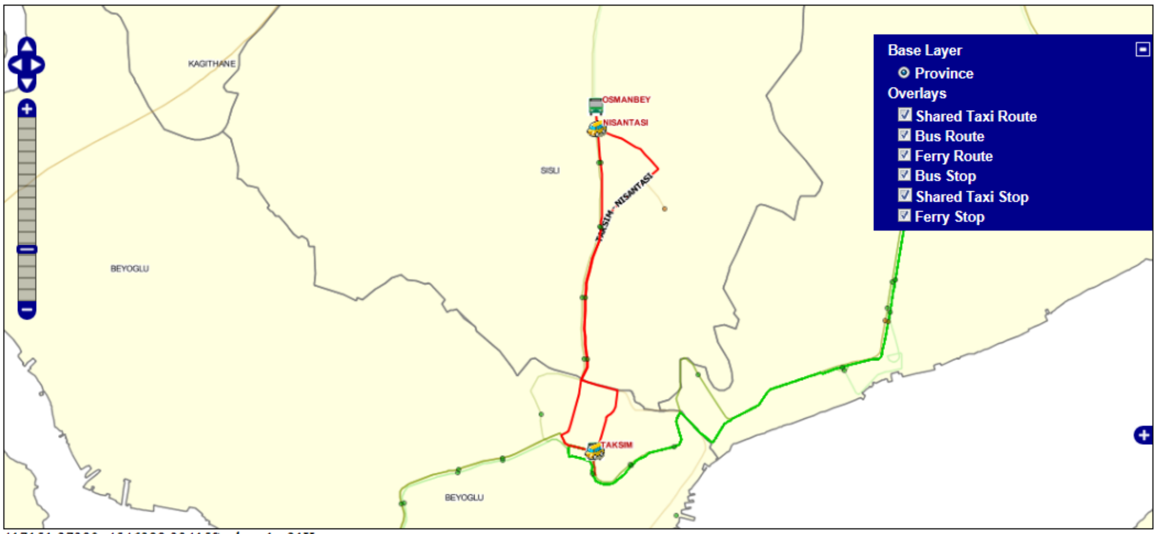
**Case - 3:** Users indicate the Start Stop: TAKSIM and End Stop: NISANTASI

Some of the citizens can define the travel route by giving the name of the known area instead of the exact name of the stop names. For example, Taksim and Nişantaşı are the very well-known areas for the citizens. In such cases, user should have possibility of search travel route based on the indicated place names even if these place names are not defined in the geodatabase. In this regard, semantic query algorithms are implemented and such place names are defined in ontology domain as synonyms with the closest stop names to overcome such ambiguity in the dataset.

Below figure 7.11 shows the alternative routes for the indicated stops although the NISANTASI place name is not existed in the transportation geodatabase as a stop.

**Semantic Public Transportation Geoportal**

Below map shows Bus, Shared Taxi and Ferry Transportation Route alternatives based on your request.




417151.27390, 4546399.33415Scale = 1 : 21K

| Transportation Route Alternatives | Route Information |
|---|---|
| Bus Route | Route Name: USTBOSTANCI-TAKSIM \|\| Route Number: 202 d \|\| Fare: 2 TL \|\| Frequency: 20 Min. \|\| Estimated Travel Time: 16 Min. |
| Shared Taxi Route | Route Name: TAKSIM-NISANTASI \|\| Route Number: D_58 g \|\| Fare: 4 TL \|\| Frequency: 20 Min. \|\| Estimated Travel Time: 15 Min. |
| Ferry Route | Ferry Route is not found based on your request! |

**Route Alternatives are highlighted (in red color) on the map view**

**Note: Estimated Travel Time and Frequency info calculated based on the time table of transportation systems.**

Search New Travel Route

**Figure 7.11 :** Showing the alternative routes from Start Stop: TAKSIM to End Stop: NISANTASI.

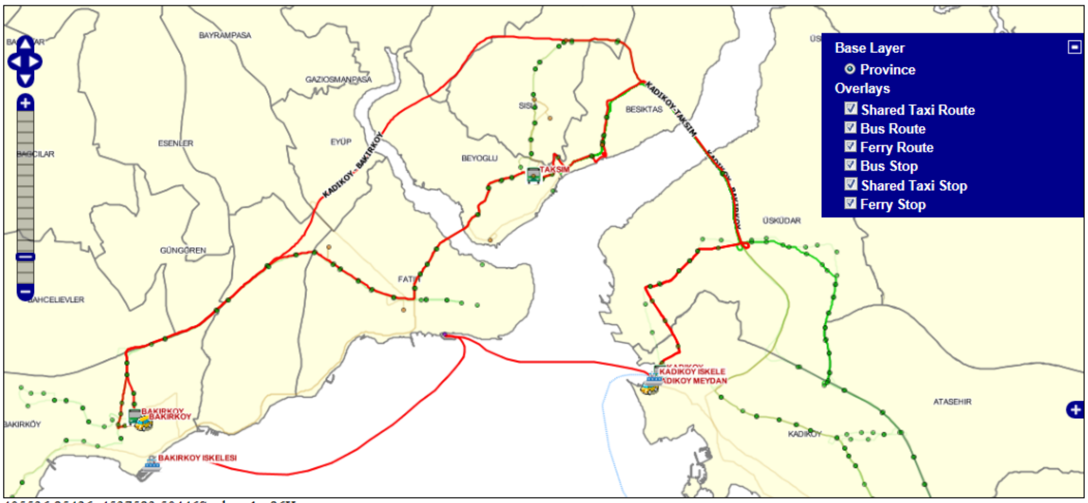
**Case - 4:** Users indicate the Start Stop: KADIKOY and End Stop: BAKIRKOY

The developed transportation algorithm also considers the one transfer option in its own transportation for the means of bus, sharedtaxi or ferry. The algorithm first attempts to search for no-transfer travel. If no zero transfer travel is found, it seeks for potential one-transfer travel in the transportation domain.

Below figure 7.12 shows the alternative routes (including Bus, SharedTaxi, and Ferry) from Start Stop: KADIKOY to End Stop: BAKIRKOY considering transfer possibility. The search result presents the bus route as one transfer travel. Passenger needs to take KADIKOY to TAKSIM bus at first and then needs to change at first intersection stop (which is highlighted on the map as TAKSIM) to take second bus line.

## Semantic Public Transportation Geoportal

Below map shows Bus, Shared Taxi and Ferry Transportation Route alternatives based on your request.




405536.85426, 4537583.50446Scale = 1 : 86K

| Transportation Route Alternatives | Route Information |
|---|---|
| Bus Route | First Route Name: KADIKOY-TAKSIM \|\| Route Number: 110 g \|\| Fare: 2 TL \|\| Frequency: 20 Min. \|\| Estimated Travel Time: 115 Min. Second Route Name: ATAKOY-TAKSIM \|\| Route Number: 71T d \|\| Fare: 2 TL \|\| Frequency: 20 Min. \|\| Estimated Travel Time: 119 Min. |
| Shared Taxi Route | Route Name: KADIKOY-BAKIRKOY \|\| Route Number: D_11 g \|\| Fare: 9 TL \|\| Frequency: 20 Min. \|\| Estimated Travel Time: 110 Min. |
| Ferry Route | Route Name: BOSTANCI-BAKIRKOY \|\| Route Number: Line-1 \|\| Fare: 7 TL \|\| Frequency: 45 Min. \|\| Estimated Travel Time: 32 Min. |

**Route Alternatives are highlighted (in red color) on the map view**

**Note: Estimated Travel Time and Frequency info calculated based on the time table of transportation systems.**

Search New Travel Route

**Figure 7.12 :** Showing the alternative routes (including Bus, SharedTaxi, Ferry) from Start Stop: KADIKOY to End Stop: BAKIRKOY considering transfer possibility.

# 8. CONCLUSIONS AND RECOMMENDATIONS

## 8.1 Conclusions

As stated in the introduction, interoperability is the important key for the today's geographic information systems. The geographic information is usually available as datasets stored in databases and accessible via GISs. However, these GISs and databases are often more heterogeneity, each of them created for one specific purpose and shows differences in modeling, representation, and accessing of sources.

Problems related to heterogeneity have been categorized previously:

1) Syntax (related to different data formats),
2) Structure (related to differences in schemas), and
3) Semantics (related to the differences in intended meaning of terms in specific contexts).

