

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Telecommunications Service Domain Ontology: Semantic Interoperation Foundation of Intelligent Integrated Services

Xiuquan Qiao, Xiaofeng Li and Junliang Chen
*State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
China*

1. Introduction

Network is the bearer of services and services are the soul of network. The convergent network extends the original communications service type and gradually forms new convergent services which integrate the traditional telecommunication services and a large number of value-added services or contents on Internet (Kolberg et al., 2010). The integrated service is essentially to handle the data and services across heterogeneous networks and service platforms. Facing the heterogeneity and diversity of service resources, integrated services need to run in a multi-terminal, multi-access network and multi-platform heterogeneous environment. These tremendous changes of service environment present a significant interoperability challenge for traditional service provisioning theory. Nowadays, the provision of context-awareness, adaptive personalized services is the development goal of future ubiquitous network (Park et al., 2009). It can enable seamless information exchange between humans, with humans and with entities (e.g., mobile devices), as well as entities and entities at any time, any place and in any way. To meet the development needs of adaptive personalized convergent services, dynamic service discovery and composition technologies are explored widely in the telecommunication service field (Bashah et al., 2010; Niazi & Mahmoud, 2009).

Today, semantic web service (McIlraith, 2001), as an establishing research paradigm, is defined as an augmentation of web service with semantic annotation, to facilitate the higher automation of service discovery, composition, invocation and monitoring in an open environment. Integration of the semantic web technology and telecommunications systems is explored widely in the telecommunication service field (Do & Jorstad, 2005; Vitvar & Viskova, 2005; Qiao et al., 2008a; Gutheim, 2011; Khan et al., 2011; Zander & Schandl, 2011). It is well known that ontology is the semantic interoperability and knowledge sharing foundation for semantic web services matching and context reasoning. Therefore, how to construct the telecommunications service domain ontology is an important factor of successfully applying semantic web services into telecommunication service systems (Veijalainen, 2007, 2008). However, telecommunication service field consists of a large number of concepts/terminologies and relations. How to abstract the sharing domain

concepts and reasonably organize them is a big challenge. Some related work has been done mainly in applying ontology technology to the mobile service domain. Based on the need for a standardized ontology that describes semantic models of the domains relevant for scalable NGN (Next Generation Network) service delivery platforms, the (Villalonga et al., 2009; Su et al., 2009) provide an overview of Mobile Ontology which comprises a core ontology and several subontologies, and its application examples in the service delivery platform. This work, as a part of IST SPICE project (IST SPICE project, 2008), is a meaningful attempt to establish a standardized ontology for mobile service delivery in NGN. In addition, IST SIMS project explored the semantic interfaces as a new means to specify and design service components and to guarantee compatibility in static and dynamic component compositions. And they also defined a domain-specific ontology, and its main purpose of the ontology is to establish a common description of the SIMS-related concepts and their semantics (Rój, 2008). The (Zhu et al., 2010) introduces a mobile ontology construction and retrieval system architecture. However, there lacks a general domain ontology modelling methodology for telecommunications service and the corresponding engineering approach to support the development work for domain ontology. The (Li et al., 2010) briefly introduced the constructing method of telecommunications service domain ontology (TSDO) proposed by our research team. However, the approach is not perfect at that time and still needs to be further improved. In fact, telecommunication service domain ontology, as the important semantic interoperability foundation of telecom network, still has no significant progress up to now. This has become the biggest obstacle to hamper the applications of semantic web technology in telecom field.

In this chapter, we clearly presented a practical domain ontology modelling approach for telecommunications service field. Under the guidance of this approach, our research team has created an open telecommunications service domain ontology knowledge repository which consists of around 430 telecommunications services-related ontology concepts/terminologies and 245 properties. Based on this domain ontology, we described the telecom network capability services in the semantic level to validate its feasibility. The semantic annotation facilitated the accurate service description, discovery of telecommunication network services and addressed the semantic interoperability problem. The proposed model-driven domain ontology modelling approach separates domain conceptual model from the concrete ontology modelling languages, it enhances the reusability of domain conceptual model and greatly reduces the technical difficulty of domain ontology modelling.

The remainder of this chapter is structured as follows. In Section 2 we presented a general domain ontology modelling methodology for telecommunications service field, and also proposed a specific model-driven domain ontology modelling approach to support the above presented methodology. Section 3 introduced the experimental environment and the demo service to validate the feasibility of domain ontology. Finally, conclusions are drawn.

2. Domain ontology modelling methodology for telecommunications service

Here, technical modelling details for the proposed approach are described, namely telecommunications service domain ontology modeling methodology and a corresponding model-driven implementation mechanism.

2.1 Domain ontology modeling methodology

Based on our practical experiences in recent years, a concrete domain ontology modeling methodology is summarized as shown in Figure 1. The modelling process is illustrated in detail as follows.

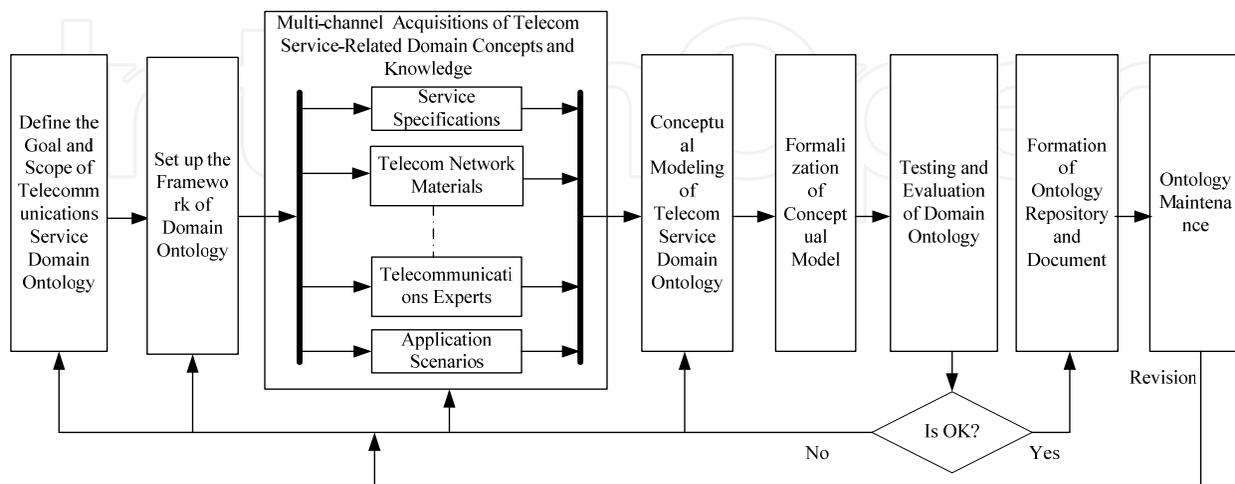


Fig. 1. Domain ontology modelling methodology.

2.1.1 Define the scope of telecommunications service domain ontology

The first step is mainly to define the scope and border of domain ontology. Telecommunications service domain ontology mainly addresses the semantic interoperability of telecommunications service. This domain ontology mainly provides the shared domain vocabularies and knowledge to support the semantic web applications in the telecommunication service field, such as semantic telecom service description, service discovery, and service context modelling. Therefore, TSDO should involve the service-related domain concepts and knowledge. For example, telecom services often involve network type, network carrier, billing policy, user terminal, service quality, service customer, service category, .etc. In fact, telecommunication service field consists of a large number of concepts/terminologies and relations. Some concepts have the higher sharing degree. However, some concepts are only related to concrete application field, such as service context ontology, service description ontology. Therefore, how to abstract the sharing domain concepts and reasonably organize them is a big challenge. The reusability and extensibility are two important ontology modeling factors considered. So an efficient ontology hierarchy modelling approach is needed.

In practice, we adopted a layered ontology modeling method to organize the domain concepts to improve the reusability and extensibility (see Figure 2). Common ontology, like time and space ontologies, can be shared in the different domains, like telecom, medical domain or any other domains. The concrete domain ontology can be shared by the different domain-related application ontologies. For example, TSDO may be used to create the service context ontology, network management ontology, etc. This method well distinguishes the border of TSDO, common ontology and telecom service-related application ontology.

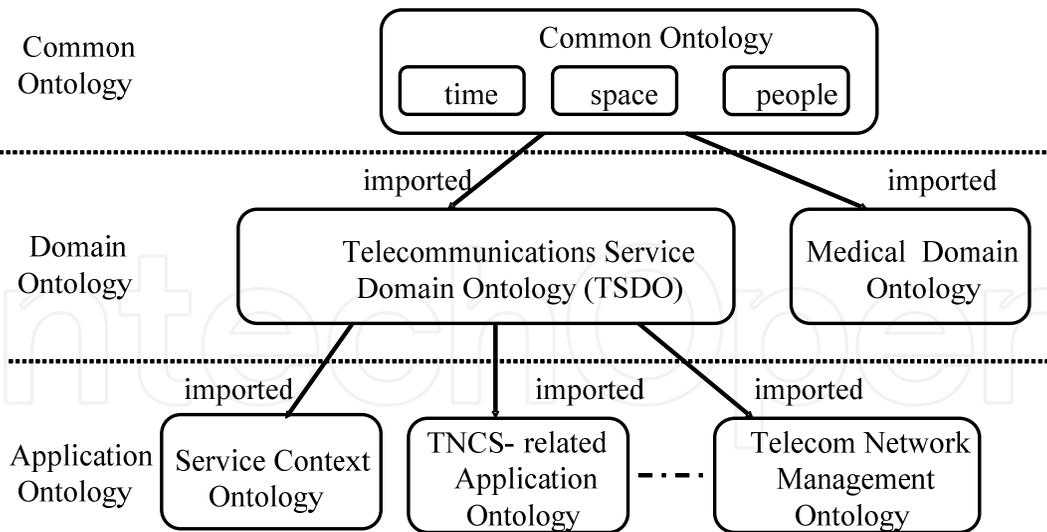


Fig. 2. Layered ontology modelling method.

2.1.2 Set the framework of telecommunications service domain ontology

When the goal and scope of TSDO are clear, the specific organization framework of TSDO should be set up. As TSDO involves a large number of telecom service domain concepts and relationships, how to reasonably classify and organize these terminologies is an important issue. Specifically, we adopted a modular modelling approach to construct TSDO. The principle of modular modelling is the “strong cohesion and loose coupling” way. The correlations among different concepts are the main reference of module division. The goal of modular modelling is to ensure that the correlation of concepts in the same module is stronger. Based on this modular design principle, TSDO is divided into several sub-ontologies as shown in Fig. 3.

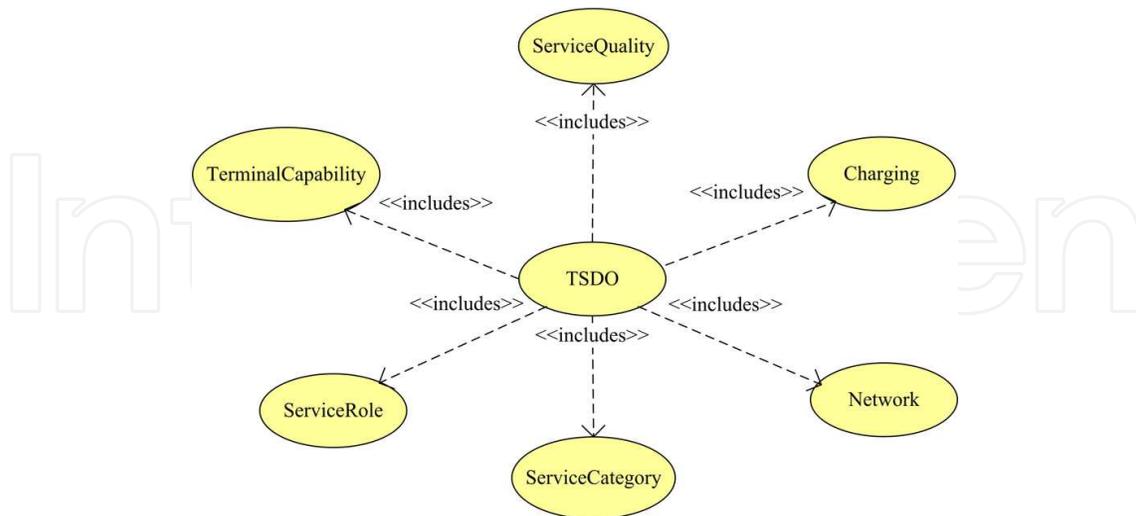


Fig. 3. The framework of telecommunications service domain ontology.

