Ontologizing Lexicon Access Functions based on a LMF-based Lexicon Taxonomy

Yoshihiko Hayashi[1], Chiharu Narawa[2], Monica Monachini[3], Claudia Soria[3], Nicoletta Calzolari[3]

[1]Graduate School of Language and Culture, Osaka University and NICT Language Grid Project

1-8 Machikaneyama, Toyonaka 560-0043, Japan

hayashi@lang.osaka-u.ac.jp

2 Department of Social Informatics, Kyoto University

Yoshida-Honmachi, Kyoto 606-8501, Japan

narawa@ai.soc.i.kyoto-u.ac.jp

3Instituto di Linguistica Computazionale, Consiglio Nazionaledelle Ricerche

Via Moruzzi 1, 56124 Pisa - Italy

monica.monachini, claudia.soria, nicoletta.calzolari@ilc.cnr.it

Abstract

This paper discusses *ontologization* of lexicon access functions in the context of a service-oriented language infrastructure, such as the Language Grid. In such a language infrastructure, an access function to a lexical resource, embodied as an atomic Web service, plays a crucially important role in composing a composite Web service tailored to a user's specific requirement. To facilitate the composition process involving service discovery, planning and invocation, the language infrastructure should be ontology-based; hence the ontologization of a range of lexicon functions is highly required. In a service-oriented environment, lexical resources however can be classified from a service-oriented perspective rather than from a lexicographically motivated standard. Hence to address the issue of interoperability, the taxonomy for lexical resources should be ground to principled and shared lexicon ontology. To do this, we have ontologized the standardized lexicon modeling framework LMF, and utilized it as a foundation to stipulate the service-oriented lexicon taxonomy and the corresponding ontology for lexicon access functions. This paper also examines a possible solution to fill the gap between the ontological descriptions and the actual Web service API by adopting a W3C recommendation SAWSDL, with which Web service descriptions can be linked with the domain ontology.

1. Introduction

Given a situation where a wide variety of language data resources and natural language processing (NLP) tools/systems have been actively disseminated, a strong need for a Web-based language infrastructure, on which tailored *language services* can be efficiently composed, disseminated and consumed, is becoming clearer. Here a language service simply means a Web service whose functionalities are somehow related to human language. We have been working on a domain ontology called *language service ontology* that can serve as a common ground for describing elements of a composite language service (Hayashi et al., 2008). As access functionalities to a range of lexical resources, or lexicons, are considerably important in composing a useful composite language service, we need to have a proper sub-ontology for stipulating a range of lexicon access functions.

The sub-ontology of lexicon access functions in the language service ontology should be derived by basing on the natures of lexicon access functions; a class of lexicon access function is basically defined by type of the query (input) and the corresponding results (output), as well as type of the target lexicon (language resource). This means that a proper taxonomy of lexicons considering these dimensions is quite important in defining proper sub-ontology for the lexicon access functions. Here, from the perspective of interoperability, it should also be noted that the taxonomy should be grounded to some shared standard like LMF (Lexical Markup Framework) (Francopoulo et al., 2006).

The rest of this paper is organized as follows. The next section gives the whole picture of the language service ontology in a service-oriented infrastructure, and gives a brief description on the configuration of the lexicon access function class. Section 3 introduces the *ontologization* of LMF, while Section 4 discusses how a service-oriented lexicon taxonomy can be grounded to the ontologized LMF. Section 5 then discusses the ontologization of lexicon access functions, and presents a possible solution to relate an actual Web service API with the ontological description. Finally, next two sections summarize the related work and our future work.

2. Language Service Ontology

Recently, several activities and projects have been reported to realize Web-based language infrastructures; these include a service-oriented infrastructure such as the Language Grid [1] (Ishida, 2006), as well as the research infrastructure such represented by the pan-European effort called CLARIN[2] (Calzolari, 2008). These two types of infrastructures share issues of

[1] http://langrid.nict.go.jp
[2] http://www.clarin.eu

interoperability and reusability of language data resources and NLP tools/systems, even though their primary objectives are totally different.

In principle, most of the existing language data resources and NLP tools/systems have been created independently, resulting in a situation where data format, annotation scheme, access method and other features are all idiosyncratic. To address this issue, standardization is inevitable: standardized APIs are necessary for NLP tools/systems; standardized data semantics as well as data format are required for language data resources. In addition and importantly, these standards should be designed based on a comprehensive shared ontology which covers all possible elements of a language infrastructure. In this sense, a language infrastructure, particularly that in an open environment, should be ontology-based.