To satisfy increased demands for the use, sharing, and exchange of geographic data in the common applications, methods to support interoperability are required by the GIS community. In recent years, the focus has started to shift from managing several stand-alone geoinformation datasets to harmonized management of interoperable GISs. Through interoperability and standardization, these harmonized networks improve the availability, accessibility and usability of geographic data. Such a facility that enables the interoperability and harmony between geographic datasets is the Spatial Database Infrastructure (SDI). It improves availability, accessibility and usability of geographic data "for all users within all levels of an organization. The development of SDI, such as the European INSPIRE, is necessitating existing spatial data standards to be harmonized or revised, based on modern international standards to overcome the interoperability issues.

In addition to this, researches in geospatial interoperability must go beyond geometry-related and database-structure concerns to take into account of semantics heterogeneity. In traditional GIS systems, users can formulate queries using keywords or/and spatial filters. However, this approach falls short to fully express the user needs, while the information sources are suffer from the semantic heterogeneity. Semantic heterogeneity problems caused by mainly the reasons as follows:

- Homonyms: naming conflicts occur when classes or attribute types with different semantics are given the same names or
- Synonyms: when classes or attribute types that are semantically the same are named differently.

If the terms are synonyms and/or homonyms in the dataset, keyword-based spatial search inherently restricted by the ambiguities of natural language. Different terms may refer to similar concepts and the same terms may refer to different concepts. In this context, the promising approach is that implementing the geospatial semantic by means of ontology on the geospatial dataset to overcome this kind of ambiguities. The relevant concepts and relations, which are reasoned by inferred semantics, are discovered in SDI to search suitable GIS data and services.

The aim of this research is to investigate the implementation of INSPIRE as a SDI standard and Semantic Technology to empower the traditional GIS approach. In this regard, a public transportation geoportal is developed which uses the revised sample transportation data complaint with INSPIRE Transport Network Data Theme and the Transportation Ontology Domain, including concepts, relationships and individuals to provide semantic spatial search.

The methodology of this research firstly applied for the existing transportation dataset to harmonized based on the INSPIRE Transport Data Theme to meet the interoperability in European SDI Level. With the use of the UML object-oriented schematic language, the new transportation data model is structured by following the guidelines explained in the INSPIRE Transport Network Data Specification. The UML formalisms give the possibility to describe a data-model in a visual and schematic way and can be interpreted by the standardized meanings of the symbols and relations. This technical UML model also serves as input for the automatic Geodatabase generation. In this study, conversion of the UML schema into the Transportation Geodatabase is carried out by using the combination of Microsoft Visio and ArcCatalog from ESRI tools.

Microsoft (MS) Visio CASE Tool is used to design the Transportation Data Model in UML schematic language. In particular, in the case of the Transportation Geodatabase, several diagrams able to represent the hierarchical structure of the database are proposed. The term “hierarchical” means that the Geodatabase is constituted by “feature datasets” containing in turn “feature classes”, the latter being characterized by attributes. Each feature dataset has been modeled as a UML package. Inside the packages are present the UML feature classes, with the relevant attributes, modeling the feature classes.

The Transportation UML Schema is designed as Administrative_Units, Transport_Networks feature datasets. Province and District feature classes is created inside of the Administrative_Units feature dataset. Transport Network feature dataset is designed by including the feature classes of Road Link, Waterway Link, Road Node, Waterway Node, and also Public Transportation Route and Stop Features for Bus, Sharedtaxi, Ferry classes.

Road and Waterway Link feature classes are derived from the CommonTransportLink abstract class and node classes are derived as defined from the CommonTransportNode abstract as defined in INSPIRE Transport Network Data Specification. As for the creation of Public Transportation Route Classes, Common TransportLink Sequence abstract class is used since it defines “a link sequence may represent a route” based on the INSPIRE Transport Network Theme. Public Transportation Stops (bus, sharedtaxi and ferry) are derived from the CommonTransportPoint class as referred to the INSPIRE Transport Network Specification as well. Other attributes refer to range of values have been modeled as <<coded and range>> stereotype domains compliant with the INSPIRE Transport Network Domains.

After the UML schema is created in MS Visio according to the structure of UML and the rules set up by the ESRI-template, the schema can be exported to an XMI-file, using add-on functionality for MS Visio. XMI is a metadata format that enables interchange of data and metadata between tools, middleware and repositories. It links modeling-formats such as UML with metadata formats, such as XML.

The last steps of the UML to Geodatabase conversion take place by means of ArcCatalog functionality. ArcCatalog Schema Generator Tool is used to read this

XMI file and generate Transportation Network Geodatabase based on the UML design.

In addition to this, the ontology development methodology is used to describe geospatial public transportation network semantically during this research, which allows for automated reasoning and intelligent spatial queries. In this context, an ontology domain is established base on the Transport Network Schema, which can describe the semantic knowledge, concepts, and relations of transportation means – bus, sharedtaxi and ferry. In order to make use of these knowledge effectively, we have designed the knowledge base system based on the OWL ontologies and rules with the help of Protégé Ontology Editor and RacerPro Reasoning Engine.

In this study, OWL ontology consists of Concepts, Properties and Individuals (Classes) are developed for the Public Transportation Network. The concepts mainly involve Transportation Route, Transportation Link and Transportation Stop in the ontology domain. These concepts are further divided into sub concepts. Transportation Route is divided into Bus, Sharedtaxi, Ferry Route sub concepts. Likewise, Transportation Link and Stop concepts are divided into Bus, Sharedtaxi, Ferry Link and Stop sub concepts, which are taxonomically related by the transitive "is-a" relation.

Once we have model the concepts of the ontology, relationships are defined between these concepts and restrict their interpretation using properties. There are two main types of properties, object properties and datatype properties, which represent relationships in OWL ontology.

Four main object property relations are created in this ontology domain: hasLink, hasStop, hasStartStop, and hasEndStop. Each object property has also a corresponding inverse property, which enables to infer implicit taxonomic relationships and equalities automatically.

In order to define relationships between concepts in ontology restrictions needs to be declared via the object property. For example, an existential restriction describes a class of individuals that have at least one (some) relationship along a specified property to an individual that is a member of a specified class. According to this, Public Bus Route and Public Bus Link relationship is described through the object property hasLink with "some" existential restriction. In addition, we have modeled the Bus Link have exactly one Start Stop and one EndStop, and we use the "exactly=1" cardinality restriction through the object property to define this relationship. Likewise, the other relationships are described as explained in the chapter 6.2.

Afterwards, we add the instances we decided as individuals of the ontology and relate those using datatype properties. Datatype properties describe the relationships between an individual and data values in OWL ontology. In this ontology domain, datatype properties are used to describe some facet of the concept or entity, like the price for travel ticket, time frequency for departure, estimated time of the travel and so on.

The obtained Public Transportation Ontology Domain is used to discover the implicit rich semantics knowledge, which optimize the query algorithm and enhance the query efficiency. To retrieve related explicit and implicit information in the context of designed relevance model, nRQL semantic query language of the RacerPro Reasoning Engine is used.

Semantic spatial query algorithms are developed in order to retrieve transportation route alternatives based on the user requested start and end stop points. The algorithm first attempts to search for no-transfer travel. If no zero transfer travel is found, it seeks for potential one-transfer travel in the transportation domain.

The several transportation means may pass by the same place but with different stop names or same stop name may appear at several different places. The developed transportation algorithm considers the synonym and/or homonym heterogeneity for querying between two stop so that this sort of ambiguities are overcome.

In addition to this, round-travel directions (From-To) can be different in the transportation system. For example, from-to direction of the bus routes are not exactly the same line with to-from direction of the bus route and they have even different stop names along the round- travel. The developed transportation algorithm is also considers the condition that round-travel directions are different in the transportation system. It is also possible to retrieve data values with the use of developed algorithms such as carfare, time frequency of the travel, estimated travel time and so on.

As a result, homogeneous reasoning system is obtained by covering the transportation modes bus, sharedtaxi and ferry in the same transportation system during this research. The result is presented on the developed geoportal for the experimental investigation that provides a knowledge-based search (semantic search) and OGC WMS service to publish obtained result based on the given stops by user.

Experimental investigation shows that the developed ontology domain express the rich semantic knowledge of three dimension transporting means (bus, shared taxi, ferry) in formalization and related administrative areas. The implementation of the geo-ontology can be a promising approach to overcome the semantic heterogeneity in the spatial dataset such as synonymic problems on stop names and differentiating the round-travel directions. In this regards, the semantics embedded in ontologies offer significant benefits to traditional GIS systems by elimination of deficiencies in the keyword-base spatial query approach. In addition to this, a standardized transportation geodatabase compliant with the INSPIRE is used for this investigation. Thus, schematic and structural interoperability are derived from the INSPIRE, by means of European SDI level, for existing transport network dataset.