Specifically, TSDO mainly comprises six sub-ontologies, including Terminal Capability Ontology, Network Ontology, Service Role Ontology, Charging Ontology, Service Quality Ontology, and Service Category Ontology.

1. **Terminal Capability Ontology:** defines main concepts about terminal software, terminal hardware, terminal browser and network characteristics supported by terminal.
2. **Network Ontology:** specifies the network concepts, network category, network features, as well as the relationships of various networks, such as mobile network, internet, and fixed network, GSM, CDMA, UMTS, WCDMA, and WLAN.
3. **Service Role Ontology:** describes the stakeholders' concepts of the service supply chain, for example, service provider, content provider, network operator, service user.
4. **Service Category Ontology:** describes a telecommunications service classification. This ontology defines the relationship between various telecommunications services, like basic service, value-added service, voice service, data service, conference service, presence service, download service, browsing service, messaging service.
5. **Charging Ontology:** defines the charging-related concepts and rules about telecommunications services, including payment methods (such as prepaid and post-paid), charging types (such as time-based, volume-based, event-based, and content-based), billing rates, as well as account balances.
6. **Service Quality Ontology:** A telecommunication network must provide the services which have the end-to-end QoS guarantee. Depending on the technical characteristics, the QoS provided by different networks is varying. Service Quality Ontology mainly defines the QoS-related concepts about telecommunications service, including access network QoS, core network QoS and user's QoE, such as call delay, message size, call through rate, positioning accuracy, network bandwidth.

2.1.3 Multi-channel acquisitions of telecom service-related domain concepts and knowledge

After the framework of TSDO is set up, it needs to collect domain concepts and knowledge (including terminologies and their relationships) from multi-channel ways for each sub-ontology of TSDO. In general, the sources of knowledge acquisition include the released telecom service specifications, senior experts in the telecom field or some typical application

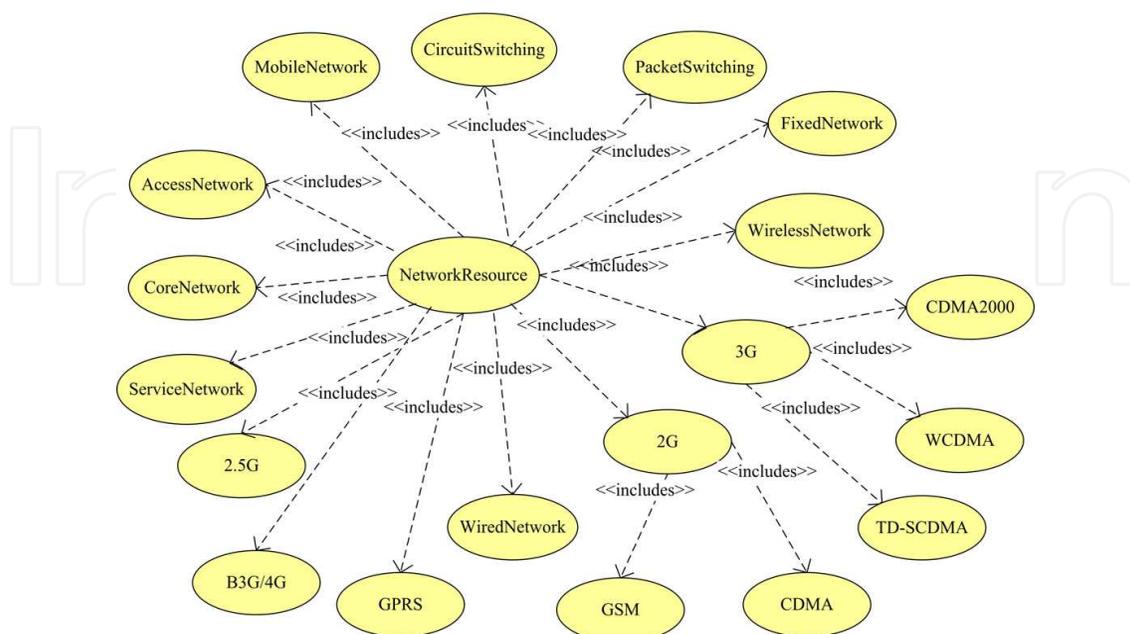


Fig. 4. Some collected domain concepts about telecom network.

scenarios. In this step, modellers need to list the collected concepts, relations and explanations as far as possible. It's unnecessary to care about the meaning overlap between the concepts and to consider how to express these concepts and their relation in class, property or instance ways. For example, Figure 4 briefly shows the concepts collection about network ontology.

2.1.4 Conceptual modelling of telecommunications service domain ontology

After the acquisition of a large number of telecom service related concepts, we need to make the concept classification, concept aggregation, and remove the duplicated concepts according to certain domain knowledge and logic. The goal of this step is to construct a conceptual model of TSDO. This concept model describes the involved domain concepts and their relations of each sub-ontology in detail. Note that, the relationships between the concepts not only involve the concepts of the same sub-ontology, may also be related to the concepts of different sub-ontologies. The concrete building of conceptual model is divided into three steps: (1) **Defining classes and class hierarchy**. In the process of defining the classes, we need to discover the inheritance hierarchy between the concepts and then distinguish the super-classes and sub-classes. (2) **Defining the properties of classes**. After the class is defined, its properties should be considered. There are two kinds of properties. One is datatype property, which is used to describe the features of the concept itself, such as name, age. The other is object property, which is used to depict the relationship between the concepts, like friendship relation between two people. (3) **The definition of domain axiom and knowledge**. When we use ontology to describe the real word things, there are often some contradictions or errors occurrences resulted by human negligence. For example, the range value of one person age property is negative, or a person has two biological fathers. To prevent these common-sense errors, some domain axiom and knowledge should be established. The axiom is to restrict the relationships of the concepts to ensure the consistency of domain knowledge, such as the range value or cardinality of properties.

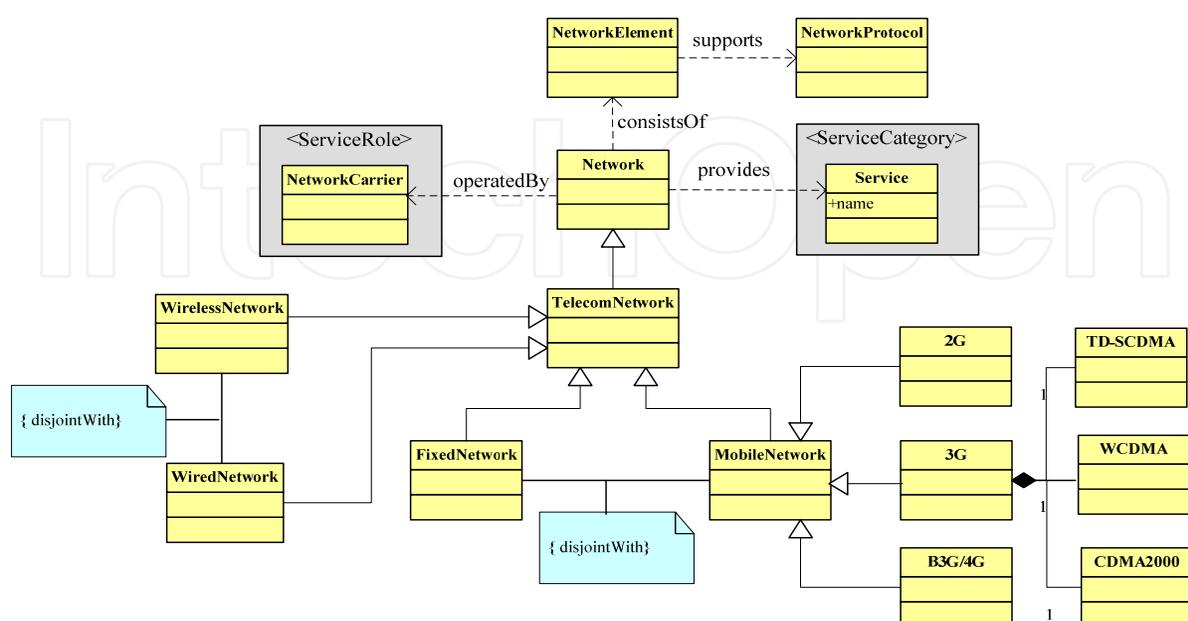


Fig. 5. Part conceptual model of network ontology.

Figure 5 shows the conceptual model of network ontology in part. Based on the terminologies collected in the above step, the class hierarchy and relationships are described. This conceptual model depicts the classification of network, the services provided by network and the operator of network. It can be seen that the ranges of object property “operatedBy” and “provides” are the concepts from ServiceRole and ServiceCategory sub-ontologies respectively. In addition, we define the domain axioms through the constraints way. For example, we define that “FixedNetwork” is disjoint with “MobileNetwork”, i.e. if N1 is an instance of concept “FixedNetwork”, then it will not be an instance of concept “MobileNetwork”.

2.1.5 Formalization of conceptual model of telecommunications service domain ontology

As the conceptual model is one high-level abstract model and independent of any concrete ontology modelling languages, we need to formalize this conceptual model through a specific ontology modelling language like OWL (Web Ontology Language) (W3C, 2004a). In general, we can use the common ontology modelling tools to formally describe the terminologies, relationships and axioms in the conceptual model. Figure 6 shows the formalization description of part concepts and relationships of Figure 5 by OWL language. The concept is formally defined by “owl:Class”, and the class hierarchy is organized by “owl:subClassOf”. The “owl:ObjectProperty” is used to describe the relationships between the concepts and the “owl:disjointWith” clearly depicts the restrictions on the two disjointed concepts.

```
<owl:Class rdf:ID="Network"/>
<owl:Class rdf:about="#TelecomNetwork">
  <rdfs:subClassOf rdf:resource="#Network"/>
</owl:Class>
<owl:Class rdf:about="#MobileNetwork">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#TelecomNetwork"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#FixedNetwork"/>
</owl:Class>
<owl:Class rdf:ID="2G">
  <rdfs:subClassOf rdf:resource="#MobileNetwork"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="operatedBy">
  <rdfs:range rdf:resource="#ServiceRole;#NetworkCarrier"/>
  <rdfs:domain rdf:resource="#Network"/>
</owl:ObjectProperty>
```

Fig. 6. Part of network ontology formalized by OWL.

2.1.6 Evaluation of telecommunications service domain ontology

Ontology evaluation is an important issue that must be addressed if TSDO are to be widely adopted in the semantic related telecommunications applications. Ontology can be

evaluated against many criteria: its coverage of a particular domain and the richness, complexity and granularity of that coverage; the specific use cases, scenarios, requirements, applications, and data sources it was developed to address; and formal properties such as the consistency and completeness of the ontology. We can test and validate whether the domain ontology satisfy the requirement or not. If yes, these ontologies will be added to the ontology repository; if no, we have to return back to previous steps to make some revisions until the requirement is satisfied.