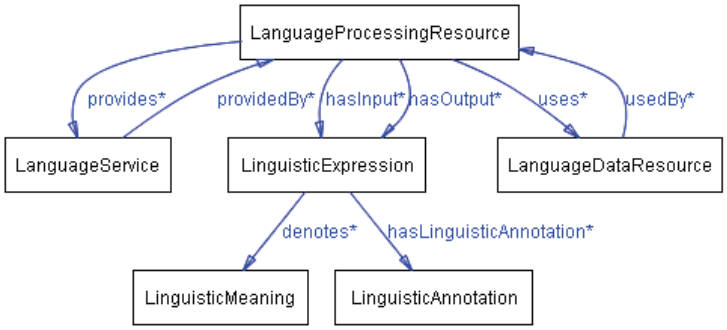

Figure 1: Top-level of the Language Service Ontology

Figure 1[3] illustrates the top-level of the language service ontology that was proposed in (Hayashi et al., 2008) with respect to the Language Grid infrastructure. Each box in the figure denotes a top-level class or concept in the ontology. As depicted in the figure, **`LanguageService`** is *`providedBy`* **`LanguageProcessingResource`** which takes **`LinguisticExpression`** as *`input/output`*, and *`uses`* **`LanguageDataResource`**. Note that, in a service-oriented point view, any data resource has to be coupled with a kind of processing resource which provides an access function. Note also, that **`LinguisticExpression`** *`denotes`* **`LinguisticMeaning`**, and can have multiple **`LinguisticAnnotation`**.

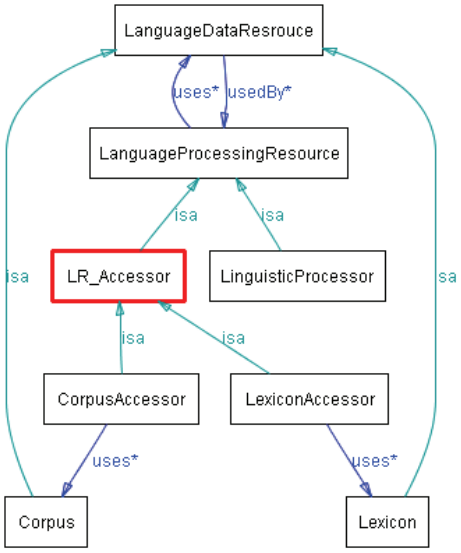

Figure 2: Configuration of Language Data Resource and the Accessor

Each top-level class in the figure is defined in further detail as a sub-ontology. Figure 2 develops one-step further the language data resource class (into **`Corpus`** and **`Lexicon`** classes), and the corresponding language processing resource class. As shown in the figure, the language processing resource is divided into **`LinguisticProcessor`**, which represents usual NLP tools/systems, and **`LR_Accessor`**, which is further divided into two sub-classes according to type of the language data resources that are used; **`LexiconAccessor`** targets **`Lexicon`**, while **`CorpusAccessor`** accesses to **`Corpus`**. The rest of the paper discusses **`Lexicon`** and the corresponding **`LexiconAccessor`** sub-ontologies.

3. Ontologization of LMF

In the language service ontology, the sub-ontology or taxonomy of lexicons may be defined based on a service-oriented perspective, resulting in a situation where the taxonomy may not be linguistically or lexicographically motivated. However it would be far better to ground the service-oriented taxonomy to some lexicon ontology that is based on shared linguistic and lexicological principles.

We have employed LMF (Lexical Markup Framework) (Francopoulo et al., 2006a) as such a framework. As known, LMF, worked out by the ISO TC37/SC4 community, is in the final stage of the international standardization process. The specification of LMF (ISO24613, 2008) states that *the ultimate goal of LMF is to create a modular structure that will facilitate true content interoperability across all aspects of electronic lexical resources*. Given this goal, the proposed modular structure of LMF consists of a core package and a number of extensions for modeling a range of lexicons including machine readable dictionaries (MRDs) and NLP lexicons. These LMF extensions are presented by extending the LMF core package, encouraging us to ontologize them by organizing the classes defined in the core package as subclasses of the top LMF class. Here *to ontologize*

[3] All the diagrams have been produced with OntoViz plugin of Protégé ontology editor.

simply means to give a corresponding OWL representation to the constructs in the framework.