## 8.2 Recommendations and Future Work

During this reasearch study, three-tier system architecture is implemented and the investigation is carried out with the use of feature dataset, include the number of 3658 Bus Links, 244 Bus Stops, 12 SharedTaxi Links, 15 SharedTaxi Stops, 3 Ferry Links and 4 Ferry Stops. Also, the transportation ontology domain, including concepts, relations and individuals considering this transportation geodatabase are used in the developed system. The preliminary verification of experimental investigation shows that the developed geoportal system perfomance is good for performing semantic search and publishing the achieved transportation route alternatives. However, working with the large feature dataset and ontologies, especially when the large set of concepts and individuals are defined, can impact the application performance and can cause the slower query response. In that case, system architecture needs be designed considering the used dataset and functionality, for example distributed web server architecture can be implemented if needed.

In this research, many methods and technics are gained about the implementation of geo-ontologies and the harmonization of INSPIRE data model specification. In the design and harmonisation process of a dataset, there is a need for complete, unambiguous and solid functional specifications. If these are not clear, it is not clear what possibilities and functions the new system should have. In this investigation, transport database harmonization process is implemented by the guidelines of INSPIRE Data Specification on Transport Networks.

The UML as structured design is used for the harmonisation process. Since UML is a standardised schematic language it can be interpreted more easily than a personally created chart with an explanation report on the side. On of the possibilities is to have UML schemas translated into geodatabases. Microsoft Visio CASE Tool and ArcCatalog Schema Wizard Tool are investigated for the creation of geodatabase from the designed UML Schema during this study. It proved to be functioning properly, but there may be good alternative processes and softwares as well.

Since more and more organisations are applying GIS and are moving from older versions to new modern GIS and database environments, the focus should move from designing completely new systems to systems based on a legacy. This is often done through combining datasets, which can be referred to as a harmonisation process. The used harmonisation method in this research can also be applied to other distributed geographic datasets. Probably the model will need only minor adjustments to make it applicable in other organisations.

As for the ontology development, we realized that some considerable work is required to derive a ontology domain, since it relies on a heuristic strategy; starting with a set of single assertions and going through more complex expressions. The development of transportation ontology domain so far is carried out semi-automatically by the use of Protégé ontology editor and RacerPro reasoning engine.

In the future work, we would like to adopt automation ontology development tool which reads the feature dataset and generate the ontology domain according to the indicated rules. In this way, we expect to speed up the ontology development process and include the more instances to enrich the ontology considering the heuristic strategy. The big challenge here is to need a transportation domain expert to define the initial concepts and relationships during the ontology development process. Also, feedbacks and guidance of and domain expert/user can be needed in order to raise the level of accuracy of the achieved ontology domain.

In addition to the future work, we would like to enrich the public transportation system with including the other transportation means such as subway, tram, minibuses and so on. In this way, there are more transportation alternatives can be provided for the travelers. Although it needs some operational effort to create new transportation features and adding new individuals in ontology domain, the developed system is implemented as modular so that new transportation features can be adopted easily considering the generic semantic algorithms, ontology domain and other relations.

Also, in a big city it is very difficult to access proper information on bus routes and their schedules for a traveler. Weather and traffic jams can render printed bus time tables useless or of little use. One other way to enhance the current public transportation system is to provide transportation services with the real time data such as the current location of the user, bus stops and running busses. Therefore users can take the suitable route and predict their arrival times more accurately, considering the traffic info etc.

Such functionality may be implemented through the support of Location Based Services (LBS). A LBS is a wireless IP-based service that uses geographic information to serve a mobile user. Generally, the sources of location information can be Global Positioning System (GPS) and location collected from mobile communication technologies like GPRS, UMTS, WLAN and 3G/4G network.

To achieve the real time current location of the transportation means like bus, a vehicle tracking system is needed which incorporates a hardware device installed in the vehicle and a remote tracking server. Tracking system using GPS for positioning information and GSM/GPRS modem that receives vehicle location information via GSM network and stores this information in database. The location information (latitude, longitude) of a vehicle is transmitted to Tracking server using GSM/GPRS modem on GSM network by using SMS or using direct TCP/IP connection with Tracking server through GPRS. This information is available to authorized users of the system via website over the internet. These systems generally available in the Transportation Authorities to track their vehicles. We can extend the current public transportation system by integrating the tracking system to provide real time current location of running busses for the users. In order to provide these kind of real-time informations into the developed system, we need to work in cooperation of the responsible transportation authority to use their tracking services.

As for the detection of real time user location, we can locate a user's position by creating a location-aware web page to make a user share their location from using their mobile device, which in this case, we can use Geolocation API. The implemented transportation system arhitecture GIS client/server interfaces like WMS can also be applied for mobile services and supports Geolocation API. The usage of standardized and moduler transportation system and therewith the avoidance of proprietary developments leads to an open structure of the GIS platform for the future works.

## REFERENCES


**ADSIC.** (2009). *AD-SDI Data Content Standard,* Retrieved from http://sdi.abudhabi.ae/Sites/SDI/Content/EN/PDF/GeospatialDataStandards/sdi-data-content-standard-reference-manuel,property=pdf.pdf

**Albrecht, J., Derman, B. and Ramasubramanian, L.** (2008). Geo-ontology Tools: The Missing Link, *Transactions in GIS*, Vol. 2, No. 4, pp. 409-424.

**Amirian, P., Alesheikh, A. A.** (2008). Implementation of a Geospatial Web service Using Web Services Technologies and Native XML Databases, *Middle-East Journal of Scientific Research,* Vol. 3, No. 1, pp. 36-48.

**Aydın, G.** (2007). Service Oriented Architecture for Geographic Information Systems Supporting Real Time Data Grids, *PhD Thesis,* Department of Computer Science, Indiana University, USA.

**Baader, F., Horrocks, I. and Sattler, U.** (2003). Description Logics as Ontology Languages for the Semantic Web, *In Festschrift in honor of Jörg Siekmann,* LNAI, Springer-Verlag, pp. 228-248.

**Bode, C.** (2006). An Ontology-Based Repository for Web Services, *Diplomarbeit,* Universitat München, Munich, Germany.

**Boucher, S., Zimányi, E**. (2010). An Ontology-Based GeoDatabase Interoperability Platform, *Cases on Semantic Interoperability for Information Systems Integration: Practices and Applications*, Chapter 14, pp. 294-315, IGI Global Publisher.

**Brentjens, T.** (2004). OpenGIS Web Feature Services for editing cadastral data, *Master of Science Thesis,* Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, Netherlands.

**Číhalová, M., Ďuráková, D. Hrubá, L. and Rapant, P.** (2008). Building Of Traffic Infrastructure Ontology For Mas, *International Symposium GIS Ostrava 2008,* Ostrava, Czech Republic, January 27-30.

**de Vries, W. T., Crompvoets, J., Stoter, J. and Vandenberghe, I.** (2011). Atlas of INSPIRE – Evaluating SDI Development through an Inventory of INSPIRE Experiences of European National Mapping Agencies, *International Journal of Spatial Data Infrastructures Research,* Vol. 6, pp. 126-144.

**Fan, Z. and Zlatanova, S.** (2011). Exploring Ontology Potential In Emergency Management, *Applied Geomatics,* Vol. 3, No. 2, pp. 109-122.

**Ganapathy, G. and Sagayaraj, S.** (2011). To Generate the Ontology from Java Source Code, *International Journal of Advanced Computer Science and Applications,* Vol. 2, No. 2, pp. 111-116.

**Gone, M. and Schade, S.** (2008). Towards Semantic Composition of Geospatial Web Services – Using WSMO in Comparison to BPEL, *International Journal of Spatial Data Infrastructures Research,* Vol. 3, pp. 192-214.

**Haarslev, V., Hidde, K., Möller, R. and Wessel, M.** (2011). The RacerPro Knowledge Representation and Reasoning System, *Semantic Web Journal,* Vol. 3, No. 3, pp. 267-277.

**Hanzlova, M., Vanis, P. and Fabian, P.** (2011). SDI For Nature Conservation: The Nature-SDIplus Data Harmonisation Approach, *Symposium GIS Ostrava 2011,* Ostrava, Czech Republic, January 24-26.

**Houda, M., Khemaja, M., Oliveira, K. and Abed, M.** (2010). A Public Transportation Ontology To Support User Travel Planning, *2010 Fourth International Conference on Research Challenges in Information Science (RCIS),* Nice, France, May 19-21.

**INSPIRE Data Specification on Transport Networks.** address: <http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/INSPIRE_Data Specification_TN_v3.0.pdf>, date retrieved: 07.09.2012.