In the specific use process, we often can find some existing shortcomings of domain ontology. The utilization of domain ontology to formally describe the concrete application scenario is a very effective evaluation approach. For example, when we defined the TSDO, we use network, service role and service category sub-ontologies to describe the network carrier resource (see Figure 7). We found that the operating scope of network carrier is an important characteristic. But the concept “NetworkOperator” of service role sub-ontology lacks this property. Actually, some carriers can provide services through out nation; however, some carriers can only provide services in a specific province or region. Therefore, the property “CoverageScope” should be added to the concept “NetworkOperator” of service role sub-ontology.

```

<ServiceRole:NetworkOperator rdf:ID="ChinaMobileCommunicationOperator"/>
  <ServiceRole:CoverageScope rdf:resource="&LocationSpace;#TroughOutNation"/>
</ServiceRole:NetworkOperator>

<TelecomNetwork:GSM rdf:ID="ChinaMobileNetwork">
  <TelecomNetwork:operatedBy rdf:resource="#ChinaMobileCommunicationOperator"/>
  <TelecomNetwork:provides rdf:resource="&ServiceCategory;#DataService"/>
  <TelecomNetwork:provides rdf:resource="&ServiceCategory;#VoiceService"/>
</TelecomNetwork:GSM>

```

Fig. 7. Ontology description of china mobile communication operator.

2.1.7 Maintenance of telecommunications service domain ontology

The construction of domain ontology is the basis of ontology applications. However, as the different domain experts or ontology modelers may have the different understandings of the same domain concepts or relationships, some created ontologies may need to be further revised or improved in the practical utilization process. In addition, the knowledge of real world is growing and updated continuously. This also results that regular maintenance is necessary after ontologies have been constructed. Ontology maintenance refers to a series of amendments, corrections, improvements and adaptive maintenance for ontology, which mainly consists of improving maintenance and adaptive maintenance. The improving maintenance is to revise or correct some existing errors of domain ontology. However, the adaptive maintenance refers to the extensions of existing domain ontology with the external real world changes, such as the knowledge increase or technology advances.

In addition, with the maturity of ontology technology, there are some ontologies developed by different research teams or communities to satisfy their different application needs. The main advantage of ontology is the knowledge sharing and reuse. How to realize the interoperation with these existent distributed ontologies is a big problem of ontology

maintenance. Therefore, sometimes, it needs to integrate several existent ontologies to address the reuse of different ontology knowledge. To implement the different ontology integration, the relationships among different ontologies should be analyzed. As the distributed feature and openness of WWW, knowledge ontologies maybe have the direct or indirect semantic relationships. For example, two ontologies maybe involve some same or similar concepts. The main relationships consist of two kinds: one is the repeat of terminologies definition. Some terminologies of this ontology might be equivalent to those defined in that ontology. It consists of the class equivalent and the property equivalent. For this equivalent relationship, we can use equivalent ontology mapping method to resolve as shown in Figure 8. The other is the subsumption of terminologies definition. It means that some terminologies of one ontology might subsume the semantic scope of those terminologies defined in other ontology. It also involves the class subsumption and property subsumption. For example, Figure 9 shows two independent ontologies: ontology 1 and ontology 2. In fact, the concept “Netowrk” of ontology 1 subsumes the concept “Internet” of ontology 2 in the semantic scope. Therefore, we can use the subsumption relationship to integrate these two ontologies into a new ontology.

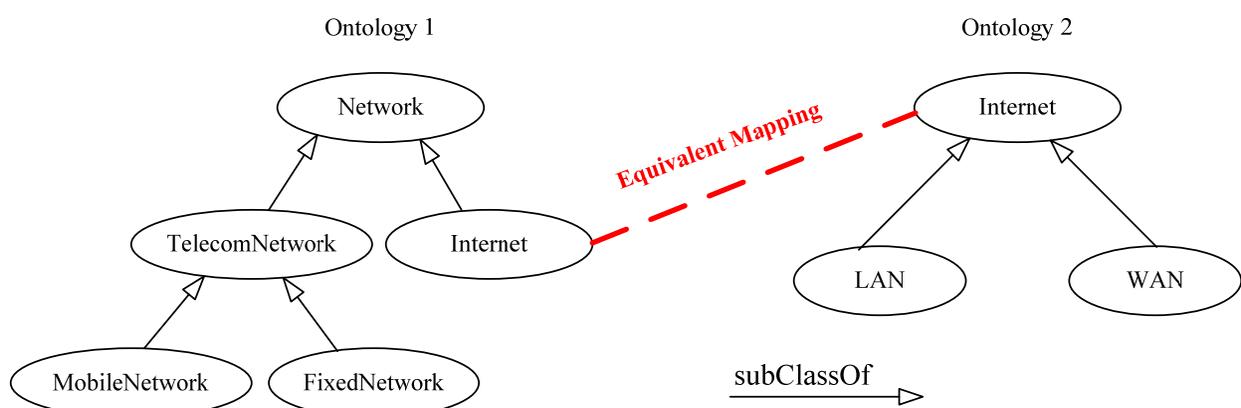


Fig. 8. Ontology integration based on the equivalent mapping.

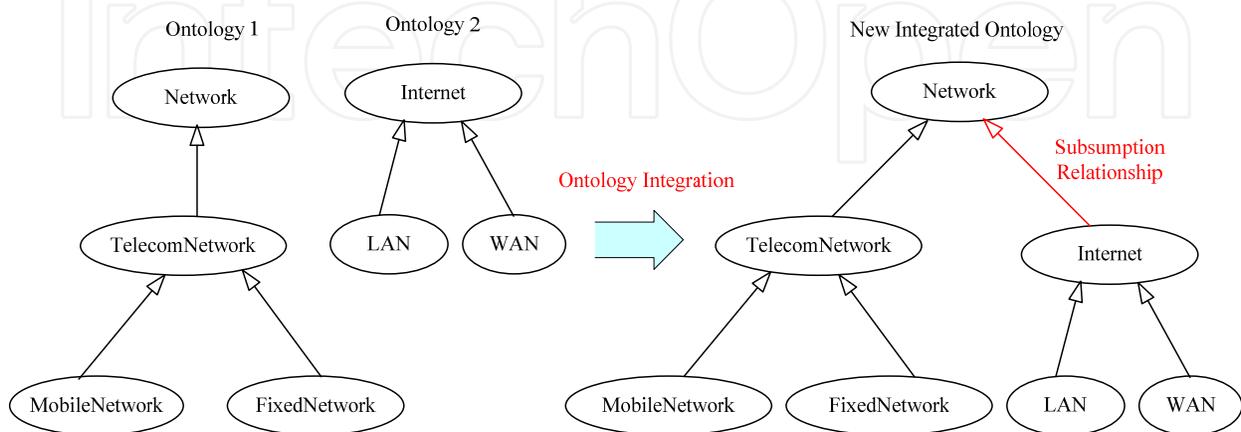


Fig. 9. Ontology integration based on the subsumption relationship.

2.2 A model-driven domain ontology modeling implementation approach

From the above descriptions in section 2.1, it can be seen that the construction of TSDO is a complex work, which involves not only several steps like terminology acquisition, concept modelling and formal description, but also different modellers like domain experts, formalization modeller. Currently, it lacks of a unified modelling tool to efficiently support this methodology. As the ontology modelling languages consists of a large number of logical symbols and formal description knowledge, it is not easy for general domain experts or software developers to understand and master. Although there are some visual modelling tools like Protege (Stanford, 2004) to support ontology modelling, the ontology modelling process still lacks the relation with mature software engineering method. For the general software developers, the current ontology modelling approach is not easy to master and it needs a strong professional background. Therefore, in the actual process of building domain ontology, domain experts often use UML (Unified Modelling Language) (OMG, 2005a) modelling tool or other office software to acquire domain terminologies or create concepts model, and then formalization modellers formalize the conceptual model by a specific ontology language through ontology modelling tool like Protege. As the existing UML modelling tool do not support the ontology modelling directly and the common ontology modelling tools also do not support the requirements and high-level conceptual modelling, the above proposed modelling process has to switch between different modelling tools. A key problem is that the high-level conceptual model cannot be automatically transformed into formal model encoded by a specific ontology language. This brings a lot of management and maintenance inconveniences of ontology modelling. The existing ontology modelling approach has limited the large-scale ontology development. Therefore, it needs a practical engineering approach and a unified modelling tool to support this modelling methodology completely.

Essentially, ontology engineering emphasizes the ontology modelling and knowledge reasoning; however, software engineering focuses on the complete system development methodology which mainly pays attention to requirement analysis, system design, implementation and dose not have the logical reasoning capability. So how to use mature software engineering theory and method to support the ontology development is very significant. Today, Model Driven Development (MDD) (Selic, 2003) is gaining significant momentum in both the software industry and the software engineering academic community. Model Driven Architecture (OMG, 2003), standardized by the Object Management Group (OMG), is a new strategy for designing software systems. Its main goal is to separate system function specification from specific implementation technique completely, enabling system's kernel function specification to be independent of the specific implementation platform technology. Therefore, MDA can retain the neutrality of programming languages, middleware platforms and vendors. In the face of heterogeneous and evolving technology, MDA is supposed to ensure: portability, increased application reuse and reduced development time. Thereby MDA minimizes the affection of technique changes.

Considering the development of domain ontology is a complex process and MDA is a new modeling approach which focuses on the model rather than the specific implementation technical details, we integrated MDA with ontology engineering together, and proposed a model driven domain ontology modeling approach to support the modelling methodology

described in section 2.1. By this approach, domain experts or general software developers, who are familiar with UML, can conveniently build the domain conceptual model by UML modelling tools and then this conceptual model can be automatically transformed into the corresponding ontology model encoded by a specific ontology language. As this approach separates domain conceptual model from the concrete ontology modelling languages like OWL, it enhances the reusability of domain conceptual model and reduces the technical difficulty of domain ontology modelling. The implementation details are described in the following sections.

2.2.1 Overview of model-driven TSDO modeling approach

MDA adopts the model-based development mode (Miller & Mukerji, 2003) as shown in Figure 10. Computation Independent Model (CIM) mainly describes the requirements of software system, which specify the system function and boundary. Platform Independent Model (PIM) is the high level abstraction of system function, without any information related to implementation techniques; Platform Specific Model (PSM) is the model which contains specific implementation platform technique information. The MDA-based development process is: firstly, establishing CIM based on the system requirements; secondly, according to the specifications of CIM, creating PIM with the platform independent modeling language, such as UML; thirdly, transforming the PIM to PSM according to some specific mapping rules; lastly, generating platform specific code automatically or semi automatically. In this process, modeller can further refine the created models in CIM, PIM or PSM stage.

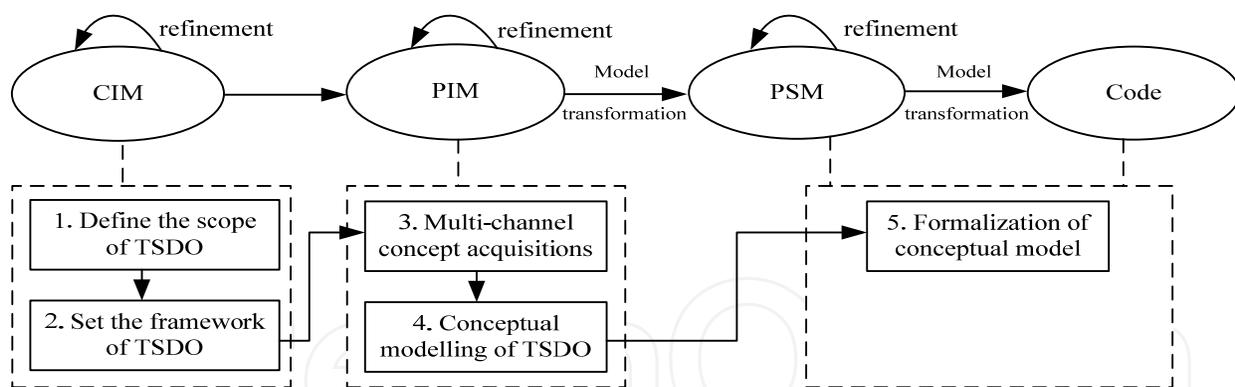


Fig. 10. Model-driven TSDO modelling approach.