Figure 3 illustrates the ontological configuration for the LMF core model. Although the specifications of LMF are given by using UML (Unified Modeling Language) diagrams, we can convert these diagrams relatively straight forward on OWL by applying some conversion conventions; for example, we have converted the *aggregation* in UML into `hasXXX` property. As stated in the figure, we have defined the LMF core package as an independent sub-ontology; the namespace **`lmfcore`** prefixed to the entities indicates this situation. All the other extensions are defined in another sub-ontology which imports the LMF core ontology; the namespace **`lmfall`** represents the whole sub-ontology. This account is somehow different from the original LMF specification, where, managing all types of lexicon in a single ontological space is not considered. However this account gives us an opportunity to stipulate a range of lexical resources in a unique ontological space, and this is mandatory in a service-oriented language infrastructure.

Figure 4 shows a part of the LMF NLP Semantics extension, which is associated in particular with the lexical semantic notions of the extension. Note that this extension has been defined by sub-classing the classes in the LMF core package. The point is certain sub-class of the lexicon class is defined so as to have a particular type of the lexical entry. For example, in Figure 4, **`lmf.Sem.Lexicon`**, as a sub-class of **`lmfcore:Lexicon`**, is defined as having **`lmf.Sem.LexicalEntry`** that is, in turn, a sub-class of **`lmfcore:LexicalEntry`**. Again, this account is somehow different from the original LMF specification, where, for example, sub-classing of the lexicon class is not allowed.

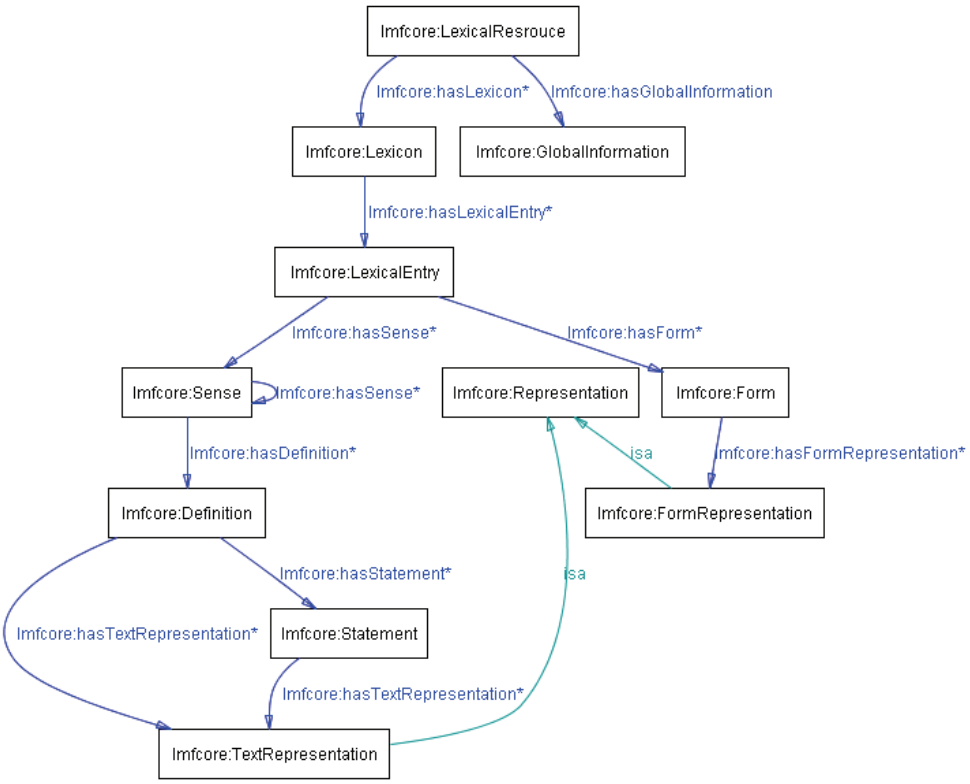

Figure 3: Configuration for LMF core package

4. LMF-based Lexicon Taxonomy

Shown in Figure 5 is an extremely simplified view of the lexicon taxonomy, which is presented just to show the notion of service-oriented lexicon taxonomy; where the top-level class **`Lexicon`** is first divided into **`DictionaryForHumanUse`** and **`LexiconForNLP`**. The former includes a class for so-called machine readable dictionaries (**`MRD`**), which is further divided into **`MonolingualDictionary`** and **`BilingualDictionary`**. The latter, on the other hand, derives a class for computational concept lexicon (**`ConceptLexicon`**), which has been introduced in order to stipulate WordNet-type lexical resources.