**INSPIRE Directive.** date retrieved: 07.09.2012, address: **<**http://inspire.jrc.ec.europa.eu>

**Introduction to CASE tool.** date retrieved: 07.09.2012, address: <http://resources.arcgis.com/content/building-geodatabases-case-tools>

**Jung, C. T. and Sun, C. H.** (2010). Ontology-driven Problem Solving Framework for Spatial Decision Support Systems, *GSDI 12 World Conference,* Singapore, October 19-22.

**Klien, E. M.** (2006). Automating the Semantic Annotation of Geodata, *Workshop on Mastering the Gap: From Information Extraction to Semantic Representation,* Budva, Montenegro, June 11-14.

**Klien, E. M.** (2008). Semantic Annotation of Geographic Information, *PhD Thesis,* Universitat Münster, Germany.

**Knublauch, H., Musen, M. A., Rector, A. L.** (2004). Editing Description Logic Ontologies with the Prot´eg´e-OWL Plugin, *International Workshop on Description Logics,* Whistler, BC, Canada.

**Kovarova, A. and Szalayova, L.** (2010). Semantics in the Field of Widgets: A Case Study in Public Transportation Departure Notifications, *Semantics in Adaptive and Personalized Services* (Vol. 279, pp. 93-107). Springer Berlin Heidelberg Publisher, Germany.

**Lemmens, R.** (2006). Semantic interoperability of distributed geo-services, *Publications on Geodesy 63,* Netherlands Geodetic Commission, Delft, Netherlands.

**Leon, L. M. P.** (2006). Ontology mapping for discovery of geographical services, *Master of Science Thesis,* International Institute For Geo-Information Science And Earth Observation, Enschede, Netherlands.

**Lorenz, B., Ohlbach, H. J., Kim, L. Y.** (2005). Ontology of Transportation Networks, Department of Computer Science, University of Munich, Retrieved from http://rewerse.net/deliverables/m18/a1-d4.pdf

**Malik, Z., Rezgui, A., Medjahed, B., Ouzzani, M. and Sinha, A. K.** (2010). Semantic Integration In Geosciences, *International Journal of Semantic Computing,* Vol. 4, No. 3, pp. 301-330. World Scientific Publishing Company, Singapore.

**Niaraki, A. S. and Kim, K.** (2009). Ontology based personalized route planning system using a multi-criteria decision making approach, *Expert Systems with Applications 36,* pp. 2250-2259.

**Noy, N. F. and McGuinness, D. L.** (2001) *Ontology Development 101: A Guide to Creating Your First Ontology,* Retrieved from: http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html.

**Pornsamutsin, Y.** (2010). Using WFS-T in combination with geolocation for online transactional editing of geographic features through mobile browsers, *Master of Science Thesis,* Department of Computer Science and Media Technology, Gjøvik University College, Gjøvik, Norway.

**Portele, C.** (2009). INSPIRE Drafting Team "Data Specifications, *INSPIRE & GSDI Conference,* Rotterdam, Netherlands, June 15-19.

**Prasanna, V. and Bakshi, A.** (2010). Integrated Modeling and Simulation Framework for Freight Transportation in Metropolitan Areas, *METRANS Project Final Report,* University of Southern California, Los Angeles, California, USA.

**Protégé.** date retrieved: 10.11.2012, address: <http://protege.stanford.edu/>

**Racer Systems.** date retrieved: 10.11.2012, address: <http://www.racer-systems.com/>

**Sahin, K. and Gumusay, M. U.** (2008). Service Oriented Architecture (SOA) Based Web Services for Geographic Information Systems, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Vol. XXXVII, Part B2, pp. 625-630.

**Samper, J. J., Tomás, V. R., Martinez, J. J. and van den Berg, L.** (2006). An Ontological Infrastructure for Traveller Information Systems, *IEEE Intelligent Transportation Systems Conference,* Toronto, Canada, September 17-20.

**Sayar, A., Pierce, M. and Fox, G. C.** (2005). Developing GIS Visualization Web Services For Geophysical Applications, *ISPRS Workshop Commision II WG/2,* Ankara, Turkey, November 24-25.

**Schade, S. and Lutz, M.** (2010). Opportunities and Challenges for Using Linked Data in INSPIRE, *Workshop On Linked Spatiotemporal Data 2010,* Zurich, Switzerland, September 14-17.

**Shen, Z., Hu, F., Kawakami, M.** (2011). Personalized Route Planning Based on the Semantic Network: a Case Study of Kanazawa City, Japan, *The 19th International Conference on GeoInformatics,* Shanghai, China, June 24-26.

**Sherimon, P. C., Vinu, P. V. and Krishnan, R.** (2011). Development Phases of Ontology for an Intelligent Search System for Oman National Transport Company, *International Journal of Research and Reviews in Artificial intelligence (IJRRAI),* Vol. 1, No. 4, pp. 97-101. Science Academy Publisher, United Kingdom.

**Storgaard, L. E.** (2012). INSPIRE in Denmark and its impact on the Danish eGovernment, *FIG Working Week 2012,* Rome, Italy, May 6-10.

**Sudra, P. W.** (2010). INSPIRE-compliant web services, *Master of Science Thesis,* Geographical Information Management and Applications (GIMA), Netherlands.

**Sun, C. H., Jung, C. T., Kuo, F. Y., Lee, W. K., Lien, M. F., Hung, K. C.** (2009). Development of an Ontology-based Service Oriented Architecture Framework for National Geographic Information System in Taiwan, *GSDI 11 World Conference - Spatial Data Infrastructure Convergence: Building SDI Bridges to Address Global Challenges,* Rotterdam, Netherlands, June 15-19.

**Turhan, A. Y.** (2011). Description Logic reasoning for Semantic Web Ontologies, *International Conference on Web Intelligence, Mining and Semantics,* Sogndal, Norway, May 25-27.

**Url-1** *<http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/ INSPIRE TechnicalArchitecture Overview_v1.2.pdf>*, date retrieved 07.09.2012.

**Url-2** *<http://www.obitko.com/tutorials/ontologies-semantic-web/resource description-framework.html>,* date retrieved: 17.10.2012.

**Url-3** *<http://www.opengeospatial.org>*, date retrieved: 12.11.2012.

**Url-4** *<http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/Data Specifications/D2.3_Definition_of_Annex_Themes_and_scope_v3.0.p df>*, date retrieved: 07.09.2012.

**Url-5** *<http://inspire.jrc.ec.europa.eu/documents/Data_Specifications /D2.5_v3.2.pdf>*, date retrieved: 07.09.2012.

**Url-6** *<http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/Data Specifications /D2.6_v3.0.pdf>*, date retrieved: 07.09.2012.

**Url-7** *<http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/Data Specifications /D2.7_v3.0.pdf>*, date retrieved: 07.09.2012.

**Vaccari, L., Shvaiko, P. and Marchese, M.** (2009). A Geo-Service Semantic Integration in Spatial Data Infrastructures, *International Journal of Spatial Data Infrastructures Research,* Vol. 4, 24-51, Creative Commons, San Francisco, California, USA.

**Vandenbroucke, D. and Biliouris, D.** (2010). *Spatial Data Infrastructures in Croatia: State of Play 2011,* Retrieved from http://inspire.jrc.ec.europa.eu/reports/stateofplay2011/rcr11HRv21.pdf

**van Oosterom, P. J. M.** (2005). Geo-information Standards in Action, *NCG / GIN Farewell Seminar Henri J.G.L. Aalders,* Netherlands Geodetic Commission, Delft, Netherlands.

**Vila, L. D.** (2011). *Personalized touristic routes implementing Semantic Web on GIS system*, Retrieved from http://hdl.handle.net/10609/6152.

**Wang, Y., Dai, J., Sheng, J., Zhou, K. and Gong, J.** (2007). Geo-ontology design and its logic reasoning, *Geoinformatics 2007: Geospatial Information Science*, Proceeding of SPIE Vol. 6753, 675309.

**Web Feature Service Implementation Specification** <http://www.opengeospatial .org/standards/wms>, date retrieved: 07.09.2012.

**Web Map Service Implementation Specification** <http://www.opengeospatial.org/ standards/wms>, date retrieved: 07.09.2012.

**Weerasinghe, H.** (2009). Development of an Interface for Ontology-based transformation between Features of different Types, *Master of Science Thesis*, Institute for Geoinformatics, University of Muenster, Münster, Germany.

**Wicaksana, I. W. S.** (2006). A Peer-to-Peer (P2P) Based Semantic Agreement Approach For Spatial Information Interoperability, *PhD. Thesis*, Gunardama University, Indonesia.