According to the modelling idea of MDA, we presented a concrete model-driven TSDO modelling approach to provide a practical engineering implementation as shown in Figure 10. The definition of TSDO scope and the establishment of domain ontology framework belong to the CIM modelling stage. Modeller can employ use case diagram of UML to define the scope of TSDO and set up its framework. In this approach, PIM mainly focuses on the multi-channel domain concept acquisitions and the further conceptual integration and refinement, i.e. conceptual modelling. UML class diagram or use case diagram can be used to model the collected domain concepts and their relationships. After acquiring the domain terminologies, the following step is to integrate and refine these

concepts and their relationships to form a high-level domain ontology model, which is independent of a specific ontology description language. The PSM and code steps are used to realize the formalization of high-level domain ontology conceptual model by a specific ontology language. By the model to model transformation technology, the high-level conceptual model (i.e. PIM) can be transformed into an ontology language specific model (i.e. PSM). And then by using model to code transformation technology, the concrete ontology description file encoded by a specific ontology language like OWL (i.e. code) can be generated from the ontology language specific model (i.e. PSM). When we need to revise or maintain the created ontology, we can return back to the CIM or PIM to modify the related models and then generated the corresponding code again. In this mode-driven ontology development approach, all processes adopt the standard UML model or UML extension mechanism (i.e. UML Profile). The technical details are described in the following sections.

2.2.2 CIM step: The scope and framework modeling of TSDO

In order to well organize the development of TSDO, this approach uses the UML use case diagram to model the scope and framework of TSDO. As is shown in Figure 11(a), the ontology hierarchy is represented by package *InfrastructureOfOntology*, which consists of three types of package: common ontology, domain ontology and application ontology. Each package contains the related ontology concepts and their relationships. The *Common Ontology* package contains some general concepts particularly designed for high reusability, where other different domain ontologies and application ontologies either import or specialize its specified concepts or relationships. This is illustrated in Figure 11(a), where it is shown how domain ontologies and application ontologies each depends on the common ontology. The common ontology is generally defined by some standard organizations or research communities. In this chapter, we mainly focus on the building of telecommunications service domain ontology. In order to facilitate reuse, the *Telecommunications Service Domain Ontology* package is further subdivided into a number of packages: *ServiceCategory*, *Network*, *TerminalCapability*, *ServiceQuality*, *ServiceRole*, and *Charging*, as shown in Figure 11(b).

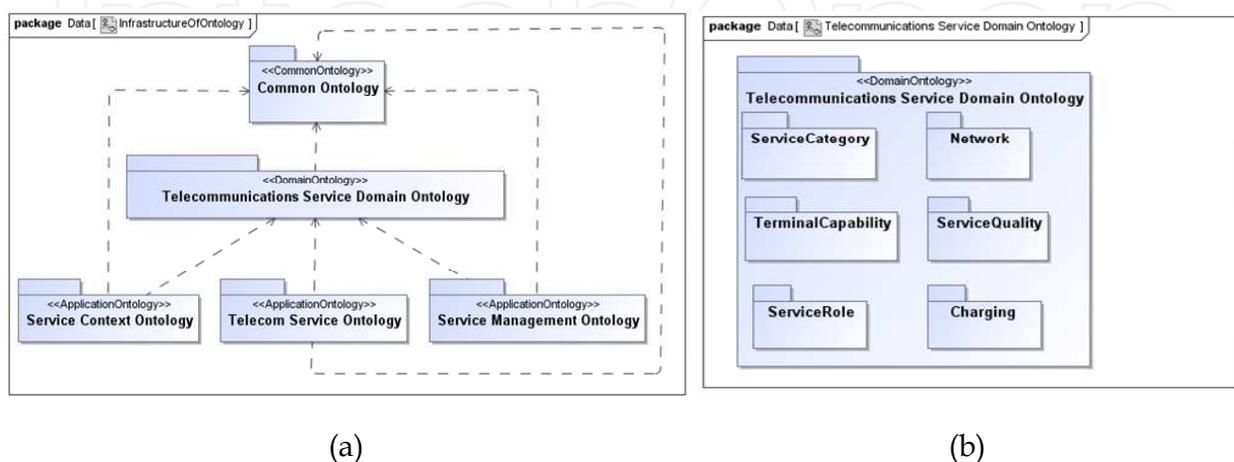


Fig. 11. CIM modelling of telecommunications service domain ontology.

2.2.3 PIM step: Terminology acquisitions and conceptual modeling of TSDO

After defining the CIM of TSDO, the following step is to construct the PIM of TSDO. It means that the telecom service related domain terminologies should be collected and then integrated into a high-level abstract domain ontology model which is independent of a specific ontology language like OWL. The collection of domain terminologies can be modeled by the UML Use Case diagram like Figure 4. However, the high-level domain conceptual modeling is the emphasis of PIM. How to model the conceptual model of domain ontology based on UML is needed to resolve. Fortunately, UML and ontology language have some common features, although sometimes represented differently. This provides a possible transformation from UML model to ontology model. For example, both ontology representation language and UML are based on *Class*. The *Generalization* elements of UML can represent the subClass or subProperty semantic of ontology. The ownedAttribute of UML Class can describe the DatatypeProperty of ontology language. The mapping example is illustrated in the Figure 12.

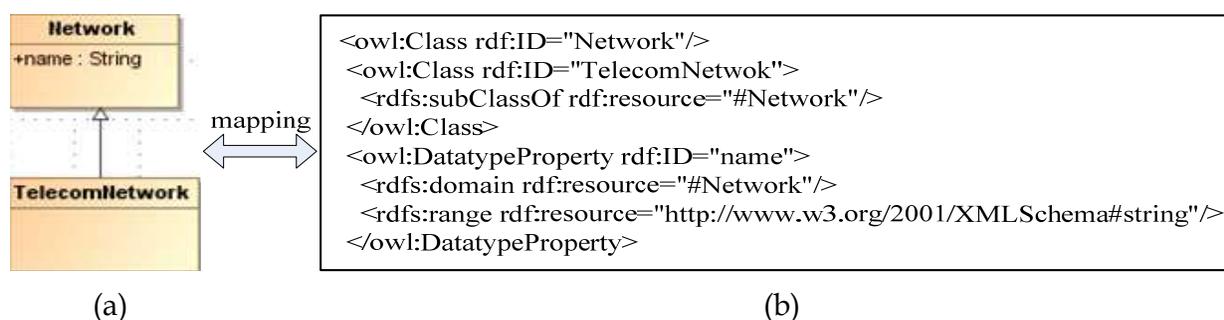


Fig. 12. The direct mapping example from UML to OWL.

However, although UML Class diagram has some constructs similar to the constructs of ontology representation language, there are still some ontology constructs which cannot be represented by UML constructs directly. We need to find the appropriate UML elements to represent some other ontology constructs, like objectProperty, equivalent class relation, and disjointing class relation. For instance, we can select the directedAssociation element of UML to represent the ObjectProperty and use the constraints anchored with association to represent the inverse, symmetric or transitive feature of ObjectProperty. An illustrated example is shown in Figure 13.

As a common software modeling language, most of software developers, system analysts and designers are familiar with UML. So, in order to decrease the technical threshold, it's a practical approach for the conceptual modeling of TSDO by UML. Although UML has some similar constructs with ontology language, however, the modeling goals and description capabilities of both languages have some differences. From the above analysis, in order to use UML to represent high-level ontology conceptual model, we need to define a specific tailored representation method to guide the modeler to build the conceptual model of domain ontology. Table 1 shows the main corresponding relation of UML elements with ontology elements. According to this semantic representation way, the modeler can use the UML elements to describe the semantic-enabled high-level ontology conceptual model like Figure 14.

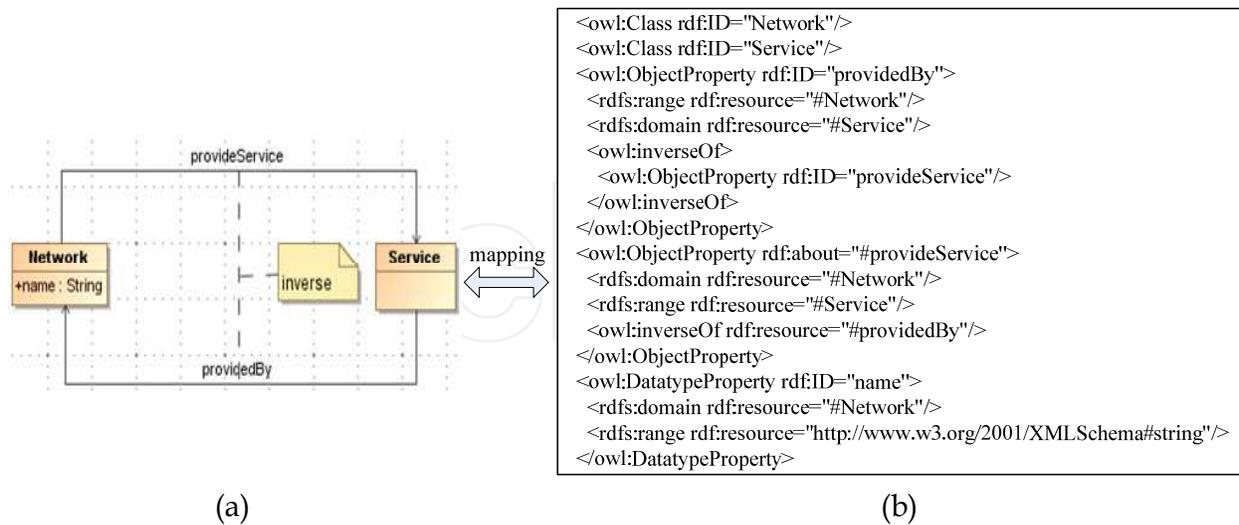


Fig. 13. The indirect mapping example from UML to OWL.

UML Elements	Ontology Elements	Comments
Class	Class	
Generalization	subClass, subProperty	
Instance	Individual	
Multiplicity	minCardinality maxCardinality	ontology cardinality declared only for range
ownedAttribute	Datatype Property	
directedAssociation	ObjectProperty	The value of "owned By" property of Association End A is the domain of ObjectProperty, the value of "owned By" property of Association End B is the range of ObjectProperty.
Constraint	Inverse Symmetric Transitive Functional	
Enumeration	oneOf	
Association Class	disjointWith equivalentClass	

Table 1. The defined UML representation method for high-level conceptual model of domain ontology.

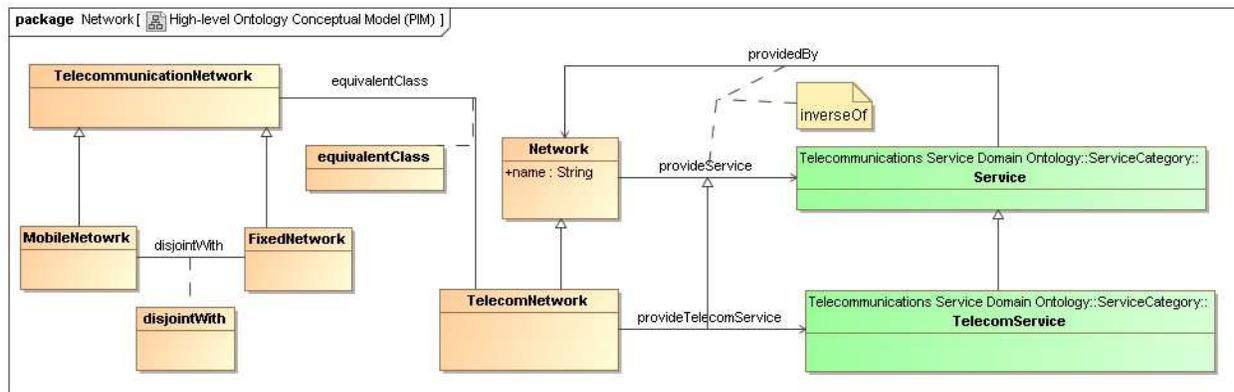


Fig. 14. PIM: A part of high-level conceptual model of network ontology.