As seen in this figure, the configuration of the service-oriented lexicon taxonomy can be quite arbitrary, rather than linguistically or lexicographically motivated. However, once we have ontologized the necessary parts of the LMF, we can ground the service-oriented lexicon taxonomy to the ontologized LMF. Figure 6 depicts the basic notion of the grounding; it states that each of the classes in the service-oriented taxonomy is defined in terms of lexical entry type that they accommodate, and the lexical entry types are defined in the ontologized LMF. For example, **`BuilngualDictionary`**, a sub-class of **`MRD`**, is defined by **`hasLexicalEntry`** property whose range is strictly restricted to **`BilingualLexicalEntry`**, which, in turn, is one of the descendant class of the **`lmfcore:LexicalEntry`** in the ontologized LMF.

When we are to represent and incorporate some new type of lexicon, we should first introduce a new lexical entry sub-class for the target lexicon in the service-oriented taxonomy, and then appropriately relate it to somewhere in the lexical entry taxonomy of the ontologized LMF.

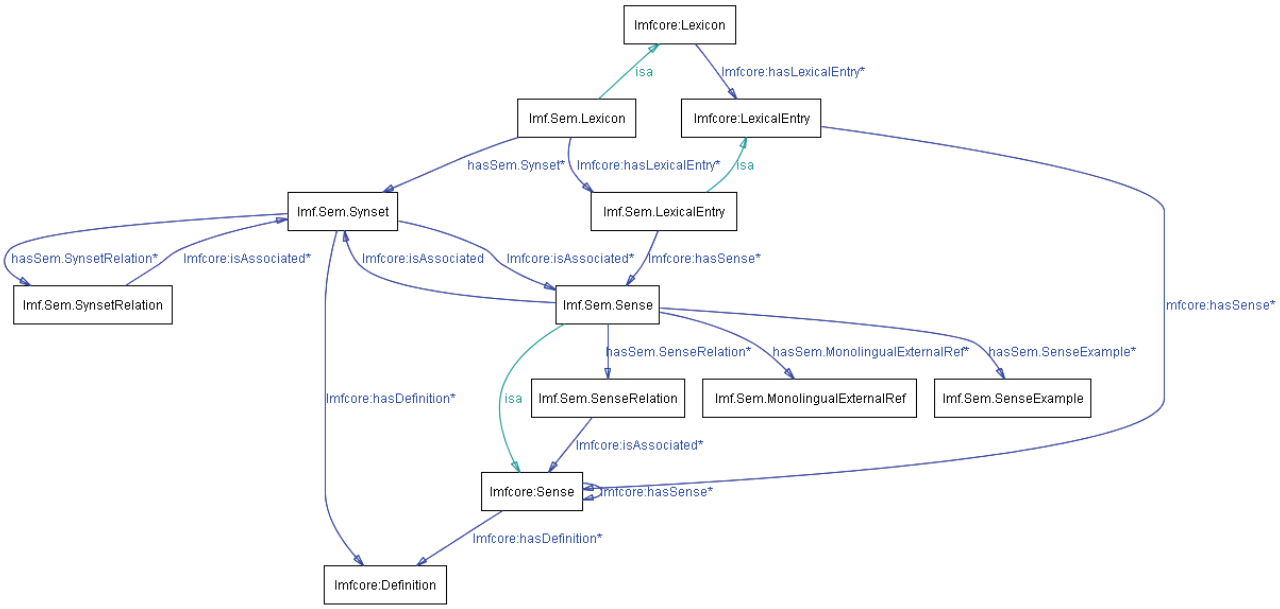

Figure 4: Configuration for LMF NLP Semantic Extension

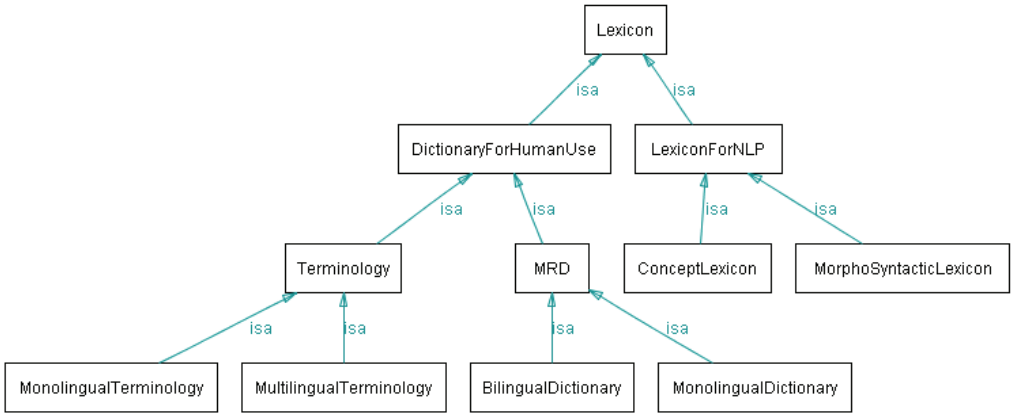

Figure 5: Service-oriented Lexicon Taxonomy: A partial and simplified view

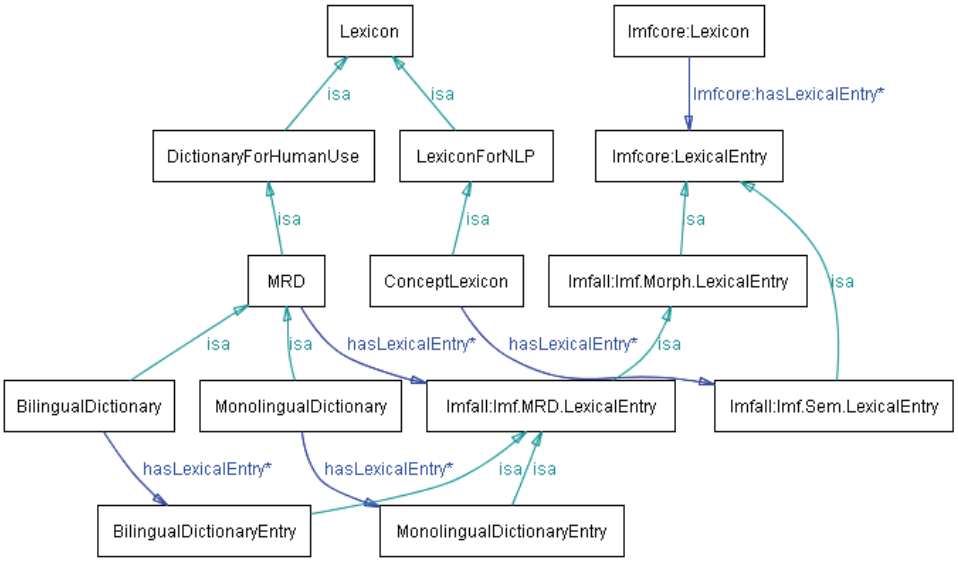

Figure 6: Grounding the Service-oriented Taxonomy to the Onotologized LMF

5. Ontologization of Lexicon Access Functions