**Wiemann, S., Bernard, L.** (2010). Conflation Services within Spatial Data Infrastructures, *The 13th AGILE International Conference on Geographic Information Science*, Guimarães, Portugal, May 10-14.

**Wytzisk, A. and Sliwinski, A.** (2004). Quo Vadis SDI?, *7th AGILE conference on Geographic Information Science*, Heraklion, Greece, April 29-May 1.

**Yilma, A. D.** (2006). Ontology Based Portals for Accessing Geospatial Data and Web Services, *Master of Science Thesis*, International Institute For Geo-Information Science And Earth Observation, Enschede, Netherlands.

**Zhang, C., Peng, Z. R., Zhao, T. and Li, W.** (2008). Transformation of Transportation Data Models from Unified Modeling Language to Web Ontology Language, *Transportation Research Record: Journal of the Transportation Research Board*, No. 2064, pp. 81-89. Transportation Research Board Publisher, Washington, USA.

**Zhang, C., Zhao, T., Li, W. and Osleeb, J. P.** (2010). Towards logic-based geospatial feature discovery and integration using web feature service and geospatial semantic web, *International Journal of Geographical Information Science*, Vol. 24, No. 6, pp. 903-923. Taylor & Francis Publisher, United Kingdom.

**Zhao, P., Yu, G. and Di, L.** (2007). Geospatial Web Services: Advances in Information Interoperability, *Emerging Spatial Information Systems and Applications*, pp. 1-35. Idea Group Publisher, Hershey, Pennsylvania, USA.

**Zurita, V. M.** (2004). Semantic-based Approach to Spatial Data Sources Integration, *PhD Thesis*, Software Departament de Llenguatges i Sistemes Informatics, Universitat Politecnica de Catalunya, Spain.

## APPENDICES

**APPENDIX A:** Transportation UML Models
**APPENDIX B:** Feature Dataset, Feature Classes and Domain Models in Created Geodatabase
**APPENDIX C:** Public Transportation Ontology Domain OWL File
**APPENDIX D:** Transportation nRQL Query Algorithm
**APPENDIX E:** Developed Application Java Codes
**APPENDIX F:** Developed Application User Interface OpenLayer Scripts