2.2.4 PIM to PSM step: Formalization of ontology conceptual model

It can be seen that the high-level ontology conceptual model described by UML is independent of a specific ontology language. So, in order to generate the formal file encoded by a specific ontology language, we need to transform the PIM into PSM according to the concrete model transformation rules. Figure 15 shows the general model transformation mechanism of model driven architecture. Model transformation is essentially to map the source model elements to other elements of the target model. Models are usually the instantiation of its meta-model. The model transformation rules are generally defined in the metamodel level and then model transformation engine apply these rules to the model level to complete the model transformation.

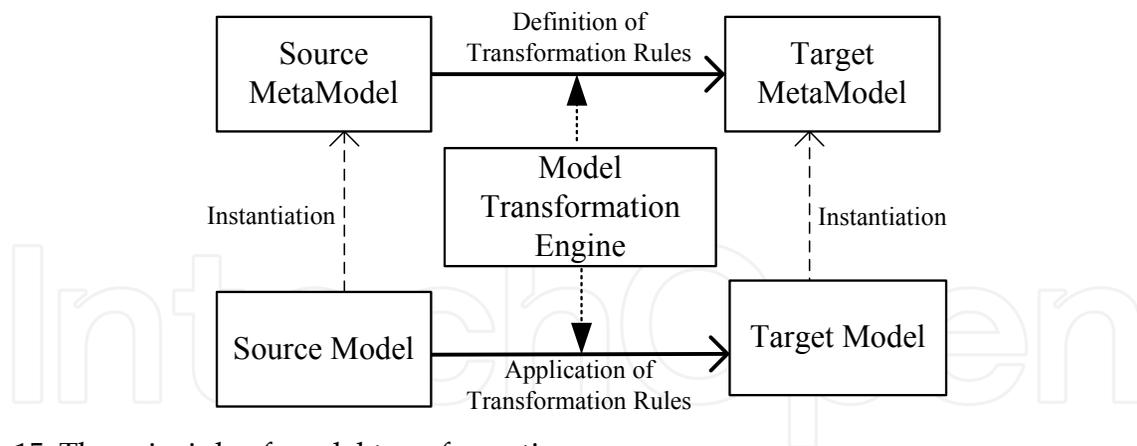


Fig. 15. The principle of model transformation.

Therefore, in order to transform the high-level ontology conceptual model (i.e. PIM) into platform specific model (i.e. PSM), we need to define the transformation rules according to the source and target metamodels. In our proposed approach, the high-level ontology conceptual model (i.e. PIM) is modeled by UML2.0, and the source metamodel is UML2.0 metamodel obviously. So we need a target metamodel relating to specific ontology language to describe the formal ontology model (i.e. PSM). In fact, OMG (Object Management Organization), which is the promoter of MDA, has considered this problem. In May 2009, OMG released the Ontology Definition Metamodel (ODM) v1.0 (OMG, 2009) based on the

meta-modeling mechanism of MDA. This specification represents the foundation for an extremely important set of enabling capabilities for MDA based software engineering, namely the formal grounding for representation, management, interoperability, and application of business semantics. The ODM is applicable to knowledge representation, conceptual modeling, formal taxonomy development and ontology definition, and enables the use of a variety of enterprise models as starting points for ontology development. ODM is based on the Meta Object Facility (MOF) (OMG, 2006) meta-modeling architecture of MDA, illustrated by Fig.16, which is based on the traditional four layer metadata architecture. From top to bottom, meta-data is abstracted to 4 layers: M3 (meta-meta model), M2 (meta model), M1 (model) and M0 (object and instance). The under-layer is the instance of its up-layer in turn. M3 layer is the end of meta-layer, namely, MOF is self-described. MOF is a common, abstract language used to define meta-model. It defines some meta-modeling constructs, such as Class, DataType, Association, Package, and Constraint. So the meta-model of ODM or UML can be defined by MOF, whose power just lies in its capability to enable interoperability among different meta-models. Currently, there are 2 approaches to construct meta-models in M2 layer. One is to make use of MOF to define a completely new meta-model from syntax to semantics. Although this approach supports to define a new meta-model that will perfectly match the concepts and relation of the concrete domain, this need the underlying programming realization of corresponding new modeling tool. This is heavy-weight meta-modeling, such as UML and ODM. The other is to extend the existent UML meta-model and then construct a standard UML Profile through UML extension mechanism (Stereotype, TaggedValue, Constraints). This approach allows both defining domain specific conception and relation through UML extension mechanism and using the intrinsic UML elements. So it's a light-weight meta-modeling approach and most of existent MDA tools support this UML Profiling-based meta-modeling mechanism currently. There is no need to develop a new modeling tool. From the above analysis, the UML Profiling-based meta-modeling mechanism approach is adopted in our approach.

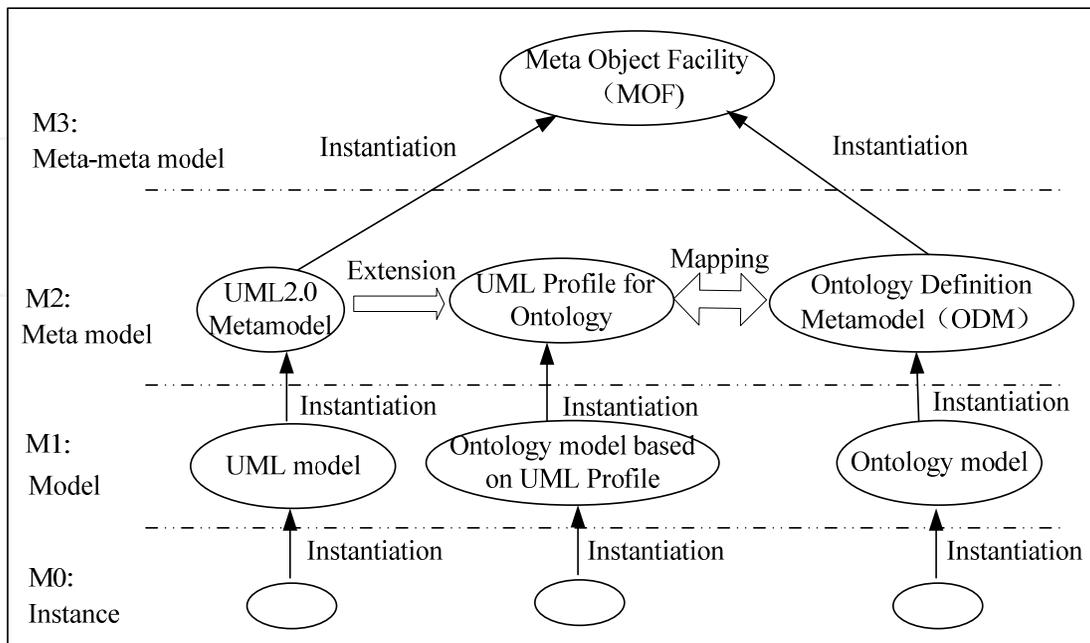


Fig. 16. ODM: the integration of semantic web and model driven architecture.

Therefore, in this approach, the metamodel of PSM employs the UML Profile for RDF and OWL defined in ODM specification. This profile is designed to support modelers developing vocabularies in Resource Description Framework (RDF) (W3C, 2004b) and richer ontologies in the Web Ontology Language (OWL) through reuse of UML notation using tools that support UML2 extension mechanisms. Table 2 specifies a part of stereotypes set that comprise the UML2 Profile for using UML to represent RDF/S and OWL vocabularies.

RDF, RDFS and OWL ontology	UML Base Class	UML Stereotype
rdfs:Resource	Class	«rdfsResource»
rdfs:Datatype	Class	«rdfsDatatype»
rdfs:domain	Association	«rdfsDomain»
rdfs:range	Association	«rdfsRange»
rdfs:subClassOf	Generalization	«rdfsSubClassOf»
rdfs:subPropertyOf	Generalization	«rdfsSubPropertyOf»
owl:Class	Class	«owlClass»
owl:Restriction	Class	«owlRestriction»
owl:ObjectProperty	Class AssociationClass Property Association	«objectProperty»
owl:DatatypeProperty	Class AssociationClass Property Association	«datatypeProperty»
owl:equivalentClass	Constraint	«equivalentClass»
owl:disjointWith	Constraint	«disjointWith»

Table 2. A part of UML Profile for RDF and OWL.

After the source and target metamodels are determined, we can define the model transformation rules from high-level ontology conceptual model (i.e. PIM) to ontology language related model (i.e. PSM). For example, based on the Table 1 and Table 2, we can define the following model transformation rules to support the model transformation like Figure 17. Notably, the source metamodel is UML2.0 metamodel and the target metamodel is UML Profile for RDF and OWL in this proposed approach.

When the transformation rules are defined, the model transformation engine can scan the elements of source model and then transform them into the corresponding elements of target model according to the transformation rules. As model transformation is a key technique used in model-driven architecture. In 2002, OMG issued a Request for proposal (RFP) on MOF Query/View/Transformation to seek a standard compatible with the MDA recommendation suite (UML, MOF, OCL, etc.). Several replies were given by a number of companies and research institutions that evolved during three years to produce a common proposal that was submitted and approved. QVT (Query/View/Transformation) (OMG,

2008) is a standard set of languages for model transformation defined by the Object Management Group. Currently, some MDA tools have declared to support the complete or part functions of QVT. For example, by using the transformation rules, the source model in Figure 14 is transformed into a target model in Figure 18.

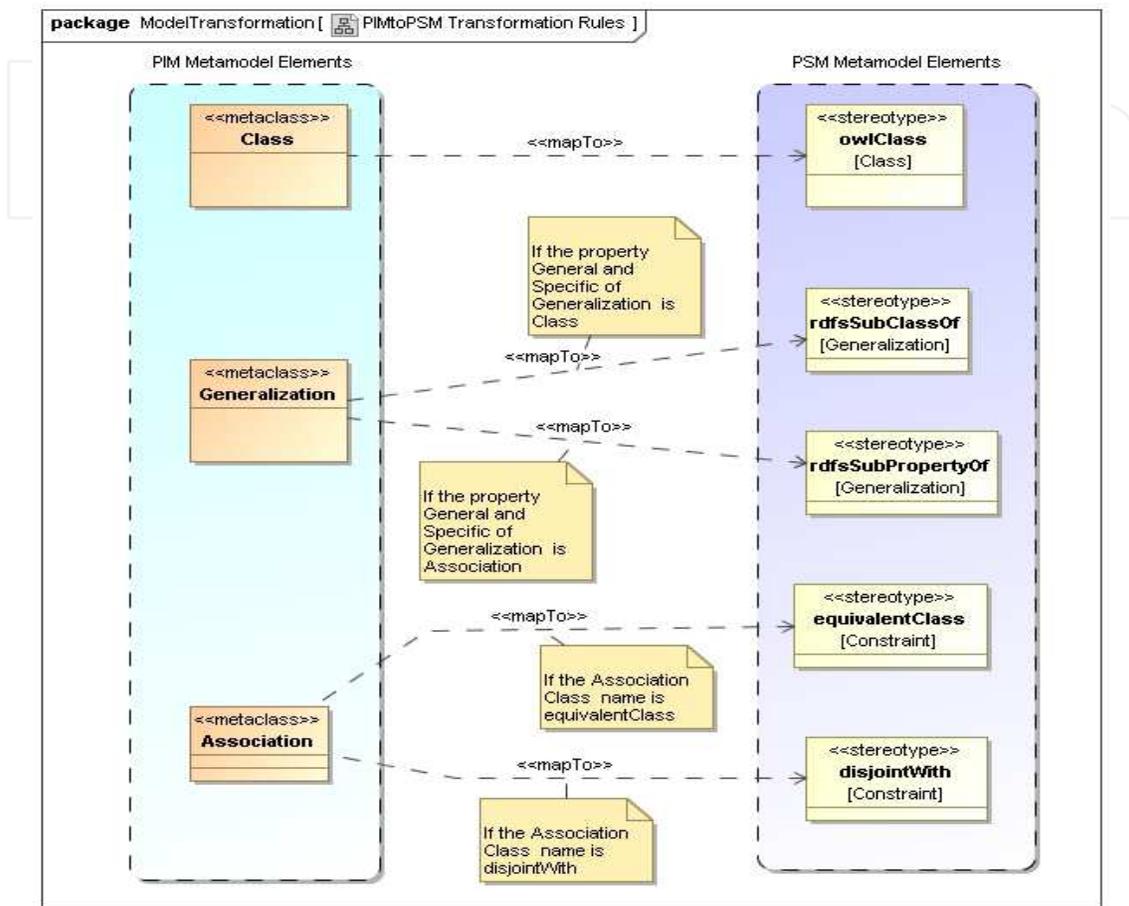


Fig. 17. A part of PIM to PSM transformation rules definitions.