As already shown in Figure 2, a range of lexicon access functions are grouped into **`LexiconAccessor`** class in the language service ontology. This class is a derived from the **`LR_Accessor`** class, which in turn is a sub-class of the **`LanguageProcessingResource`** class that can provide a language service. As a language processing resource is stipulated by the input/output data types and the language data resource which accesses to, the sub-ontology for lexicon access functions should also be configured with this principle.

Figure 7 summarizes the ontological configuration around the lexicon accessor class; its input is restricted to **`LexiconAccessQuery`**, which is a sub-class of the **`LinguisticExpressoin`**; whereas the output is restricted to **`LexiconAccessResult`**, which is also a sub-class of the linguistic expression class.

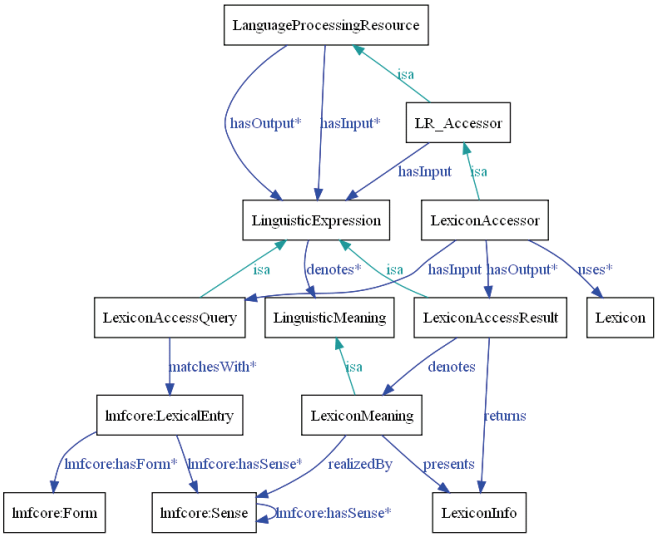

Figure 7: Configuration for Lexicon Accessor

A query to a lexicon access function, in general, consists of two types of information: matching specification and output specification. The core of the matching specification is, by necessity, a key string whose main part is a linguistic expression, typically a word, that could be further modified by some query syntax such as the regular expression. The allowed variety of key string patterns depends on the search methods that are provided by a lexicon access function. Therefore it is reasonable to sub-class the lexicon access query as a sub-class of the linguistic expression class which consists of the key string and the query conditions.