# APPENDIX A Transport Network UML Models

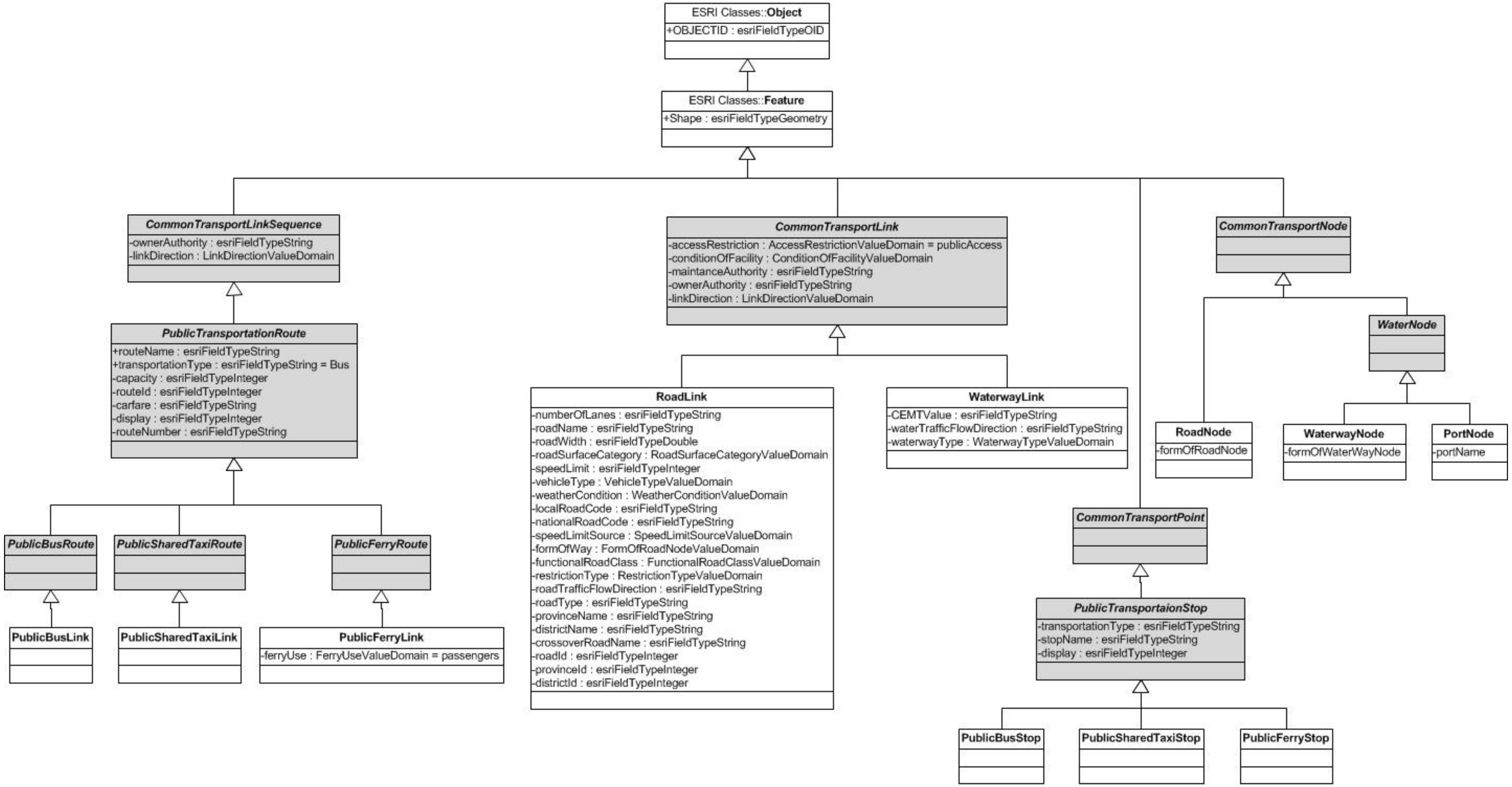



**Figure A.1 :** Transport network feature classes UML model

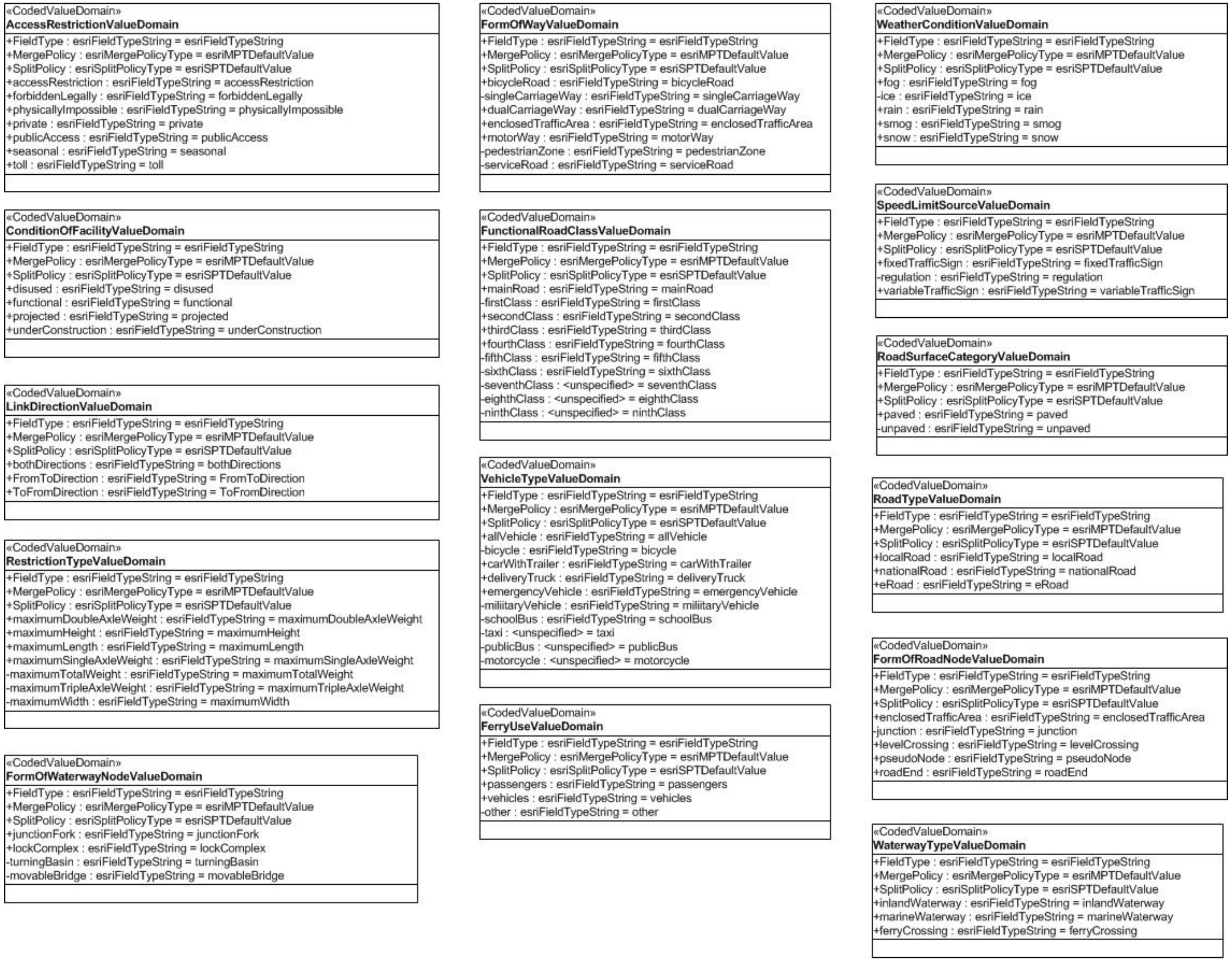



**Figure A.2 :** Transport network coded value domain models

## APPENDIX B Feature Dataset, Feature Classes and Domain Models in Created Geodatabase in ArcCatalog

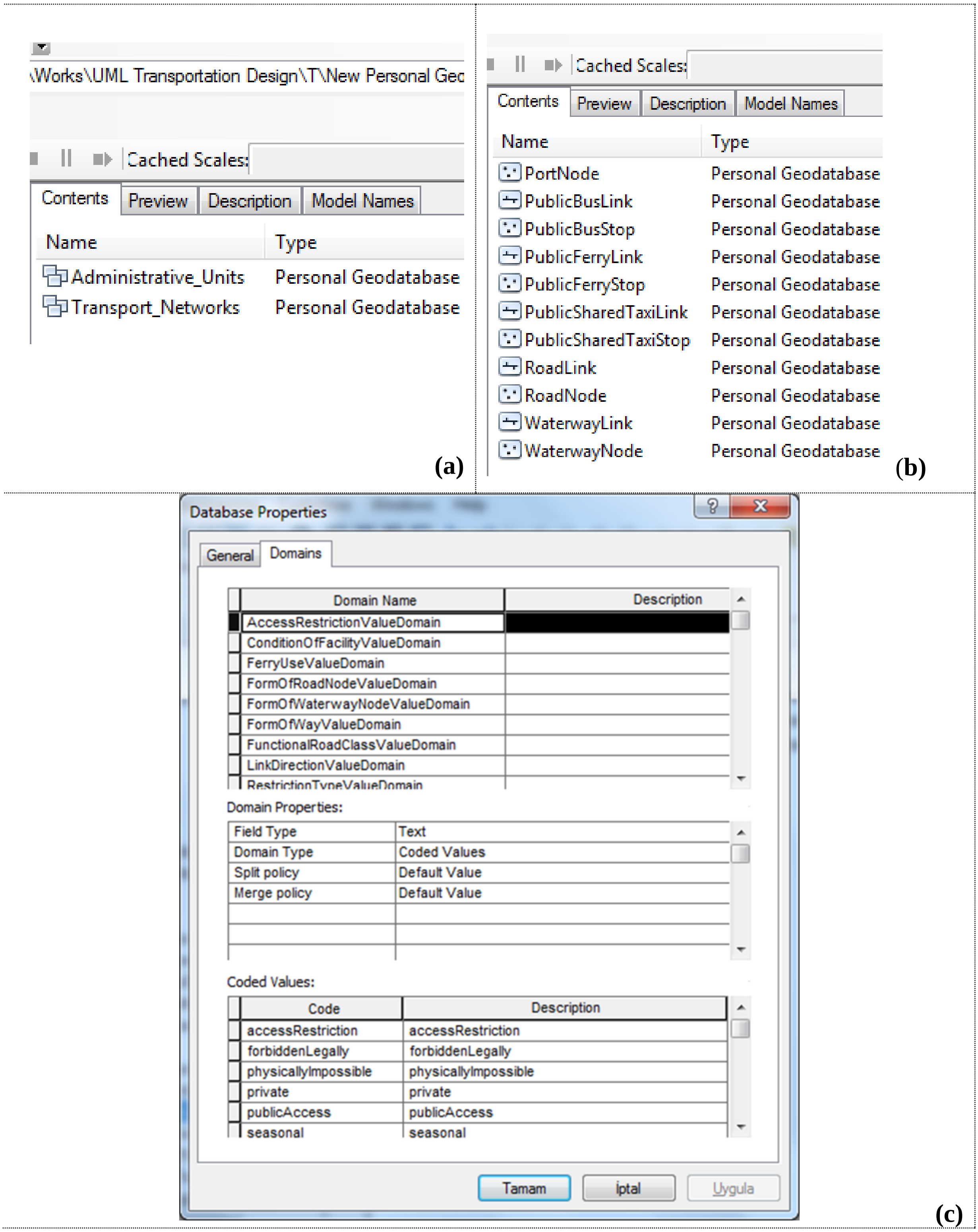



**Figure B.1 :** Generated Geodatabase: (a)Feature dataset. (b)Feature classes. (c)Domain Models

**APPENDIX C Public Transportation Ontology Domain OWL File**

```
    <owl:Ontology
rdf:about="http://www.semanticweb.org/ontologies/2012/0/Ontology132578471600
5.owl"/>                                                                 <!--
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#hasE
ndStop -->

    <owl:ObjectProperty rdf:about="&Ontology1325784716005;hasEndStop">

        <rdfs:subPropertyOf rdf:resource="&Ontology1325784716005;hasStop"/>

    </owl:ObjectProperty>                                                <!--
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#hasLi
nk -->

    <owl:ObjectProperty rdf:about="&Ontology1325784716005;hasLink">

        <rdfs:range
rdf:resource="&Ontology1325784716005;PublicTransportationLink"/>

        <rdfs:domain
rdf:resource="&Ontology1325784716005;PublicTransportationRoute"/>

        <owl:inverseOf rdf:resource="&Ontology1325784716005;isLinkOf"/>

    </owl:ObjectProperty>                                                <!--
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#hasSt
artStop -->

    <owl:ObjectProperty rdf:about="&Ontology1325784716005;hasStartStop">

        <rdfs:subPropertyOf rdf:resource="&Ontology1325784716005;hasStop"/>

        <owl:inverseOf rdf:resource="&Ontology1325784716005;isStartStopOf"/>

    </owl:ObjectProperty>                                                <!--
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#hasSt
op -->

    <owl:ObjectProperty rdf:about="&Ontology1325784716005;hasStop">
```

```
        <rdfs:domain
rdf:resource="&Ontology1325784716005;PublicTransportationLink"/>
<rdfs:range rdf:resource="&Ontology1325784716005;PublicTransportationStop"/>
    </owl:ObjectProperty>        <!--
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#isEnd
StopOf -->   <owl:ObjectProperty
rdf:about="&Ontology1325784716005;isEndStopOf">
        <owl:inverseOf rdf:resource="&Ontology1325784716005;hasEndStop"/>
        <rdfs:subPropertyOf rdf:resource="&Ontology1325784716005;isStopOf"/>
    </owl:ObjectProperty>        <!--
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#isLin
kOf -->  <owl:ObjectProperty rdf:about="&Ontology1325784716005;isLinkOf">
        <rdfs:domain
rdf:resource="&Ontology1325784716005;PublicTransportationLink"/>
<rdfs:range rdf:resource="&Ontology1325784716005;PublicTransportationRoute"/>
    </owl:ObjectProperty>       <!--
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#isStar
tStopOf -->  <owl:ObjectProperty
rdf:about="&Ontology1325784716005;isStartStopOf">
        <rdfs:subPropertyOf rdf:resource="&Ontology1325784716005;isStopOf"/>
    </owl:ObjectProperty>        <!--
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#isSto
pOf -->     <owl:ObjectProperty rdf:about="&Ontology1325784716005;isStopOf">
        <rdfs:range
rdf:resource="&Ontology1325784716005;PublicTransportationLink"/>
        <rdfs:domain
rdf:resource="&Ontology1325784716005;PublicTransportationStop"/>
        <owl:inverseOf rdf:resource="&Ontology1325784716005;hasStop"/>
    </owl:ObjectProperty>
```

```
<! --
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#Publi
cBusLink -->  <owl:Class rdf:about="&Ontology1325784716005;PublicBusLink">
        <rdfs:subClassOf
rdf:resource="&Ontology1325784716005;PublicTransportationLink"/>
        <rdfs:subClassOf>  <owl:Restriction>  <owl:onProperty
rdf:resource="&Ontology1325784716005;hasStartStop"/> <owl:onClass
rdf:resource="&Ontology1325784716005;PublicBusStop"/>
                <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
            </owl:Restriction>  </rdfs:subClassOf>  <rdfs:subClassOf>
<owl:Restriction><owl:onProperty
rdf:resource="&Ontology1325784716005;hasEndStop"/>  <owl:onClass
rdf:resource="&Ontology1325784716005;PublicBusStop"/>
<owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
            </owl:Restriction></rdfs:subClassOf> <rdfs:subClassOf>
            <owl:Restriction>  <owl:onProperty
rdf:resource="&Ontology1325784716005;isLinkOf"/>
<owl:someValuesFrom df:resource="&Ontology1325784716005;PublicBusRoute"/>
            </owl:Restriction>  </rdfs:subClassOf> </owl:Class>   <!--
http://www.semanticweb.org/ontologies/2012/0/Ontology1325784716005.owl#Publi
cBusRoute -->   <owl:Class
rdf:about="&Ontology1325784716005;PublicBusRoute">
        <rdfs:subClassOf
rdf:resource="&Ontology1325784716005;PublicTransportationRoute"/>
        <rdfs:subClassOf>   <owl:Restriction>   <owl:onProperty
rdf:resource="&Ontology1325784716005;hasLink"/>
<owl:someValuesFrom rdf:resource="&Ontology1325784716005;PublicBusLink"/>
            </owl:Restriction>  </rdfs:subClassOf>  </owl:Class>…
```