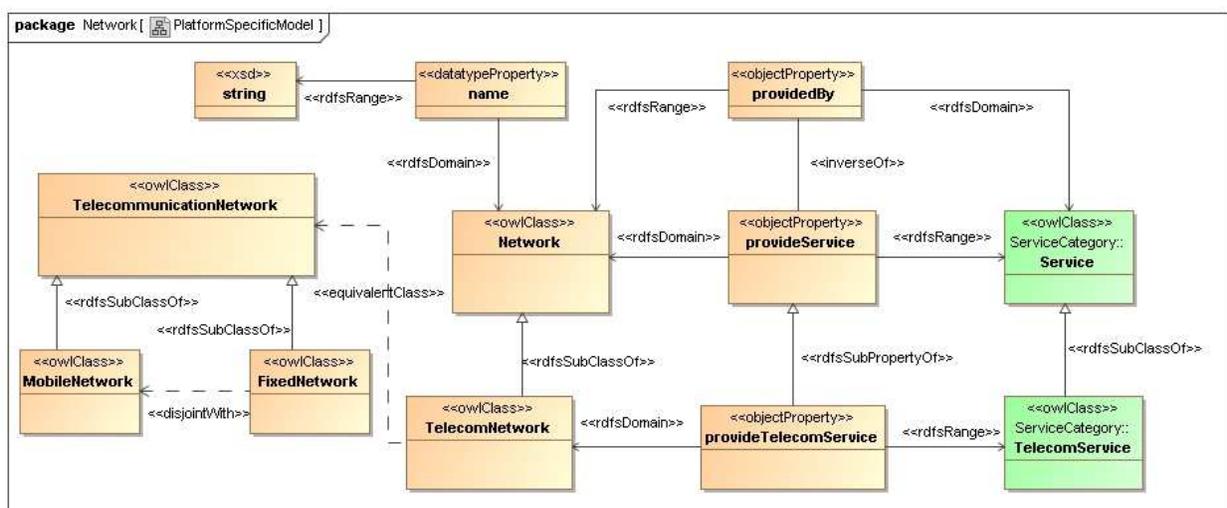


Fig. 18. PSM: A part of ontology specific model based on UML Profile for RDF and OWL.

2.2.5 PSM to code step: Formalization of ontology conceptual model

In order to generate the formal ontology file encoded by OWL, the PSM based on UML Profile for RDFS and OWL should be transformed into ontology file encoded by OWL, which is a model to code transformation process, which involves the model scanning technology.

2.2.5.1 Model to code transformation theory

Before we introduce the concrete transformation process, some related definitions are given firstly.

Definition 1. *Ontology model triples: UmlOnt (C, R, G).*

PSM based on the UML class diagram can be represented by a triple: UmlOnt(C, R, G). C is the class node set, and it is an ontology class definition of the concept in PSM. R is the relation node set, and it is the definition of the relations among ontology class. G is the relation set of C and R, which describes the relations among the nodes in C and R set.

Definition 2. *Relation Matrix (RM)*

Relation matrix is a n order square matrix, including elements a_{ij} in total of $n * n$, which

looks like
$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$
, denoted as $A = (a_{ij})_{nn}$ ($i, j = 1, 2, 3, \dots, n$) is used to

describe the relations between the class nodes and relation nodes in the PSM. And a_{ij} indicates whether there is relation between class node i and class node j , as well as the type of relation node.

Definition 3. *Connected subgraph*

Given a directed graph, which can be divided to several connected subgraphs $\{G_1, G_2, \dots, G_n\}$, these subgraphs meet the following conditions:

- In any two connected subgraphs G_i and G_j , there is no such a node x which is both in G_i and G_j at the same time. That is, $\neg \exists x, x \in G_i \cap x \in G_j$.
- In a connected subgraph G_i , there always exist directed edge between any two nodes x and y (without regard to the direction of the edge).

Definition 4. *Model Transformation Automaton (MTA)*

Model transformation automaton is a quintuple: $MTA = (Q, \Sigma, \delta, q_0, F)$, including:

- Q : A nonempty and finite set of states and one state in it corresponds to an ontology class node. $\forall q \in Q, q$ is called a state of MTA.
- Σ : Input events table, in which one input event corresponds to an ontology relation node.
- δ : Transfer function. One transfer function corresponds to a nonzero number a_{ij} ($a_{ij} \neq 0$), $\delta : Q \times \Sigma \rightarrow Q$ in relation matrix.
- q_0 : The begin state of MTA, $q_0 \in Q$.

- F: The set of terminate states. F is included by Q. Any $q \in F$, q is called a terminate state of MTA.

According to the definitions given above, the transformation engine from PSM to formal file encoded by OWL can be described as: The model transformation engine firstly scans the PSM class graph, and the scanning result generates the ontology model triples $UmlOnt(C, R, G)$. C is the set of all nodes in UML class graph, R is the set of all relations in UML class graph, G represents the structure relationship of the ontology class graph, which can be regarded as N connected subgraphs divided from a directed graph and these subgraphs correspond to N relation matrices $\{RM1, RM2, \dots, RMn\}$. One nonzero number a_{ij} represents the relation type between the class node i and j, and these relations are all included in R.

When the transformation engine finishes scanning, it input the scan result to the model transformation automaton. In this MTA, the nonempty finite set of states corresponds to C in the ontology model triples; the input events table corresponds to R in the ontology model triples; the transfer function corresponds to G in the ontology model triples; q_0 and F are elements in C. In the procedure of state transforming, the corresponding operations of model transformation are also performed in MTA. When the automaton arrives at the terminal, the transformation finishes.

2.2.5.2 The implementation mechanism of model to code transformation engine

In order to realize the model to code transformation according to the above mentioned theory, we design a model transformation engine based on Eclipse Plugin technology. In MDA, XML Metadata Interchange (XMI) (OMG, 2005b) is an Object Management Group (OMG) standard for exchanging metadata information via Extensible Markup Language (XML). As the most of MDA tools use XMI as an interchange format for UML models, the model transformation engine is responsible for scanning PSM encoded by XMI and then transforming PSM into ontology file encoded by OWL. The process of model to code transformation is indicated in Figure 19.

2.2.5.2.1 Model scanning module

When building ontology model, different modeling tool means different element label and different label structure in the model description file. Therefore, this chapter proposed a transitional model convert method, which adopts same data structure when describes different model format, i.e. the triples in *Definition 1*. This allows the model transformation is no longer constrained by the model structure. It thereby improves the versatility of transformation engine and is convenient to be maintained and updated.

The model scanning module in transformation engine scans the UML class graph encoded by XMI, and the scan result will generate two list sets in the transitional model. It is used to store the class nodes and relation nodes of UML graph, which corresponds to the C and R set in the UML ontology model triples.

2.2.5.2.2 Building relation matrix module

The function of relation matrix building module is used to generate the relation matrix of PSM, i.e. the G set in UML ontology model triples. There are mainly two kinds of nodes in

UML class graph: class node and relation node. Relation node connects class node and distinguishes them by direction, which is very similar to the directed graph. Hence directed graph is adopted to represent UML class graph. Usually matrix is used to represent the graph, and the values in matrix represent the type of relation. In ontology model class graph, there may be several independent subgraphs, which satisfy the description in definition 2. Therefore, it is necessary to handle the relation matrix to produce N independent sub relation matrices. This method can reduce the order or relation matrix and thereby reduce store space of the model, which also improves the efficiency of model transformation.

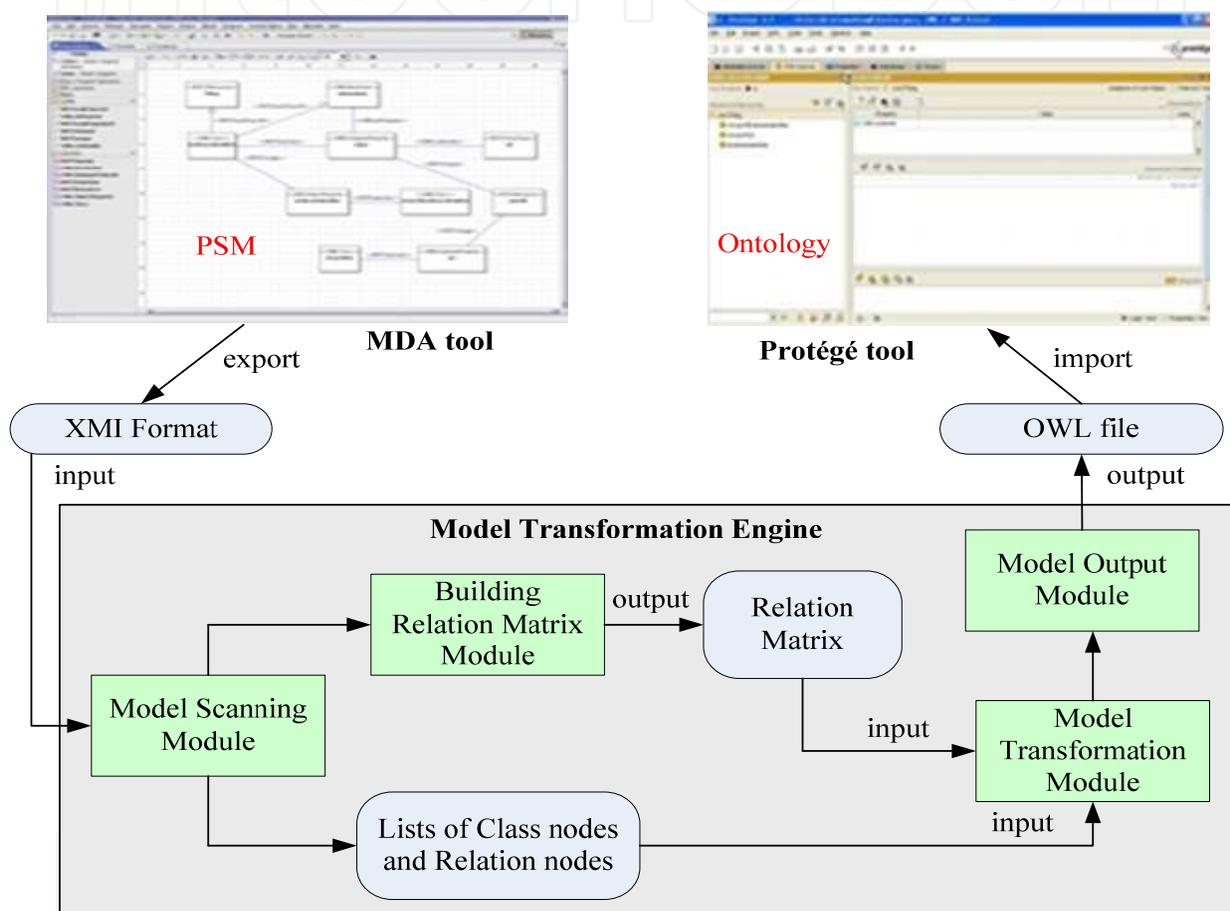


Fig. 19. Model to code transformation process.