The lexicon access result, on the other hand, denotes the essential information encoded in the matched lexicon entries. From the language service ontological viewpoint, the essential information encoded in a lexical entry is represented by the **`LexiconMeaning`** class, which is a sub-class of **`LinguisticMeaning`**. Notice that this account perfectly accords with LMF; in LMF, even in the MRD extension, the essential information, such as translation equivalents in a bilingual dictionary, is modeled as having a relation with the sense class, insisting that such information should capture some semantic aspects of the lexical entry. In Figure 7, notice that the linking between the lexicon meaning (peculiar in the service ontology world) and the sense part of a lexical entry (in LMF world) is adequately represented by the *`realizedBy`* property.

Figure 8 illustrates an instance level example of English-to-Japanese bilingual dictionary access. Here the query is an English word "bank" which has a number of translational equivalents depending on the senses. In the figure however, only two of them are developed; *financial bank* sense gives Japanese equivalent 銀行(ginkou) and バンク(bank), while *slope bank* sense is represented by 土手 (dote) and 堤 (tsutsumi). The *`matchesWith`* property links an instance of lexicon access query class with an instance of lexical entry class, capturing the relationship held between the input to the lexicon access function and the corresponding lexical entries. The relationship however is only coarsely captured, and some deep constraints associated with the given query specification is not explicitly encoded; it is beyond the scope of current ontologization.

The ontologization of lexicon access functions depicted in Figure 7 is conceptual, and there exists a substantial gap between the conceptual picture and actual Web API specifications. To fill the gap, we try to apply a newly presented W3C recommendation SAWSDL (Semantic Annotations for WSDL) (Verma and Sheth, 2007; SAWSDL, 2007). Although some author (Yu, 2007) describes SAWSDL as a lightweight approach to *Semantic Web Services*, it may provide a simple yet reasonably powerful device. With the **`sawsdl:modelReference`** construct provided by SAWSDL, we can semantically annotate a WSDL document by making references to the concepts in the domain ontology. Figure 9 exemplifies the semantic annotations applied to a WSDL document of the bilingual dictionary access[4]; whose inputs are source language, target language, key string, and the matching specification, whereas the return data is a set of translations slightly structured according to the senses.

With the current SAWSDL, we can anchor the input/output data types, as well as the service category only to some relevant concepts in the domain ontology; we cannot relate such an item to an ontological concept with some ontological *path*. This sometimes gives us a difficulty in appropriately specifying ontological anchors. For example, we can only specify **`lmfall:TextRepresentation`** class as the ontological reference to a translation equivalent; we, however in this case, need to specify that the instance of the text representation class is particularly the one linked from an instance of the **`MRD.Equivalent`** class. To remedy this problem, we have reluctantly introduced the **`LexiconInfo`** class to directly represent actual data structure returned by a lexicon access service. In Figure 9, we attached a reference to **`SimpleBilingualDictionaryInfo`** class for the output data whose data type is **`SearchResults`**.

[4] The associated class in the service ontology would be **`SimpleBilingualDictionaryAccessor`**, which is a sub-class of the **`LexiconAccessor`** class.