## APPENDIX D Public Transportation nRQL Query Algorithm

----> (add-role-assertions-for-datatype-properties)

---> To find route and related links:

(retrieve (?linkRoute ?linkStart ?linkEnd)

(and (#!Ontology1325784716005:StartStop ?linkStart #!Ontology1325784716005:isStartStopOf)

(#!Ontology1325784716005:EndStop ?linkEnd #!Ontology1325784716005:isEndStopOf)

(?linkStart #!Ontology1325784716005:PublicLink)

(?linkEnd #!Ontology1325784716005:PublicLink)

(?linkStart ?linkRoute #!Ontology1325784716005:isLinkOf)

(?linkEnd ?linkRoute #!Ontology1325784716005:isLinkOf)

(?linkStart ?linkEnd (constraint (#!Ontology1325784716005:hasLinkOrderId) (#!Ontology1325784716005:hasLinkOrderId) <= )) ))

----> To list the GeoIDs of Links based on previous query result

(retrieve (?y (datatype-fillers (#!Ontology1325784716005:hasLinkGeoId ?y))) (and

(?y #!Ontology1325784716005:iinkStart #!Ontology1325784716005:isLinkOf) (?y (and (>= #!Ontology1325784716005:hasLinkOrderId ) (<= #!Ontology1325784716005:hasLinkOrderId )))))

---->To find Start Stop and End Stop GeoIDs

(retrieve (?y (datatype-fillers (#!Ontology1325784716005:hasStartStopGeoId ?y))) (and (?y #!Ontology1325784716005:PublicLink) (= ?y #!Ontology1325784716005:LinkStart)))

(retrieve (?y (datatype-fillers (#!Ontology1325784716005:hasEndStopGeoId ?y))) (and (?y #!Ontology1325784716005:PublicLink) (= ?y #!Ontology1325784716005:linkEnd)))

----> To list datatype properties (hasRouteName, hasCarfare, hasTraveltime)

(retrieve ((datatype-fillers (#!Ontology1325784716005:hasRouteName ?y)) (datatype-fillers (#!Ontology1325784716005:hasRouteNumber ?y)) (datatype-fillers (#!Ontology1325784716005:hasCarfare ?y)) (datatype-fillers (#!Ontology1325784716005:hasCapacity ?y)) (datatype-fillers (#!Ontology1325784716005:hasTravelTime ?y)) ) (and (?y #!Ontology1325784716005:PublicRoute) ( = ?y #!Ontology1325784716005:PublicLink)))

----> if there is direct-route, finding the route with one transfer option

(retrieve (?linkRoute1 ?linkStart1 ?linkEnd1 ?linkRoute2 ?linkStart2 ?linkEnd2)

(and (#!Ontology1325784716005:StartStop ?linkStart1 #!Ontology1325784716005:isStartStopOf)

(?x ?linkEnd1 #!Ontology1325784716005:isEndStopOf)

(?linkStart1 #!Ontology1325784716005:PublicLink)

(?linkEnd1 #!Ontology1325784716005:PublicLink)

(?linkStart1 ?linkRoute1 #!Ontology1325784716005:isLinkOf)

(?linkEnd1 ?linkRoute1 #!Ontology1325784716005:isLinkOf)

(?linkStart1 ?linkEnd1 (constraint (#!Ontology1325784716005:hasLinkOrderId) (#!Ontology1325784716005:hasLinkOrderId) <= ))

(?x ?linkStart2 #!Ontology1325784716005:isStartStopOf)

(#!Ontology1325784716005:EndStop ?linkEnd2 #!Ontology1325784716005:isEndStopOf)

(?linkStart2 #!Ontology1325784716005:PublicLink)

(?linkEnd2 #!Ontology1325784716005:PublicLink)

(?linkStart2 ?linkRoute2 #!Ontology1325784716005:isLinkOf)

(?linkEnd2 ?linkRoute2 #!Ontology1325784716005:isLinkOf)

(?linkStart2 ?linkEnd2 (constraint (#!Ontology1325784716005:hasLinkOrderId) (#!Ontology1325784716005:hasLinkOrderId) <= )) ))…