2.2.5.2.3 Model transformation module

Model transformation module includes model transformation automaton and model transformation regulation table. In the process of states transition, the automaton performs transformation from PSM to OWL according to the corresponding transformation regulations. In this module, the automaton is separated with the model transformation regulations. Therefore, the changes of model transform regulation will not influence the running of automaton, and it is convenient to perform daily maintenance and update of the engine.

The model transformation automaton is a quintuple, and all information of this quintuple are included in the model transform transitional model, namely in the UML ontology

description model triples (C, R, G). The states, input events and transformation function of the automaton correspond to the class nodes, relation nodes and relation matrix set respectively in PSM class graph. The begin state of automaton is the owlClass node or objectProperty/datatypeProperty node in class nodes and the terminate states set includes all class nodes whose out-degree are zero and all nodes have been transformed by the automaton.

The model transformation regulation table defines the transformation regulation from UML Profile for RDF and OWL to OWL language. In the states jump process of model transformation automaton, corresponding regulation is used to perform model transformation. In the process of formulating the transformation regulation, the relations between every label node should be unified, which makes the regulation can be formulated depending on the OWL label structure and the relations between UML model elements. And good regulation is easy to extend in the future.

2.2.5.2.4 Model output module

Model output module only stores the formalized result of the transformation to the appointed path. And in order to verify the validity of the OWL file transformed, user can import the generate code into protégé tool for verification. Protégé is an ontology editor developed by Stanford University, which represents the OWL structure in graphic interface and makes the verification of OWL code validity more quickly and conveniently.

By using the above mentioned model to code transformation approach, the PSM of Figure 18 is transformed into the corresponding ontology encoded by OWL like Figure 20.

3. Experimental environment, use cases and evaluation

In this section, we describe our experimental environment, the implemented service use case and present the obtained evaluation results to validate the semantic interoperability enabled by telecommunications service domain ontology.

3.1 Experimental environment

In order to support this model-driven domain ontology modelling approach, Borland Together (Borland, 2006), a famous MDA tool, is employed in our experiment. By using UML extension mechanism, we implemented the UML Profile for RDF and OWL in Borland Together. In addition, Borland Together tool enable the model-to-model transformation, and this facilitates the transformation from PIM to PSM. Through the developed model-to-code transformation engine, we realize the transformation from PSM to ontology file encoded by OWL. At last, in order to verify whether the transformation is correct or not, the generated OWL file is imported in Protégé tool to test. By the experimental verification, the proposed model-driven ontology modelling approach can nicely support the constructing methodology of telecommunications service domain ontology.

Under the guidance of this approach, our research team has created a telecommunications service domain ontology knowledge repository which consists of around 430 telecommunications services-related ontology concepts/terminologies and 245 properties. Currently, these ontologies are published on our website (BUPT, 2009), see Figure 21.

```

<owl:Class rdf:ID="MobileNetwork">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="TelecommunicationNetwork"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="FixedNetwork"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="TelecomNetwork">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Network"/>
  </rdfs:subClassOf>
  <owl:equivalentClass>
    <owl:Class rdf:about="#TelecommunicationNetwork"/>
  </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:about="#FixedNetwork">
  <owl:disjointWith rdf:resource="#MobileNetwork"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#TelecommunicationNetwork"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#TelecommunicationNetwork">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <owl:equivalentClass rdf:resource="#TelecomNetwork"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="provideTelecomService">
  <rdfs:range rdf:resource="&ServiceCategeory; #TelecomService"/>
  <rdfs:domain rdf:resource="#TelecomNetwork"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="provideService"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="prvidedBy">
  <rdfs:domain rdf:resource="&ServiceCategeory; #Service"/>
  <rdfs:range rdf:resource="#Network"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="#provideService"/>
  </owl:inverseOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#provideService">
  <rdfs:range rdf:resource="&ServiceCategeory; #Service"/>
  <owl:inverseOf rdf:resource="#prvidedBy"/>
  <rdfs:domain rdf:resource="#Network"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="name">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Network"/>
</owl:DatatypeProperty>

```

Fig. 20. A part of formal network ontology encoded by OWL.

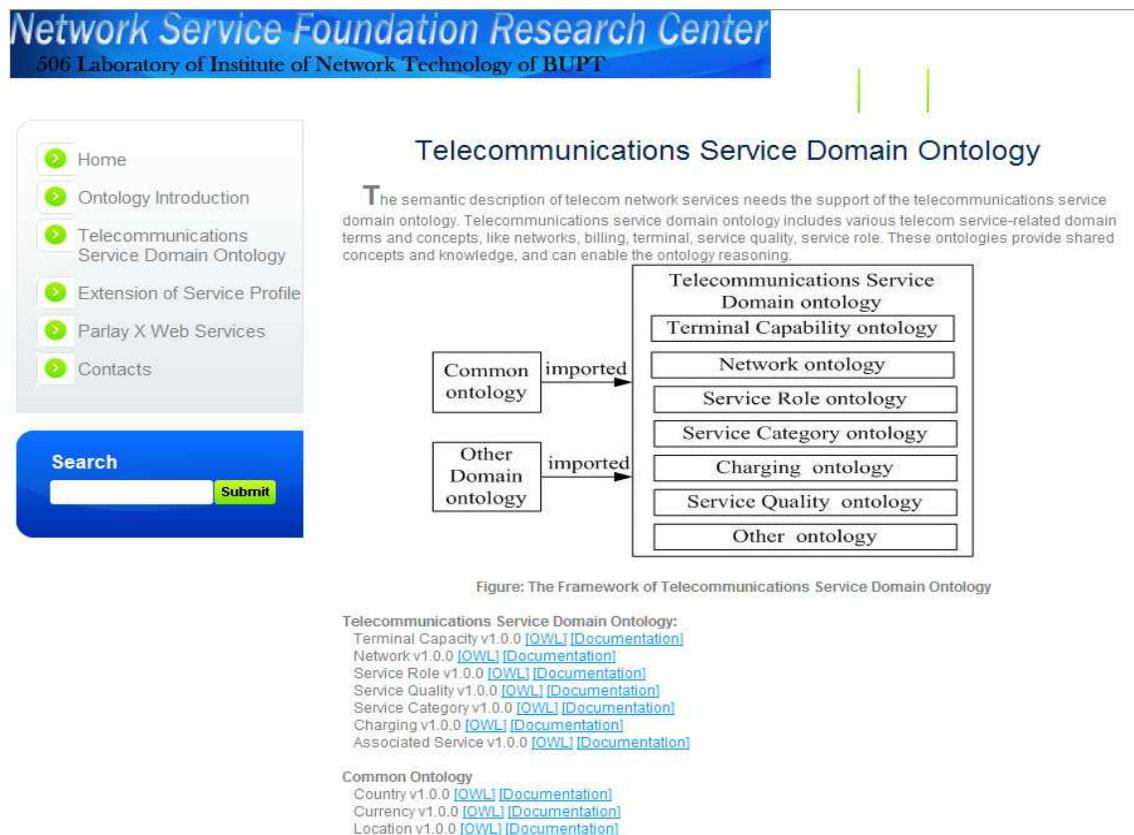


Fig. 21. The published telecommunications service domain ontology.

3.2 Use cases: Semantic telecommunications network capability services

In order to support the shift from traditional closed business model to open service ecosystem of telecom industry, NGN (Next Generation Network) and 3G network all adopt the open API (Application Programming Interface) technologies in the service layer, such as Parlay/OSA and Parlay X (Moerdijk & Klostermann, 2003). Thus, the telecommunication network services, such as call control, short messaging service, and location service, are available to the service developers in the form of APIs. This facilitates the value-added service development. With the development of distributed computing technology, Service-Oriented Architecture (SOA) is also imported into the telecommunications service domain by Parlay Web Service specifications. However, the open interface specifications of telecommunication networks are currently still in the syntactic level. As WSDL (Web Services Description Language)-based telecommunication network services lack the rich semantic annotation information, the keyword-based service matching cannot enable an accurate service discovery. So, currently value-added services often directly invoke the needed telecom network services provided by a specific network carrier. This results in the tight-coupling of application logic and service resources, which limits the provision of dynamically self-adaptive services. The applications cannot dynamically discover satisfied telecom network services and compose them according to the context environment. Facing the heterogeneous networks and personalized user demands, the self-adaptation has become a very important feature of future intelligent integrated service. Therefore, the semantic interoperability of telecom network and Internet in the service layer should be considered.

Based on this domain ontology, we described the telecom network capability services in the semantic level to validate its feasibility. We apply the semantic web service and ontology technologies to the telecommunications service domain, and present an infrastructure to enable the semantic interoperability of telecom network and Internet in the service layer (Qiao et al., 2008b). The proposed approach improves the accuracy of telecommunication network services description, discovery and matching, and unifies the semantic representations of telecommunication and Internet services.

3.3 Lessons learned

Currently, under the shift trend from Web2.0 to Web3.0 era, there have been some initial semantic web applications in Internet field. For example, the system of Twitter allows tweets to be tagged with information that will not appear in the message but can be read by computers (Twitter, 2010). Google is using structured data open standards such as microformats and RDFa to power the rich snippets feature. It's an experimental Semantic Web feature (Google, 2010). FOAF (Friend of a Friend) (FOAF, 2010) is a machine-readable ontology describing persons, their activities and their relations to other people and objects. As a "practical experiment" in the application of RDF and Semantic Web technologies to social networking, FOAF is becoming more and more popular now (FOAF, 2000). In addition, Linked Data (Linked Data, 2007) is a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF.

However, the semantic web applications in telecommunication services domain are still in an early research phase. Although RDF-based CC/PP (Composite Capability/Preference Profiles) (W3C, 2007) and UAProf (User Agent Profile) (OMA, 2001) are used to describe the terminal capability and user preference, other practical applications are very rare. Therefore, in order to eliminate the semantic gap between telecom network and Internet, the research on semantic web applications in telecommunications field still need to be further enhanced. Telecommunications service domain ontologies consist of various domain related concepts and knowledge, which is the base of semantic interoperability. The wide acceptance of standards and common practices of telecommunications service domain ontologies are still a way ahead. The promotion of the telecommunications service domain ontology by related standardization organizations would be in the foundation for the semantic interoperability of heterogeneous communications equipments and the industrial practical convergent service integration.

4. Conclusion

The network heterogeneity and service convergence are the main characteristics of future network. The provision of self-adaptive intelligent integrated services has become the pursuing goal of network carriers and value-added service providers. Dynamic discovery and composition of services are the important enabling technologies for self-adaptive integrated services. In the service discovery and composition process, semantic interoperability is a key issue. Actually, ontology, as a semantic interoperability and knowledge sharing foundation, has obtained more and more attentions. However, telecommunication service field consists of a large number of concepts/terminologies and

relations. How to abstract the sharing domain concepts and reasonably organize them is a big challenge. In this chapter, we presented a practical domain ontology modelling approach for telecommunications service field. Based on this approach, we constructed an open telecommunication service domain ontology repository to support the knowledge sharing and reuse. This will partly facilitate the semantic interoperability of the telecommunications networks and the Internet in the service layer.