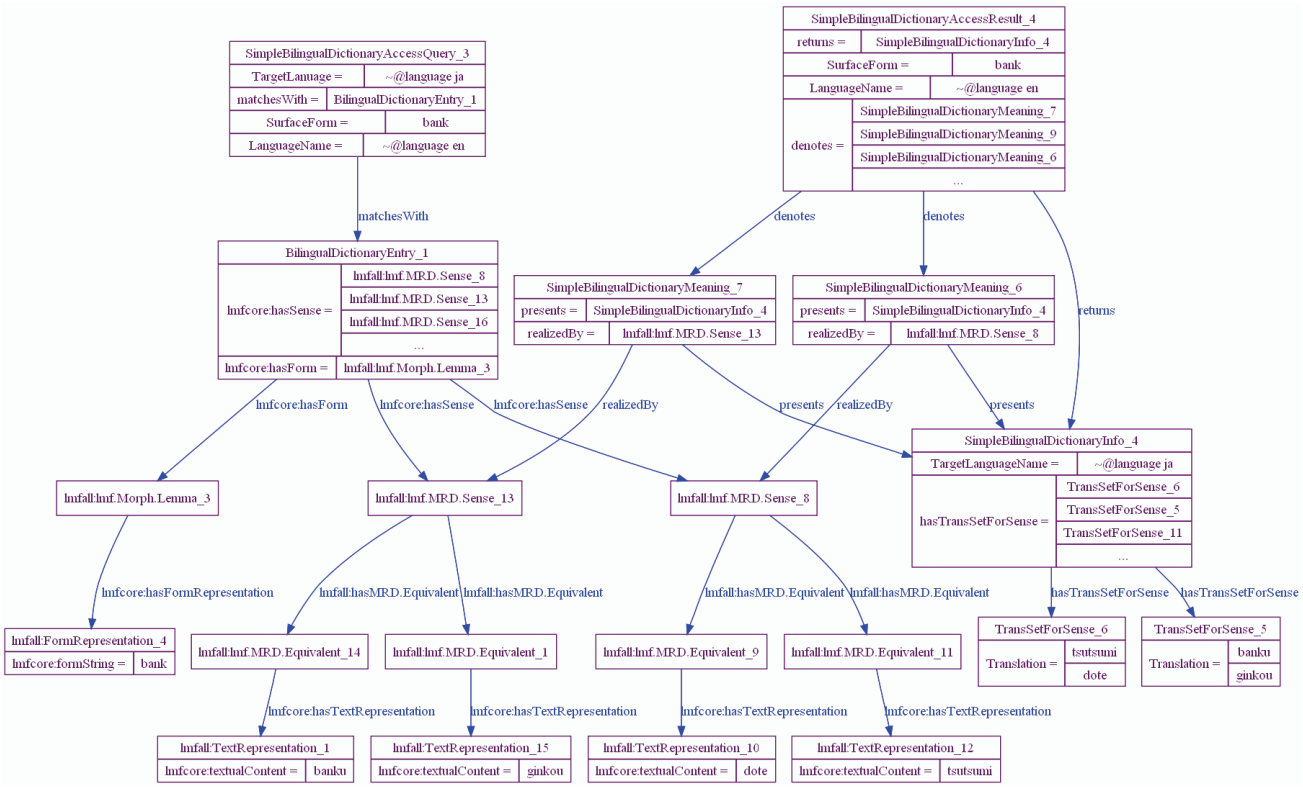

Figure 8: An Instance-level Example of Bilingual Dictionary Access (partial)

```
.....
  <wsdl:portType name="SimpleBilingualDictionary">
   <wsdl:operation name="serachSimpleBilingualDictionary"
        sawsdl:modelReference="http://langrid.nict.go.jp/lso/lso#SimpleBilingualDictionaryAccessor">
     <wsdl:input message="sbd:serachSimpleBilingualDictionaryRequest"/>
     <wsdl:output message="sbd:serachSimpleBilingualDictionaryResponse"/>
   </wsdl:operation>
.....
     <xsd:element name="serachSimpleBilingualDictionaryRequest"
        type="sbd:SearchQuery">
     </xsd:element>
     <xsd:element name="serachSimpleBilingualDictionaryResponse"
        type="sbd:SearchResults">
     </xsd:element>
.....
     <xsd:complexType name="SearchQuery"
        sawsdl:modelReference="http://langrid.nict.go.jp/lso/lso#SimpleBilingualLexiconAccessQuery">
        <xsd:sequence>
                <xsd:element name="SurfaceForm" type="xsd:string"></xsd:element>
                <xsd:element name="LanguageName" type="xsd:language"></xsd:element>
                <xsd:element name="TargetLanguage" type="xsd:language"></xsd:element>
        </xsd:sequence>
     </xsd:complexType>
.....
     <xsd:complexType name="SearchResults">
        sawsdl:modelReference="http://langrid.nict.go.jp/lso/lso#SimpleBilingualDictionaryInfo">
        <xsd:sequence>
                <xsd:element name="Language" type="xsd:language"></xsd:element>
                <xsd:element name="TransSetForSense" type="sbd:TransSetForSenseType"
                                maxOccurs="unbounded" minOccurs="1"></xsd:element>
        </xsd:sequence>
     </xsd:complexType>
     <xsd:complexType name="TransSetForSenseType">
        <xsd:sequence>
                <xsd:element name="Translation" type="xsd:string" maxOccurs="unbounded" minOccurs="1"></xsd:element>
        </xsd:sequence>
     </xsd:complexType>
.....
```

Figure 9: Fragments of the SAWSDL document for Simple Bilingual Dictionary Accessor

6. Related Work

Needless to say, LMF, which will soon be an ISO standard, is a mature framework for modeling a wide range of lexicons. It has been designed based on a number of standardization efforts including EAGLES, PAROLE, SIMPE, and MILE (Francopoulo et al. 2006b). *Ontologization*, in a broader sense, of LMF is being also discussed within the ISO TC37 community.

As for the application of LMF to a particular domain, (Quochi et al., 2007) describes a large-scale lexical resource (BioLexicon) for the biology domain, which has successfully achieved semantic interoperability and extendability by adopting ISO standards including LMF. The paper also describes a procedure for deploying "LMF lexical Web services" which integrate automatically uploaded lexical data provided by different sources and groups.

Standardization of lexicon access functions, on the other hand, has not yet been fully discussed. Although, a few relevant attempts have been visible, including DICT (the Dictionary Server Protocol) (Faith and Martin, 1997), and JADT (the Dictionary and Thesaurus API for Java) (Midha, 2004), standardized Web APIs for a range of lexicon accesses have not been published.

7. Conclusion and Future Work

This paper discussed an ontologization of access functions for a range of lexicons, particularly machine readable dictionaries and computational concept lexicons, in the context of a service-oriented language infrastructure, such as the Language Grid. The presented ontologization is based on a service-oriented taxonomy of lexicons which is grounded to a principled LMF-based lexicon ontology. For this purpose, we have proposed an ontologization of LMF. This paper also discussed a possible solution to interrelate an actual Web API of a lexicon access function with the relevant ontological descriptions by adopting a newly published W3C recommendation SAWSDL.

For the future work, we will refine both of the lexicon taxonomies: the service-oriented taxonomy and the ontologized LMF. To do this, we will be looking at a number of concrete lexicon/dictionary resources and dictionary services. For a research issue, we will consider composite lexicons access functions, which make accesses to more than one lexicons, and involves some information integration/aligning processes. In this regard, we need to seek a mechanism to (semi-)automatically configure necessary portion of ontological descriptions for the composite/combined lexicon access function from the descriptions of the atomic elements. One of the key issues here would be modeling of the virtual lexical entry type that accommodates information coming from different types of lexicons.

References

Nicoletta Calzolari. (2008). Approaches towards a "Lexical Web": the Role of Interoperability. In: Proc. of ICGL2008, pp.34-42. (Invited presentation)

Rickard E. Faith, Bret Martin. (1997). RFC2226: A Dictionary Server Protocol. The Internet Society.

Gil Francopoulo, Monte George, Nicoletta Calzolari, Monica Monachini, Nuria Bel, Mandy Pet, and Claudia Soria. (2006a). LMF for Multilingual, Specialized Lexicons. In: Proc. of LREC2006, pp.233-236.

Gil Francopoulo, Monte George, Nicoletta Calzolari, Monica Monachini, Nuria Bel, Mandy Pet, and Claudia Soria. (2006b). LMF for Multilingual, Specialized Lexicons. In: Proc. of LREC2006 Workshop on Acquiring and Representing Multilingual, Specialized Lexicons.

Yoshihiko Hayashi, Thierry Declerck, Paul Buitelaar, and Monica Monachini. (2008). Ontologies for a Global Language Infrastructure. In: Proc. of ICGL2008, pp.105-112.

Toru Ishida. (2006). Language Grid: An Infrastructure for Intercultural Collaboration. In: Proc. of SAINT2006, pp.96-100. (Invited presentation)

ISO 24613. (2008). Lexical Markup Framework (LMF) revision 16. ISO FDIS 24613:2008.

Rakesh Midha. (2004). Dictionary and Thesaurus API for Java. IBM alphaWorks. http://www.alphaworks.ibm.com/tech/jadt

Valeria Quochi, Riccardo Del Gratta, Eva Sassolini, Monica Monachini, and Nicoletta Calzolari. (2007). Toward a Standard Lexical Resource in the Bio Domain. In: Proc. of the 3rd Language & Technology Conference, pp.295-299.

SWSDL 2.0. (2007). Semantic Annotations for WSDL and XML Schema. http://www.w3.org/TR/sawsdl/

Kunal Verma, Amit Sheth. (2007). Semantically Annotating a Web Service. IEEE Internet Computing, Vol.11, No.2, pp.83-85.

WSDL 2.0. (2007). Web Services Description Language (WSDL) Version 2.0. http://www.w3.org/TR/wsdl20

Liyang Yu. (2007). Introduction to the Semantic Web and Semantic Web Services. Chapman & Hall/CRC, 341 pages.