**APPENDIX E Developed Application Java Codes**

```
package service;
import java.io.UnsupportedEncodingException;
import java.sql.*;
import java.util.Locale;
public class TransportationService {
String ip = "localhost";
int port = 8088;
RacerClient racer = new RacerClient(ip,port);
private Object anUTFString;
public String busRoute(String name, String stop2) throws
UnsupportedEncodingException {
String sqlBus = null;
Statement sql = null;
Connection db = null;
try {
//Open RacerPro Connection
racer.openConnection();
racer.addRoleAssertionsForDatatypeProperties();
try {
Class.forName("org.postgresql.Driver");
db = DriverManager.getConnection("jdbc:postgresql://localhost:5432/transportdb",
"postgres", "Admin123"); //connect to the db
sql = db.createStatement(); //create a statement that we can use later }
catch (ClassNotFoundException ex) {
System.out.println("connection error!");      }
```

```
String sameIndBusQuery1 = racer.racerAnswerQuery("nil",
"(#!Ontology1325784716005:" + busStop1 + "
#!Ontology1325784716005:PublicBusStop)");

String sameIndBusQuery2 = racer.racerAnswerQuery("nil",
"(#!Ontology1325784716005:" + busStop2 + "
#!Ontology1325784716005:PublicBusStop)");

if (sameIndBusQuery1.equals("t") && sameIndBusQuery2.equals("t")) {

query1 = racer.racerAnswerQuery("(?linkRoute ?linkStart ?linkEnd)", "(and
(#!Ontology1325784716005:" + busStop1 + " ?linkStart
#!Ontology1325784716005:isStartStopOf) (#!Ontology1325784716005:" +
busStop2 + " ?linkEnd #!Ontology1325784716005:isEndStopOf) (?linkStart
#!Ontology1325784716005:PublicBusLink) (?linkEnd
#!Ontology1325784716005:PublicBusLink) (?linkStart ?linkRoute
#!Ontology1325784716005:isLinkOf) (?linkEnd ?linkRoute
#!Ontology1325784716005:isLinkOf) (?linkStart ?linkEnd (constraint
(#!Ontology1325784716005:hasLinkOrderId)
(#!Ontology1325784716005:hasLinkOrderId) <= )))");

if (!query1.equals("nil")){

busquery = new String[routeCounter];

int k = 0;

for (int i=0; i < routeCounter; i++){

 (#!Ontology1325784716005:hasLinkGeoId ?y)))", "(and (?y
#!Ontology1325784716005:" + query1Members[i+k+0] + "
#!Ontology1325784716005:isLinkOf) (?y (and (>=
#!Ontology1325784716005:hasLinkOrderId " +
query1Members[i+k+1].substring(13) + ") (<=
#!Ontology1325784716005:hasLinkOrderId " +
query1Members[i+k+2].substring(13) + " )))))");

busquery[i]= racer.racerAnswerQuery("((datatype-fillers
(#!Ontology1325784716005:hasLinkGeoId ?y)))", "(and (?y
#!Ontology1325784716005:" + query1Members[0 + i + k] + "
```

```
#!Ontology1325784716005:isLinkOf) (?y (and (>=
#!Ontology1325784716005:hasLinkOrderId " + query1Members[1 + i +
k].substring(13) + ") (<= #!Ontology1325784716005:hasLinkOrderId " +
query1Members[2 + i + k].substring(13) + " )))))");

System.out.println("sorgudan sonra" +  busquery[i]);

k = k + 2;   }

//Lists StartStopGeoId and EndStopGeoId based on links result

String busStartStopGeoId = racer.racerAnswerQuery("((datatype-fillers
(#!Ontology1325784716005:hasStartStopGeoId ?y)))", "(and (?y
#!Ontology1325784716005:PublicBusLink) (= ?y #!Ontology1325784716005:" +
query1Members[1] + "))");

busStartStopGeoId = busStartStopGeoId.replace(":datatype-fillers ", ":datatype-
fillers");

busStartStopGeoId = busStartStopGeoId.replace(" ?y", "?y");

busStartStopGeoId = busStartStopGeoId.substring(119, (busStartStopGeoId.length()
-5));

String busEndStopGeoId = racer.racerAnswerQuery("((datatype-fillers
(#!Ontology1325784716005:hasEndStopGeoId ?y)))", "(and (?y
#!Ontology1325784716005:PublicBusLink) (= ?y #!Ontology1325784716005:" +
query1Members[2] + "))");

busEndStopGeoId = busEndStopGeoId.replace(":datatype-fillers ", ":datatype-
fillers");

busEndStopGeoId = busEndStopGeoId.replace(" ?y", "?y");

busEndStopGeoId = busEndStopGeoId.substring(117, (busEndStopGeoId.length() -
5));  }

String[] tempsplit = null;

int query2ArrayNum = 0;

query2Members = new String[routeCounter][];

for (int i=0; i < routeCounter; i++){
```

```
tempsplit = correctedquery2[i].split(" ");
query2ArrayNum = tempsplit.length;
query2Members [i] = new String[query2ArrayNum];
for (int j=0; j < query2ArrayNum; j++)
query2Members[i][j] = tempsplit[j];     }
//: DB Connection
        try {
           for (int i=0; i < routeCounter; i++){
               for (int j=1; j < query2Members[i].length; j+=2){
                   sqlBus = "UPDATE \"PublicBusLink\" SET display = 1 WHERE gid =
"+ query2Members[i][j] +" ";
             sql.executeUpdate(sqlBus);      }   }
           sqlBus = "UPDATE \"PublicBusStop\" SET display = 1 WHERE gid = "+
busStartStopGeoId +" ";
           sql.executeUpdate(sqlBus);
            sqlBus = "UPDATE \"PublicBusStop\" SET display = 1 WHERE gid = "+
busEndStopGeoId +" ";
           sql.executeUpdate(sqlBus);
           } catch (SQLException ex) {
           ex.printStackTrace();
           }     }
…
```

## APPENDIX F Developed Application User Interface OpenLayer Scripts

```
<script src="http://openlayers.org/api/OpenLayers.js"></script>
<script defer="defer" type="text/javascript">
var zoom = 5; var map; var highlightlayer;
OpenLayers.IMAGE_RELOAD_ATTEMPTS = 5;
OpenLayers.DOTS_PER_INCH = 25.4 / 0.28;
 function init(){
format = 'image/png';
var bounds = new OpenLayers.Bounds(390002.312, 4529446.454, 430824.028,
4558720.049  );
var options = {controls:[], maxExtent: bounds, maxResolution: 95.90514062500006,
projection: "EPSG:32636", units: 'm' };
map = new OpenLayers.Map($('map'), options);
province = new OpenLayers.Layer.WMS(
"Province", "http://localhost:8080/geoserver/wms",    {
layers: 'cite:Province', srs: 'EPSG:32636', format: format,
tiled: true, tilesOrigin : map.maxExtent.left + ',' + map.maxExtent.bottom,
isBaseLayer: true, displayInLayerSwitcher: 'FALSE',  visibility: true   },
{ buffer: 0,  displayOutsideMaxExtent: true   }  );
buslink = new OpenLayers.Layer.WMS(
"Bus Route", "http://localhost:8080/geoserver/wms",
{transparent: 'TRUE',  layers: 'cite:PublicBusLink',  srs: 'EPSG:32636',
format: format, isBaseLayer: false, visibility: true   },
displayInLayerSwitcher: 'TRUE'}   );
busstop = new OpenLayers.Layer.WMS(
"Bus Stop", "http://localhost:8080/geoserver/wms",
```

```
{ transparent: 'TRUE', layers: 'cite:PublicBusStop', srs: 'EPSG:32636',
format: format,  isBaseLayer: false, visibility: true        },
{displayInLayerSwitcher: 'TRUE'}   );
ferrystop = new OpenLayers.Layer.WMS(
"Ferry Stop", "http://localhost:8080/geoserver/wms",
{transparent: 'TRUE', layers: 'cite:PublicFerryStop',  srs: 'EPSG:32636',
format: format, isBaseLayer: false, visibility: true      },
{displayInLayerSwitcher: 'TRUE'}     );
ferrylink = new OpenLayers.Layer.WMS(
"Ferry Route", "http://localhost:8080/geoserver/wms",
{transparent: 'TRUE', layers: 'cite:PublicFerryLink', srs: 'EPSG:32636',
format: format, isBaseLayer: false, visibility: true        },
{displayInLayerSwitcher: 'TRUE'}   );
sharedtaxilink = new OpenLayers.Layer.WMS(
"Shared Taxi Route", "http://localhost:8080/geoserver/wms",
{transparent: 'TRUE', layers: 'cite:PublicSharedTaxiLink', srs: 'EPSG:32636',
format: format, isBaseLayer: false, visibility: true     },
{displayInLayerSwitcher: 'TRUE'}          );
sharedtaxistop = new OpenLayers.Layer.WMS(
"Shared Taxi Stop", "http://localhost:8080/geoserver/wms",
{transparent: 'TRUE', layers: 'cite:PublicSharedTaxiStop', srs: 'EPSG:32636',
format: format,  isBaseLayer: false, visibility: true     },
{displayInLayerSwitcher: 'TRUE'}   );
map.addLayers([sharedtaxilink, buslink, province, ferrylink, busstop, sharedtaxistop,
ferrystop]); map.zoomToExtent(bounds); map.updateSize();
// build up all controls
```

```
map.addControl(new OpenLayers.Control.PanZoomBar({position: new
OpenLayers.Pixel(2, 15)}));
map.addControl(new OpenLayers.Control.Navigation());
map.addControl(new OpenLayers.Control.LayerSwitcher());
map.addControl(new OpenLayers.Control.Scale($('scale')));
map.addControl(new OpenLayers.Control.MousePosition({element: $('location')}));
map.addControl(new OpenLayers.Control.OverviewMap());     };
// sets the HTML provided into the nodelist element
function setHTML(response){
document.getElementById('nodelist').innerHTML = response.responseText;};
function errorHTML(response){alert("req erro:" + response.responseText);};
</script>   </head>
<body onLoad="init()">
<h1 id="title">Semantic Public Transportation Geoportal</h1> <div
id="info"></div>
<p id="shortdesc">Below map shows Bus, SharedTaxi and Ferry Transortation
alternatives based on your request. </p>
<div id="map"></div> <div id="wrapper"> <div
id="location"style="float:left">location</div>
 <div id="scale" style="float:left"></div>
<div id="scale" style="clear:both"></div </div>
<table>  <tbody>    <tr class="head"> <td>Transportation Route
Alternatives</td><td>Route Information</td>  </tr>
<tr>   <td>Bus Route</td>  <td>${busRouteMessage}</td>    </tr>…
```

**CURRICULUM VITAE**

| | |
|---|---|
| **Name Surname:** | Arif GÜNAY |
| **Place and Date of Birth:** | Antalya, 1982 |
| **Address:** | İzzetpaşa Sok. Doruk Apt. No: 43, Şişli/İSTANBUL |
| **E-Mail:** | arifgun@hotmail.com |
| **B.Sc.:** | Geodesy and Photogrammetry Engineering |
| **M.Sc.:** | Geomatics Engineering |