5. Acknowledgment

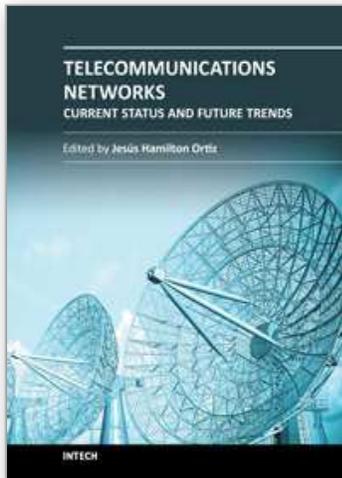
This work was supported by National Key Basic Research Program of China (973 Program) under Grant No. 2012CB315802, National Natural Science Foundation of China under Grant No. 60802034, No. 61171102 and No. 61132001, Beijing Nova Program under Grant No. 2008B50 and New generation broadband wireless mobile communication network Key Projects for Science and Technology Development under Grant No. 2011ZX03002-002-01. We thank Huawei Technologies Co., Ltd. for cooperation in promotion of this work. Thanks also to Dr. Anna Fensel, a Senior Researcher at FTW – Telecommunications Research Center Vienna and STI Innsbruck, University of Innsbruck, Austria, for her valuable comments and suggestions.

6. References

- Bashah, N.S.K., Jorstad, I. & Thanh, D.V. (2010). Service Discovery in Future Open Mobile Environments. *Proceedings of ICDS 2010 Fourth International Conference on Digital Society*, pp.47-53, ISBN 978-1-4244-5805-9, St. Maarten, Netherlands Antilles, February 10-16, 2010
- Borland, (2006). Together: Visual Modeling for Software Architecture Design. 26.09.2011, Available from <http://www.borland.com/us/products/together/>
- BUPT, (2009). Semantic web application for telecommunications service. 26.09.2011, Available from <http://www.int.bupt.cn/jsp/centers/bupt506/intro.htm>
- Do, T.V., Jorstad, I. (2005). A service-oriented architecture framework for mobile services. *Proceedings of Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop AICT/SAPIR/ELETE*, pp. 65 – 70, ISBN 0-7695-2388-9, Lisbon, Portugal, July 17-20, 2005
- FOAF, (2010). FOAF Vocabulary Specification. 26.09.2011, Available from <http://xmlns.com/foaf/spec/>
- FOAF project, (2000). The Friend of a Friend (FOAF) project. 26.09.2011, Available from <http://www.foaf-project.org/>
- Google, (2010). Google's Semantic Web Push: Rich Snippets Usage Growing. 26.09.2011, Available from http://www.readwriteweb.com/archives/google_semantic_web_push_rich_snippets_usage_grow.php
- Gutheim, P. (2011). An ontology-based context inference service for mobile applications in next-generation networks. *IEEE Communications Magazine*, Vol. 49, No. 1, (Jan. 2011), pp. 60 – 66, ISSN 0163-6804
- IST SPICE project, (2008). 26.09.2011, Available from <http://www.ist-spice.org/>

- Khan, A., Asghar, S. and Fong, S. (2011). Framework of integrated Semantic Web Services and Ontology Development for Telecommunication Industry. *Journal of Emerging Technologies in Web Intelligence*, Vol. 3, No. 2, (May 2011), pp.110-119, ISSN 1798-0461
- Kolberg, M., Merabti, M. & Moyer, S. (2010). Consumer communication applications drive integration and convergence. *IEEE Communications Magazine*, Vol.48, No. 12, (December 2010), pp. 24 - 24, ISSN 0163-6804
- Niazi, R., Mahmoud Q.H. (2009). An Ontology-Based Framework for Discovering Mobile Services. *Proceedings of CNSR 2009 Seventh Annual Communication Networks and Services Research Conference*, pp.178-184, ISBN 978-0-7695-3649-1, Moncton, New Brunswick, Canada, May 11-13, 2009
- Li, X.F., Peng, H., Li, Y. & Qiao, X.Q. (2010). Research on Constructing Telecom Services Domain Ontology. *Proceedings of 2010 International Conference on Computational Intelligence and Software Engineering (CiSE)*, pp.1-4, ISBN 978-1-4244-5391-7, Wuhan, China, Dec. 10-12, 2010
- Linked Data, (2007). Linked Data - Connect Distributed Data across the Web. 26.09.2011, Available from <http://linkeddata.org/>
- McIlraith, S.A., Son, T.C. & Zeng H.L. (2001). Semantic Web services. *IEEE Intelligent Systems*, Vol. 16, No.2, (Mar.-Apr. 2001), pp. 46 - 53, ISSN 1541-1672
- Miller, J. & Mukerji, J. (2003). MDA Guide V1.0.1. 26.09.2011, Available from <http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- Moerdijk, A.-J. & Klostermann, L. (2003). Opening the Networks with Parlay/OSA: Standards and Aspects Behind the APIs, *IEEE Network*, vol.17, no.3, (May-June 2003) pp: 58-64, ISSN 0890-8044
- OMA, (2001). User Agent Profile. 26.09.2011, Available from <http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf>
- OMG. (2003). Model Driven Architecture. 26.09.2011, Available from <http://www.omg.org/mda/>
- OMG. (2005). Unified Modeling Language. 26.09.2011, Available from <http://www.omg.org/spec/UML/>
- OMG. (2005). XML Metadata Interchange. 26.09.2011, Available from <http://www.omg.org/spec/XMI/>
- OMG. (2006). Meta Object Facility. 26.09.2011, Available from <http://www.omg.org/mof/>
- OMG. (2008). Meta Object Facility (MOF) 2.0 Query/View/Transformation (QVT). 26.09.2011, Available from <http://www.omg.org/spec/QVT/index.htm>
- OMG. (2009). Ontology Definition Metamodel (ODM). 26.09.2011, Available from <http://www.omg.org/spec/ODM/1.0/>
- Park, K.L., Yoon, U.H. & Kim S.D. (2009). Personalized Service Discovery in Ubiquitous Computing Environments. *IEEE Pervasive Computing*, Vol.8, No.1, (Jan.-March 2009), pp.58 - 65, ISSN 1536-1268
- Qiao, X.Q., Li X.F. and You, T. (2008). A Semantic Description Approach for Telecommunications Network Capability Services. *Proceedings of 11th Asia-Pacific Network Operations and Management Symposium*, pp.334-343, ISBN 978-3-540-88622-8, Beijing, China, October 22-24, 2008

- Qiao, X.Q., Li, X.F., You, T., Sun, L.H. (2008). Semantic Telecommunications Network Capability Services. *Proceedings of the 3rd Asian Semantic Web Conference (ASWC 2008)*. pp. 508–523, ISBN 978-3-540-89703-3, Bangkok, Thailand, Dec. 8-11 2008
- Rój, M. (2008). Recommendations for telecommunications services ontology, In: *IST SIMS project deliverables*, 26.09.2011, Available from <http://www.ist-sims.org>.
- Selic, B. (2003). The pragmatics of model-driven development. *IEEE Software*, Vol. 20, No. 5, (Sept.-Oct. 2003), pp. 19 – 25, ISSN 0740-7459
- Stanford Protégé team. (2004). Protégé. 26.09.2011, Available from <http://protege.stanford.edu/>
- Su X.M., Alapnes, S. and Shiaa M.M. (2009). Mobile Ontology: Its Creation and Its Usage, In: *Constructing Ambient Intelligence*, H. Gerhauser, J. Hupp, C. Efstratiou and J. Heppner (Eds.), pp.75-79, Springer-Verlag Berlin Heidelberg, ISBN 3-642-10606-4, Germany
- Twitter, (2010). Semantics, tagging and Twitter. 26.09.2011, Available from <http://ml.sun.ac.za/2010/04/23/semantics-tagging-and-twitter/>
- Veijalainen, J. (2007) Developing Mobile Ontologies; Who, Why, Where, and How? *Proceedings of 2007 International Conference on Mobile Data Management*, pp. 398 – 401, ISBN 1-4244-1241-2, Mannheim, Germany, May 1-1, 2007
- Veijalainen, J. (2008). Mobile Ontologies: Concept, Development, Usage, and Business Potential. *International Journal on Semantic Web and Information Systems (IJSWIS)*, Vol. 4, No. 1, (Sep. 2008), pp.20-34, ISSN: 1552-6283
- Villalonga, C., Strohbach, M., Snoeck, N., Sutterer, M., Belaunde, M., Kovacs, E., Zhdanova, A.V., Goix, L.W., Droegehorn, O. (2009). Mobile Ontology: Towards a Standardized Semantic Model for the Mobile Domain. *Proceedings of the 1st International Workshop on Telecom Service Oriented Architectures (TSOA 2007) at the 5th International Conference on Service-Oriented Computing*, pp. 248-257, ISBN, 978-3-540-93850-7. Vienna, Austria, September 17, 2007
- Vitvar, T., Viskova, J. (2005). Semantic-enabled Integration of Voice and Data Services: Telecommunication Use Case, *Proceedings of 2005 IEEE European Conference on Web Services (ECOWS2005)*, pp. 138-151, ISBN 0-7695-2484-2, Vaxjo, Sweden, November14-16, 2005
- W3C. (2004). OWL Web Ontology Language Overview. 26.09.2011, Available from <http://www.w3.org/TR/owl-features/>
- W3C. (2004). Resource Description Framework (RDF). 26.09.2011, Available from <http://www.w3.org/RDF/>
- W3C, (2007). Composite Capabilities/Preference Profiles: Structure and Vocabularies 2.0. 26.09.2011, Available from <http://www.w3.org/Mobile/CCPP/>
- Zander, S., Schandl, B. (2011). Semantic Web-enhanced Context-aware Computing in Mobile Systems: Principles and Application. In: *Mobile Computing Techniques in Emerging Markets: Systems, Applications and Services*, Kumar, A.V. Senthil (Eds.), Retrieved from <http://eprints.cs.univie.ac.at/2870/>
- Zhu, J.W., Li, B., Wang, F., Wang, S.CH. (2010). An Overview of CRS4MO Project: Construction and Retrieval System for Mobile Ontology. *Journal of Computational Information Systems*, Vol.6, No.6, (June, 2010), pp. 2009-2015, ISSN 1553-9105



Telecommunications Networks - Current Status and Future Trends

Edited by Dr. Jesús Ortiz

ISBN 978-953-51-0341-7

Hard cover, 446 pages

Publisher InTech

Published online 30, March, 2012

Published in print edition March, 2012

This book guides readers through the basics of rapidly emerging networks to more advanced concepts and future expectations of Telecommunications Networks. It identifies and examines the most pressing research issues in Telecommunications and it contains chapters written by leading researchers, academics and industry professionals. Telecommunications Networks - Current Status and Future Trends covers surveys of recent publications that investigate key areas of interest such as: IMS, eTOM, 3G/4G, optimization problems, modeling, simulation, quality of service, etc. This book, that is suitable for both PhD and master students, is organized into six sections: New Generation Networks, Quality of Services, Sensor Networks, Telecommunications, Traffic Engineering and Routing.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Xiuquan Qiao, Xiaofeng Li and Junliang Chen (2012). Telecommunications Service Domain Ontology: Semantic Interoperation Foundation of Intelligent Integrated Services, Telecommunications Networks - Current Status and Future Trends, Dr. Jesús Ortiz (Ed.), ISBN: 978-953-51-0341-7, InTech, Available from: <http://www.intechopen.com/books/telecommunications-networks-current-status-and-future-trends/telecommunications-service-domain-ontology-the-semantic-interoperation-foundation-of-intelligent-int>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen