

Designing Metadata with Existing Ontologies

Supervisor Professor Toru Ishida

Department of Social Informatics
Graduate School of Informatics
Kyoto University

Heeryon CHO

February 9, 2005

Designing Metadata with Existing Ontologies

Heeryon CHO

Abstract

Semantic Web is expected to become an enhanced information foundation for both humans and machines. Once data on the WWW is annotated with machine understandable metadata linked to ontology, complex tasks can be automated using software agents. Among the issues surrounding the Semantic Web, difficulties in designing metadata act as a bottleneck to widespread adoption of the Semantic Web. This is due to the restrictions inherent in the metadata design task: Both humans and machines can create metadata, but in the case of the former, domain knowledge and design skill is required. In the case of the latter, metadata with only limited expressivity can be created.

To tackle the problem of designing metadata, many existing researches propose design methodologies that focus on designing new metadata from scratch. Although most of them agree on the importance of reusing existing ontologies, few give detailed solutions. This research takes up this less-tackled problem of reusing existing ontologies. The goal is to propose a metadata design method that utilizes existing ontologies as design material, and to specify what information, if provided in each design step, will be helpful to metadata designer in making design decisions.

To achieve this, 1) existing ontologies on the WWW should be analyzed. This is necessary because these ontologies later constitute the design material. Unless ontologies on the WWW are abundant in variety and in quantity, referring to them would not be useful. Once the existence of various ontologies is confirmed, 2) problems in metadata design process need to be clarified. In particular, information needs that arise in a restricted setting, which prohibits human designer from referring to existing ontologies, should be investigated to highlight what information support may be useful. Based on the information needs obtained, 3) a step by step metadata design process which incorporates existing ontologies should be modeled, and design material metadata designer could refer to should be proposed.

Firstly, to see what ontologies actually exist on the WWW, ten ontologies containing Person Class were analyzed with a focus on person related class definitions, property definitions and class hierarchies. Secondly, to clarify what problems arise in the design process, metadata design experiment was conducted by the author. As data, a social networking service's user input form was selected. A restriction was placed on the author by refraining from referring to existing ontologies. This was done to highlight information needs that arise during the design process. Lastly, based on the information needs observed in the experiment, metadata design process that utilizes existing ontologies was modeled, and detailed design material which may aid the designer's decision making was proposed. Also, a prototype tool that generates design material useful to the designer is implemented.

1) Analysis showed that depictions of person varied according to ontologies. Even though many shared similar person properties like name and email, different properties (e.g. id, has_employees, knows), superclasses (e.g. Agent, Persistent_Item, SpatialThing), and subclasses (e.g. Employee, Faculty, Nurse) appeared in different ontologies.

2) Experimental result revealed that following problems occurred during metadata design process with limited information: Firstly, determining whether an item is a class or a property was difficult. Secondly, defining relationship between classes was difficult. This includes subsumption relations between classes, and class to class relationships which can be defined using object property definition. Lastly, defining all major properties of a given class was difficult.

3) Based on the experimental findings, the author proposed a metadata design process that incorporates existing ontologies. This process introduces the "RIDE" loop, where human designer refers to the existing ontologies for design hints and ideas. As for the design material, for class/property uncertainty problem, both class and property examples are listed to display which is the majority definition. For defining class relationship problem, existing relationships are displayed. For comprehensive property coverage problem, properties in multiples ontologies are displayed in table format.

既存オントロジを用いたメタデータ設計支援

曹 喜蓮

内容梗概

セマンティック Web は、人間はもちろん、機械の理解をも前提とした双方の次世代情報基盤として注目を浴びている。本格的なセマンティック Web が実現されると、機械が理解可能なメタデータを WWW(World Wide Web)上のデータに付加することで、複雑な処理が自動化できるようになる。しかし、そのためには様々な問題を解決しなければならず、その中でもメタデータの設計問題をまず克服しなければならない。

メタデータは、人間と機械の両者によって作成できるが、前者の場合、設計者がドメイン知識と設計能力を持っていないといけないという制約があり、後者の場合、限られた種類のメタデータしか作成できないという制約がある。このため、専門能力を持たない人間がメタデータを設計することは難しく、これがセマンティック Web 普及の妨げとなっている。

メタデータ設計に関するこれまでの研究は、一からのメタデータ設計を取り上げたものが多く、オントロジの再利用を詳しく扱ったものは少ない。本研究は、オントロジの再利用に焦点を合わせ、既存のオントロジを、設計者が設計に利用するとき、どのように利用すれば設計に役立たせられるかを解明することを目的とする。

このためにはまず 1) 既存のオントロジにはどのようなものがあり、その特徴が何かを調べる必要がある。すでに WWW には千個以上ものオントロジがあり、その数はますます増えると予想される。本研究はメタデータが普及した近未来のメタデータ設計支援を見据えているため、オントロジの現状調査は、長期的に有効な設計支援法の確立のためにも重要である。次に 2) 人間設計者によるメタデータ設計の問題点を明確にする必要がある。特に、オントロジ再利用によるメタデータ設計を念頭に入れたメタデータ設計の問題点を洗い出す必要がある。これは、既存の多くの研究が一からのメタデータを対象とし、再利用をあまり扱っていないからである。最後に 1 と 2 に基づき 3) 既存オントロジを用いたメタデータ設計支援ツールを提案する必要がある。

そこで本研究は、WWW オントロジの現状調査として、人クラス(Person Class)を含む十個のオントロジを集め、人クラスを中心にクラス・プロパティの定義とクラス階層の分析を行う。次に、メタデータ設計における困難を洗い

出すために、試験的なメタデータ設計実験を行う。実験には著者が自ら参加し、あるソーシャルネットワーキングサイトのユーザ入力フォームからメタデータを設計するというタスクを行う。なお、このとき、他のオントロジを参照することを禁ずる制約を任意に設けることで、設計過程で生じる情報要求を浮き彫りにする。最後に、既存オントロジを用いた具体的なメタデータ設計プロセスと、各設計プロセスにおいて設計者の設計作業を支援するためには既存オントロジをどう活用すればいいかを考察し、メタデータ設計支援ツールを提案する。

結果、1) WWWには対象ドメインを反映した様々なオントロジがあることが分かった。人のメタデータを含む十個のオントロジを分析したところ、「従業員(Employee)」、「～の上司である(isBossOf)」、「肩書きを持っている(hasPosition)」などのメタデータを含むオントロジは会社や組織を、「患者(Patient)」、「識別番号(id)」、「健康保険(insurance.plan)」などを含むオントロジは病院を、「教職員(Faculty)」、「学位(degree)」、「アドバイザー(advisor)」などを含むオントロジは学校を、対象ドメインとして写していることが分かった。特に、このような対象ドメインの描写は、サブクラスレベルのクラスやプロパティの定義でより具体的に表れることが分かった。

2) 著者によるメタデータ設計実験を通して、いくつかのメタデータ設計時の困難を明らかにした。設計時における問題点として、以下のような困難があることが分かった。

- 設計対象がクラスであるかプロパティであるかを判定する判定問題
- クラス間の包摂関係を定義するクラス階層の定義問題
- オブジェクトプロパティを用いたクラス間の関係定義の問題
- クラスの主要プロパティの総括的な定義問題

3) 上の問題点に基づき、その解法となるような設計支援ツールを提案した。このツールは、設計者の入力を用いて関連する既存オントロジを収集し、次の方法で設計者のメタデータ設計を支援する。

- 設計対象を含むクラスとプロパティを抜き出し、どちらの定義がより多くの人によって支持されるかを提示する。
- 複数のオントロジからクラス階層だけを取り出し提示する。
- 複数のオントロジからオブジェクトプロパティだけを取り出し提示する。
- 複数のオントロジから設計対象の関連プロパティを取り出し、表形式に整理し、提示する。

Designing Metadata with Existing Ontologies

Contents

Chapter 1 Introduction	1
Chapter 2 Metadata and Ontology	5
2.1 Dublin Core	6
2.2 Ontology and Ontology Library	6
2.3 Applications	7
Chapter 3 Analysis of Existing Ontologies	8
3.1 Investigating Superclasses, Subclasses and Properties	9
3.2 Investigating Class Hierarchies	17
3.3 Remaining Five Ontologies	19
Chapter 4 Metadata Design Experiment	21
4.1 Objective	21
4.2 Data and Method	21
4.2.1 Target Document	21
4.2.2 What Information Is Given	22
4.2.3 Metadata Design Step	28
4.3 Result	28
4.3.1 Class/Property Uncertainty	29
4.3.2 Subsumption Relation Uncertainty	31
4.3.3 Class-to-Class Relation Uncertainty	32
4.3.4 Uncertainty of Comprehensive Metadata Coverage	34
Chapter 5 Ontology as Design Material	35
5.1 Metadata Design Process Focusing On Ontology Reuse	35
5.2 Prototype Tool	37
5.2.1 System Architecture	38
5.2.2 Sample data	39
5.2.3 Graphical User Interface	39
5.3 Design Materials	39

5.3.1 Majority Rule as Hints to Class/Property Decision	40
5.3.2 Seeing How Others Define Subsumption Relations	43
5.3.3 Seeing How Others Define Object Property.....	45
5.3.4 Comparing Properties Through Table	49
Chapter 6 Conclusion	51
Acknowledgments	53
References	54
Appendix:	A-1
A.1 Practical Ontologies Available on the WWW	A-1
A.2 Experimental Social Networking Ontology.....	A-7

Chapter 1 Introduction

Semantic Web is expected to become a foundation for automating complex processes by enabling machines to execute tasks like setting up schedules or searching for products that meet multiple criteria [1]. Execution of such tasks will be realized once great amount of metadata are annotated to data on the World Wide Web (WWW).

Metadata that constitute the basis of the Semantic Web can either be annotated automatically or semi-automatically by machines, or annotated manually by humans. Since annotation process requires much time and effort, many researches on metadata focus on automatic annotation techniques or annotation tools that support human manual annotation. But prior to the annotation process, a set of metadata describing the target domain must be created. Metadata creation process concentrates on designing appropriate metadata, a difficult task that often relies on the expertise of human metadata designer. Reusing existing ontologies should be considered before building one from scratch.

In order to reuse existing ontologies, however, relevant ontology must first be searched. Once it is found, it must be refined and extended to fit the target domain. Although many ontologies exist on the WWW, finding ontologies that precisely match users' needs is often difficult. It is more the case that portions of those ontologies are relevant.

This research investigates a method of utilizing existing ontologies on the WWW to support metadata design process. In particular, a metadata design process, which searches, organizes and displays relevant metadata contained in existing ontologies, is proposed. Detailed design steps and design material human metadata designer can refer to will be described, and a prototype tool that implements and outputs the above process and material is introduced.

Existing research that closely relates to this research is research on ontology development toolkit called SWOOPed [7]. SWOOPed also focuses on the problem of reusing existing ontologies, and as an answer to this, provides an easy ontology construction environment to enable human designer to directly

reuse metadata in existing ontologies. The difference of this research to SWOOPed is that this research aims at generating and displaying design material to the human designer rather than providing an editing environment. Moreover, organizing relevant metadata contained in multiple, not single, ontologies is the challenge that this research faces.

Another research that takes up the ontology reengineering problem is [15]. Here, a rough ontology generated from database schema is grounded to a foundational ontology to guarantee its consistency. This research, however, does not go as far as to deal with the matter of ontology consistency. It will remain focused on specific metadata handling.

Other metadata related researches deal with the evaluation, annotation, and construction of metadata from scratch. On metadata evaluation, two researches [30] [29] are well-known: the first proposes ontology design criteria, and the second proposes what design decisions are needed in ontology construction, and how to evaluate the created ontology.

Researches on metadata annotation deal with automatic annotation techniques and annotation support tools that support human manual annotation. These researches tackle the problem of actual annotation of metadata to large amount of data. Last but not least, research on ontology construction deals with methodologies of creating metadata from scratch. All of these researches, including this one, share the same ultimate goal of realizing the Semantic Web, but each tackle different issues pertaining to metadata. Some of these researches are introduced next to clarify the position of this research.

Automatic Annotation

Researches on automatic annotation use structural analysis, natural language processing, and other techniques to annotate large amount of data. For example, [34] automatically annotates content-rich HTML documents such as news articles in news portals through analyzing the document's structure. In [35], named entity recognition is used to automatically annotate named entities in a text document. Knowledge bases and ontology are incorporated during the annotation process.

S-CREAM [36] leverages information extraction technology to enable semi-automatic annotation of metadata. However, the system must have manually created training data prior to the execution of semi-automatic annotation. PANKOW [37] picks up expressions like “is-a” and “such-a” in WWW documents to statistically determine what an instance or named entity’s concept is. Since PANKOW uses data on WWW as reference, training data need not be created in advance.

The advantage of automatic or semi-automatic annotation by machine is that large amount of metadata can be annotated with little effort. As a result, the cost of metadata annotation is greatly reduced. However, at present, automatic annotation techniques can be applied to limited kinds of data. In the case of [34] [35], target data that can be handled are news items and named entities respectively. Therefore, complex tasks that can be processed using automatically annotated metadata may be limited at the moment.

Annotation Support Tools

Research on annotation support tools aims at supporting human annotation process. These tools try to reduce both the difficulty of the manual annotation process and the error in the resulting annotation.

SMORE seamlessly integrates content creation and annotation, and also allows searching for existing ontologies in local databases or on the WWW [38]. The search capability of the tool enables users to filter specific classes and properties in an existing ontology. Ont-O-Mat is similar to SMORE in that it allows content creation and annotation to be performed together [40].

MnM provides a generic process model of annotation, which consists of five activities: browse, markup, learn, test and extract [39]. MnM allows user to search and select a relevant ontology to be used in annotation.

Annotea is not a tool for directly supporting metadata annotation, but it still supports human metadata handling by dealing with the problem of maintaining and providing metadata [41].

Ontology construction methodology

Researches on ontology construction were actively carried out in the nineties together with researches on enterprise modeling, fault recovery system,

military planning and others. Projects like Cyc, IDEF, Enterprise Ontology, TOVE, KACTUS, SENSUS, METHONTOLOGY have proposed various ontology construction methodologies [42]. Other than above, Ontolingua [43] also proposes ontology construction methodology.

In the next chapter, key terms for this research are defined and few real world applications leveraging the related technology will be introduced. Chapter 3 explains the result of the analysis performed on existing ontologies on the WWW. Chapter 4 describes metadata design experiment conducted by the author, and what difficulties were observed during the experiment. Based on the experimental findings, chapter 5 proposes a metadata design process and detailed design material that utilizes existing ontologies. Moreover, a prototype tool that implements the design process and outputs the design material is introduced. Lastly, conclusion and future work is discussed in chapter 6.

Chapter 2 Metadata and Ontology

Metadata is data about data. In particular, metadata is machine understandable information for the web [2]. A set of metadata that forms a semantic structure can be viewed as ontology. Ontology is an explicit specification of a conceptualization [3].

RDF (Resource Description Framework) [4] and OWL (Web Ontology Language) [5] are standard web-based frameworks for metadata and ontology respectively. RDF provides a framework for describing data about data. It describes metadata in triples, which consist of a subject, a predicate and an object. OWL is an ontology description language. Three sublanguages of OWL are currently provided: OWL Lite, OWL DL, and OWL Full. RDF and OWL are W3C (World Wide Web Consortium) recommendations.

Metadata is used on the WWW and several web applications already exist. Ontology has been used in expert systems and enterprise modeling in the past and application to the WWW is under way. Although standards for metadata and ontology are in place and the potential power of the Semantic Web has been publicized for some time, the adoption of metadata and ontology related technology remains slow.

One reason for this is major hurdle placed on metadata creation. Metadata can be created by both humans and machines, but in both cases intrinsic restrictions exist: In the case of the former, some domain knowledge and design skill is required. In the case of the latter, metadata with only limited expressivity can be created. These restrictions act as obstacles to widespread creation and usage of metadata.

Real world metadata and ontology examples somewhat reflect these difficulties in metadata creation. So far, there are not many well-known metadata standards. Compared to metadata, ontologies seem to be varied in content and abundant in quantity, but the number still remain in the thousands. Hereinafter, a well-known metadata standard and present state of ontologies and ontology library will be explored, and few examples of real-world applications will be introduced.

2.1 Dublin Core

Dublin Core Metadata is a well-known metadata standard which defines vocabularies for describing resources [6]. The current Dublin Core Element Set, Version 1.1 consists of fifteen metadata elements (title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage, rights). These elements seem on first glance simple and straightforward, but it took more than five years for the working group to come to an agreement with these fifteen elements. The first element issued was issued in August 1998. The current Version 1.1 was issued in December 2004.

2.2 Ontology and Ontology Library

Ontology has been studied for some time in the area of expert systems and knowledge bases. Several ontology construction methodologies have been proposed, and some were discussed in Chapter 1. With the advent of the WWW and recent release of OWL as W3C recommendation, application of ontology to information integration and inference on the WWW is actively being explored. Already many ontologies exist on the WWW albeit not all are written in OWL. Various practical ontologies and domain specific ontologies are publicly available on the WWW. A list of domain specific ontologies is given in Appendix A1.

As number of ontologies increase, several ontology libraries and search engines have begun to appear: SchemaWeb [8], Protégé OWL Library [9], KSL Interactive Ontology Server [10], and DAML Ontology Library [11] store total of some thousand ontologies in their repositories. Search engines that can search whole or parts (classes and properties) of ontology are also available [12] [13].

Although environment for utilizing existing ontologies has started to emerge, reusing ontology remains a difficult task: Searching through existing ontologies does not assure high precision, high recall search result. Manually investigating ontologies one by one would consume too much time and effort. Refining and extending relevant ontology to fit the target domain would still

require expertise on the part of human designer. There needs to be a way for human metadata designer to create metadata without so much skill and effort.

2.3 Applications

Regardless of difficulties in metadata creation, several WWW applications incorporating metadata are in use. First of these is FOAF. FOAF (Friend of a Friend) project [14] is an experimental project which aims at facilitating the creation and maintenance of social networks between people through the WWW. FOAF provides a set of metadata for describing people and connection between people using RDF/XML. FOAF Explorer [16] is a visualization tool that enables humans to browse virtual neighborhoods of friends.

The second is RSS. RSS (RDF Site Summary) [17] is an extensible metadata description and syndication format. An XML application, RSS conforms to the W3C's RDF specification and is extensible via XML-namespace and/or RDF based modularization [18]. RSS Reader is a content aggregating tool that collects information written in RSS. Two examples of RSS Readers are given in [19] and [20]. RSS Readers are typically used for gathering latest news items and website updates.

Third and fourth applications utilize user-created metadata, but the author is unsure whether these two implement standard Semantic Web technology like RDF and OWL. The third is in the form of social bookmark or social link management [21]. This social bookmark application allows multiple users to share their bookmark using self-defined metadata. And the last is an online photo sharing system [22]. In this photo application, users can add metadata that describe their photos when uploading the photos. Using user-added photo metadata, searching and maintaining photo become easy.

Certainly above are not an exhaustive list of real world applications, but there still are not many applications that leverage the power of metadata and ontology. This is in part due to the metadata creation bottleneck. Unless difficulties in metadata creation are circumvented, wide-spread adoption of metadata and ontology will not be achieved.

Chapter 3 Analysis of Existing Ontologies

As mentioned in the previous chapter, there are many ontologies now available on the WWW. According to multiple searches conducted over the WWW using the file type extension option (e.g. filetype:rdf) of the search engine, the existence of more than one thousand ontologies was confirmed. These ontologies are written in various ontology description languages including DAML (DARPA Agent Markup Language), DAML+OIL (Ontology Inference Layer), RDFS (RDF Schema), and OWL (Web Ontology Language). By checking the creation date of these ontologies, it was confirmed that newer ontologies tend to be written in OWL than other languages. Moreover, searches performed over a period of time have verified that the number of ontologies is increasing.

Understanding the overall feature of the ontologies on the WWW is a prerequisite for this research since this research seeks to utilize existing ontologies as design material to support metadata design process. Unless ontologies on the WWW are abundant in variety and in quantity, design material generated from these ontologies would not be useful.

In this section, the contents of several actual ontologies are analyzed to grasp the characteristic of existing ontologies. Ten ontologies containing Person Class are selected as sample ontology to be analyzed. Setting the Person Class as pivot, superclasses, subclasses, and properties of the Person Class are analyzed. Although the actual analysis was conducted over the ten ontologies, the author further selects five to be explained, in order to keep the explanation succinct and to the point. The URLs of the ten ontologies along with the summarized tables of the remaining five ontologies is provided at the end of this Chapter, i.e. Section 3.3.

Based on the analysis, the following conclusions are drawn:

- Understanding of the target domain is gained at the subclass level.
- Two kinds of class hierarchy patterns exist: one incorporating outside classes (open) and the other using classes in the same ontology (closed).

3.1 Investigating Superclasses, Subclasses and Properties

Tables 3.1 through 3.5 each show an organized display of superclasses, subclasses, and properties of the Person Class defined in each ontology. On the first row of each table, the URI of the ontology is given. Starting with the second row with the cell-shaded heading “CLASS | SUBCLASSOF” to the row just before the cell-shaded heading “PROPERTIES | DOMAIN | RANGE”, subclasses and superclasses of the Person Class are displayed. The two columns (“CLASS”, “SUBCLASSOF”) reflect the subsumption relation between two classes. This relation can be read in the table as “some class positioned on the right-hand column subsumes some class positioned on the left-hand column.” That is, the class on the right-hand (left-hand) column is the superclass (subclass) of the class on the left-hand (right-hand) column. Using the cell-shaded heading, one may also read “the left-hand column class is the SUBCLASSOF the right-hand column class.”

The “PROPERTIES | DOMAIN | RANGE” heading and below are Person Class-related properties. A property is determined as a property of a certain class on the basis of its domain. That is, if a property had Person Class as its domain, then that property is considered as Person Class property. In the tables, both the direct and indirect properties of the Person Class are displayed. Here, indirect properties denote superclass properties of the Person Class. Moreover, subclass properties of Person Class are also displayed. For example, in Table 3.1, properties of Alumni Class, which is the subclass of Person Class, are displayed. Finally for each property, the range is indicated if it is explicitly defined in the ontology.

Hence, the three columns below the “PROPERTIES | DOMAIN | RANGE” heading can be interpreted as “some property has some class as domain and some resource as range” or “some class has some property which has some range.” For example, the homepage property in the first column, tenth row of Table 3.1 can be interpreted as “homepage has Person Class as domain and <http://ainge.cs.uga.edu/gis/lstdis#Link> as range” or “Person Class has homepage property which has range <http://ainge.cs.uga.edu/gis/lstdis#Link>”. Hereinafter, each table will be explained in detail.

Table 3.1: Ontology covering the academic domain

URI	http://www.cs.uga.edu/~ch/GlobalInfoSys/project/aboutUs.rdfs	
CLASS	SUBCLASSOF	
Person	http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource	
Faculty	Person	
ResearchStudent	Person	
ResearchStaff	Person	
Alumni	Person	
PROPERTIES	DOMAIN	RANGE
name	Person	http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal
homepage	Person	http://ainge.cs.uga.edu/gis/lstdis#Link
affiliation	Alumni	http://ainge.cs.uga.edu/gis/lstdis#Link
degree	Alumni	http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal
advisor	Alumni	Faculty
thesis	Alumni	http://ainge.cs.uga.edu/gis/lstdis#Thesis
grad_date	Alumni	http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal
contact	Alumni	http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal

Table 3.1 shows Person Class-related classes and properties defined in “http://www.cs.uga.edu/~ch/GlobalInfoSys/project/aboutUs.rdfs.” Here, “http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource” is defined as superclass of Person Class. Faculty, ResearchStudent, ResearchStaff, and Alumni Class are defined as subclasses of Person Class. There are two direct properties of Person Class: name and homepage. Other properties like affiliation, degree, advisor, thesis, and grad_date, and contact are defined as properties of Alumni Class. Based on the kinds of metadata defined in this ontology, one can assume that this ontology covers the academic domain.

Notice that this understanding is derived at the subclass level. Classes like Faculty, ResearchStudent, ResearchStaff, and Alumni, and properties like degree, advisor, thesis, and grad_date are terms that represent the academic domain. “http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource,” superclass of Person Class, on the other hand, is too abstract to give any hints to where this ontology might be used.

Moving on to the next table, Table 3.2 shows Person Class-related metadata defined in “http://www.openhealth.org/ASTM/simplified-model.rdfs.” Patient and Provider Class are defined as subclasses of Person Class. Furthermore, Physician and Nurse Class are defined as subclasses of Provider Class. For property definition, person.name, address, and id are defined as Person Class properties, whereas insurance.plan is defined as Patient Class property.

Table 3.2: Ontology covering the healthcare domain

URI	http://www.openhealth.org/ASTM/simplified-model.rdfs	
CLASS	SUBCLASSOF	
Patient	Person	
Provider	Person	
Physician	Provider	
Nurse	Provider	
PROPERTIES	DOMAIN	RANGE
person.name	Person	Name
address	Person	Address
id	Person	Identifier
insurance.plan	Patient	Payor

Based on the metadata defined in Table 3.2, one may easily assume that this ontology covers the healthcare domain. Again, this understanding comes from metadata information provided at the subclass level. Person Class's direct subclass Patient and indirect subclasses Physician and Nurse all reflect the healthcare domain. In terms of property definition, Patient Class property, insurance.plan, gives some hint about the ontology usage.

Table 3.3: Ontology covering the organization domain

URI	http://www.daml.ri.cmu.edu/ont/homework/atlas-cmu.daml	
CLASS	SUBCLASSOF	
Employee	Person	
PROPERTIES	DOMAIN	RANGE
sex	Person	Sex
name_Person	Person	
has_employes	Person	Organization
email	Employee	EMail
job_title	Employee	
office	Employee	Office
start_date	Employee	http://www.daml.ri.cmu.edu/ont/homework/atlas-date.daml#Date
end_date	Employee	http://www.daml.ri.cmu.edu/ont/homework/atlas-date.daml#Date
employer	Employee	Organization
Expertise	Employee	
associated_with_project	Employee	Project
employment_categories	Employee	http://www.daml.ri.cmu.edu/ont/homework/atlas-employment_categories.daml#Employment_Categories

Table 3.3 shows Person Class-related metadata defined in the ontology "http://www.daml.ri.cmu.edu/ont/homework/atlas-cmu.daml." Here, Employee Class is defined as subclass of Person Class. Direct properties of Person Class are sex, name_Person, and has_employes (perhaps, has_employees is correct. It may be a typo in the ontology file). For Employee Class properties,

email, job_title, office, start_date, end_date, employer, expertise, associated_with_project, and employment_categories are defined. Again, the understanding of the domain becomes clearer at the subclass level. In this example, has_employes property of the Person Class may provide some hint about the ontology domain, but far more information is provided at the subclass level. Almost all Employee Class properties, with the exception of email which is frequently defined in many other ontologies, contain some notion related to organization.

So far we have seen that the understanding of the domain described in the ontology is derived at the subclass level. However, there may be cases where subclass level metadata are not defined in the ontology. In such cases, going down the class hierarchy to check subclass level metadata will return null result. The next two examples demonstrate this situation.

Table 3.4 shows Person Class-related metadata defined in the ontology “http://simile.mit.edu/repository/ontologies/official/cidoc_crm.rdfs.” This ontology describes the CIDOC Conceptual Reference Model (CRM), which provides definitions and a formal structure for describing concepts and relationships used in cultural heritage documentation [32]. Searching the Person Class class hierarchy returns eight superclasses, Biological_Object, Physical_Object, Physical_Stuff, Legal_Object, Stuff, Actor, Persistent_Item, and CRM_Entity, and no subclasses.

Table 3.4: CIDOC CRM cultural heritage ontology (continued->)

URI	http://simile.mit.edu/repository/ontologies/official/cidoc_crm.rdfs	
CLASS	SUBCLASSOF	
E21.Person	E20.Biological_Object	
E21.Person	E39.Actor	
E20.Biological_Object	E19.Physical_Object	
E19.Physical_Object	E18.Physical_Stuff	
E18.Physical_Stuff	E72.Legal_Object	
E72.Legal_Object	E70.Stuff	
E70.Stuff	E77.Persistent_Item	
E39.Actor	E77.Persistent_Item	
E77.Persistent_Item	E1.CRM_Entity	
PROPERTIES	DOMAIN	RANGE
P96B.gave_birth	E21.Person	E67.Birth
P97B.was_father_for	E21.Person	E67.Birth
P98.was_born	E21.Person	E67.Birth
P100B.died_in	E21.Person	E69.Death

Table 3.4: CIDOC CRM cultural heritage ontology (<-continued->)

PROPERTIES	DOMAIN	RANGE
P11B.participated_in	E39.Actor	E5.Event
P14B.performed	E39.Actor	E7.Activity
P22B.acquired_title_through	E39.Actor	E8.Acquisition_Event
P23B.surrendered_title_through	E39.Actor	E8.Acquisition_Event
P28B.surrendered_custody_through	E39.Actor	E10.Transfer_of_Custody
P29B.received_custody_through	E39.Actor	E10.Transfer_of_Custody
P49B.is_former_or_current_keeper_of	E39.Actor	E18.Physical_Stuff
P50B.is_current_keeper_of	E39.Actor	E18.Physical_Stuff
P51B.is_former_or_current_owner_of	E39.Actor	E18.Physical_Stuff
P52B.is_current_owner_of	E39.Actor	E18.Physical_Stuff
P74F.has_current_or_former_residence	E39.Actor	E53.Place
P75F.possesses	E39.Actor	E30.Right
P76F.has_contact_point	E39.Actor	E51.Contact_Point
P105B.has_right_on	E39.Actor	E72.Legal_Object
P107B.is_current_or_former_member_of	E39.Actor	E74.Group
P109B.is_current_or_former_curator_of	E39.Actor	E78.Collection
P131F.is_identified_by	E39.Actor	E82.Actor_Appellation
P8.witnessed	E19.Physical_Object	E4.Period
P25B.moved_by	E19.Physical_Object	E9.Move
P36B.was_registered_by	E19.Physical_Object	E15.Identifier_Assignment
P47F.is_identified_by	E19.Physical_Object	E42.Object_Identifier
P48F.has_preferred_identifier	E19.Physical_Object	E42.Object_Identifier
P54F.has_current_permanent_location	E19.Physical_Object	E53.Place
P55F.has_current_location	E19.Physical_Object	E53.Place
P56F.bears_feature	E19.Physical_Object	E26.Physical_Feature
P57F.has_number_of_parts	E19.Physical_Object	http://www.w3.org/2000/01/rdf-schema#Literal
P59F.has_section	E19.Physical_Object	E53.Place
P13B.was_destroyed_by	E18.Physical_Stuff	E6.Destruction
P24B.changed_ownership_through	E18.Physical_Stuff	E8.Acquisition_Event
P30B.custody_transferred_through	E18.Physical_Stuff	E10.Transfer_of_Custody
P34B.was_assessed_by	E18.Physical_Stuff	E14.Condition_Assessment
P44F.has_condition	E18.Physical_Stuff	E3.Condition_State
P45F.consists_of	E18.Physical_Stuff	E57.Material
P46B.forms_part_of	E18.Physical_Stuff	E18.Physical_Stuff
P46F.is_composed_of	E18.Physical_Stuff	E18.Physical_Stuff
P49F.has_former_or_current_keeper	E18.Physical_Stuff	E39.Actor
P50F.has_current_keeper	E18.Physical_Stuff	E39.Actor
P51F.has_former_or_current_owner	E18.Physical_Stuff	E39.Actor
P52F.has_current_owner	E18.Physical_Stuff	E39.Actor
P53F.has_former_or_current_location	E18.Physical_Stuff	E53.Place
P58F.has_section_definition	E18.Physical_Stuff	E46.Section_Definition
P111B.was_added_by	E18.Physical_Stuff	E79.Part_Addition
P113B.was_removed_by	E18.Physical_Stuff	E80.Part_Removal
P104F.is_subject_to	E72.Legal_Object	E30.Right
P105F.right_held_by	E72.Legal_Object	E39.Actor
P16B.was_used_for	E70.Stuff	E7.Activity
P39B.was_measured_by	E70.Stuff	E16.Measurement_Event
P43F.has_dimension	E70.Stuff	E54.Dimension
P101F.had_as_general_use	E70.Stuff	E55.Type
P130B.features_are_also_found_on	E70.Stuff	E70.Stuff
P130F.shows_features_of	E70.Stuff	E70.Stuff
P12B.was_present_at	E77.Persistent_Item	E5.Event
P92B.was_brought_into_existence_by	E77.Persistent_Item	E63.Beginning_of_Existence
P93B.was_taken_out_of_existence_by	E77.Persistent_Item	E64.End_of_Existence
P123B.resulted_from	E77.Persistent_Item	E81.Transformation
P124B.was_transformed_by	E77.Persistent_Item	E81.Transformation

Table 3.4: CIDOC CRM cultural heritage ontology (<-continued)

PROPERTIES	DOMAIN	RANGE
P1F.is_identified_by	E1.CRM_Entity	E41.Appellation
P2F.has_type	E1.CRM_Entity	E55.Type
P3F.has_note	E1.CRM_Entity	http://www.w3.org/2000/01/rdf-schema#Literal
P15B.influenced	E1.CRM_Entity	E7.Activity
P17B.motivated	E1.CRM_Entity	E7.Activity
P41B.was_classified_by	E1.CRM_Entity	E17.Type_Assignment
P62B.is_depicted_by	E1.CRM_Entity	E24.Physical_Man-Made_Stuff
P67B.is_referred_to_by	E1.CRM_Entity	E73.Information_Object
P70B.is_documented_in	E1.CRM_Entity	E31.Document
P129B.is_subject_of	E1.CRM_Entity	E73.Information_Object
P136B.supported_type_creation	E1.CRM_Entity	E83.Type_Creation
P137B.exemplifies	E1.CRM_Entity	E55.Type
P138B.has_representation	E1.CRM_Entity	E36.Visual_Item
P140B.was_attributed_by	E1.CRM_Entity	E13.Attribute_Assignment
P141B.was_assigned_by	E1.CRM_Entity	E13.Attribute_Assignment

Note that symbols placed in front of the class and property names, which consist of combination of uppercase alphabet letter(s) and number, are omitted here in the text for simplicity. Of the eight superclasses, *Biological_Object* and *Actor* are defined as direct superclasses of the *Person Class*. However, it is difficult to ascertain which domain this ontology covers only by looking at these superclasses. Even though several unique superclasses like *Persistent_Item*, *Physical_Stuff*, *Stuff* and *Legal_Object* are defined (is person a “persistent item?”), because superclasses express more abstract concept than their child class, trying to gain domain knowledge from superclass alone will fail. Turning to other metadata for hints, we look at *Person Class*–related properties. As direct properties of the *Person Class*, *gave_birth*, *was_born*, *was_father_for* and *died_in* are defined. At this stage, still little hint about the ontology usage is acquired. As we move on to superclass properties, properties like *is_current_keeper_of*, *right_held_by*, *is_current_owner_of*, *surrendered_custody_through*, *possesses* and *changed_ownership_through* indicate some kind of ownership. Based on these properties, one may surmise that this ontology covers some domain that deals with rights and ownership. Other superclass properties such as *bears_feature*, *has_number_of_parts*, *has_condition* and *is_composed_of* exhibit some means of expression that handles the outward appearance or composition of something. Moreover, this something may be assessed, as in *was_assessed_by*, or destroyed, as in

`was_destroyed_by`. The target domain is still not clear based on the superclass properties examined so far, but this much is known: that this ontology deals with ownership and rights, that some expression mechanism for depicting the outward appearance of some object is prepared, and that some object handled by this ontology may be assessed and destroyed. Compared to the understanding obtained at the subclass level (e.g. Table 3.1, 3.2, 3.3), the understanding gained at superclass level is abstract, and figuring out the target domain of the ontology using superclass level metadata may be difficult. However, several hints about the usage (to define rights and ownership, to express outward appearance, to assess and destroy something) are obtained by looking at the superclass properties.

Finally, Table 3.5 shows Person Class-related metadata defined in “<http://simile.mit.edu/repository/ontologies/official/foaf.owl>.” FOAF metadata is defined in this ontology. The explanation of FOAF is given in Chapter 2, Section 2.3. Here again, no subclasses of Person Class are defined. Meanwhile four direct superclasses are given, of which Agent Class is the only superclass sharing the same namespace as the Person Class. Three other direct superclasses, <http://www.w3.org/2000/10/swap/pim/contact#Person> (here on SWAP/Person), http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing (here on GEO/SpatialThing) and <http://xmlns.com/wordnet/1.6/Person> (here on WN/Person) are defined as class in some other ontologies.

Looking at the direct properties of the Person Class, we see basic person properties like `first_name`, `surname`, `family_name`, affiliation related properties like `workplaceHomepage`, `wrokInfoHomepage`, `schoolHomepage`, production related properties like `currentProject`, `pastProjectAs`, `plan`, `publications`, and miscellaneous properties like `myersBriggs` (metrics for human personality), `geekcode`, `img`, `knows`, `interest`, `topic_interest`. Among these, miscellaneous properties seem to best reflect the characteristic unique to this ontology.

As for superclass properties, various IDs (`jabberID`, `aimChatID`, `icqChatID`, `yahooChatID`, `msnChatID`) are defined as properties of Agent Class. No properties are defined for WORDNET/Person.

Table 3.5: FOAF (Friend of a Friend) Ontology

URI	http://simile.mit.edu/repository/ontologies/official/foaf.owl	
CLASS	SUBCLASSOF	
Person	Agent	
Person	http://www.w3.org/2000/10/swap/pim/contact#Person (SWAP/Person)	
Person	http://xmlns.com/wordnet/1.6/Person (WN/Person)	
Person	http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing (GEO/SpatialThing)	
SWAP/Person	http://www.w3.org/2000/10/swap/pim/contact#SocialEntity (SWAP/SocialEntity)	
PROPERTIES	DOMAIN	RANGE
geekcode	Person	http://www.w3.org/2000/01/rdf-schema#Literal
first_name	Person	http://www.w3.org/2000/01/rdf-schema#Literal
surname	Person	http://www.w3.org/2000/01/rdf-schema#Literal
family_name	Person	http://www.w3.org/2000/01/rdf-schema#Literal
plan	Person	http://www.w3.org/2000/01/rdf-schema#Literal
myersBriggs	Person	http://www.w3.org/2000/01/rdf-schema#Literal
img	Person	Image
knows	Person	Person
workplaceHomepage	Person	Document
wrokInfoHomepage	Person	Document
schoolHomepage	Person	Document
interest	Person	Document
topic_interest	Person	http://www.w3.org/2000/01/rdf-schema#Resource
publications	Person	http://www.w3.org/2000/01/rdf-schema#Resource
currentProject	Person	http://www.w3.org/2000/01/rdf-schema#Resource
pastProject	Person	http://www.w3.org/2000/01/rdf-schema#Resource
mbox	Agent	http://www.w3.org/2000/01/rdf-schema#Literal
mbox_sha1sum	Agent	http://www.w3.org/2000/01/rdf-schema#Literal
gender	Agent	http://www.w3.org/2000/01/rdf-schema#Literal
jabberID	Agent	http://www.w3.org/2000/01/rdf-schema#Literal
aimChatID	Agent	http://www.w3.org/2000/01/rdf-schema#Literal
icqChatID	Agent	http://www.w3.org/2000/01/rdf-schema#Literal
yahooChatID	Agent	http://www.w3.org/2000/01/rdf-schema#Literal
msnChatID	Agent	http://www.w3.org/2000/01/rdf-schema#Literal
homepage	Agent	Document
weblog	Agent	Document
tipjar	Agent	Document
made	Agent	http://www.w3.org/2000/01/rdf-schema#Resource
holdsAccount	Agent	OnlineAccount
SWAP/middleInitial	SWAP/Person	
SWAP/middleName	SWAP/Person	
SWAP/departmentName	SWAP/Person	
SWAP/firstName	SWAP/Person	
SWAP/lastName	SWAP/Person	
SWAP/personalSuffix	SWAP/Person	
SWAP/personalTitle	SWAP/Person	
SWAP/organization	SWAP/Person	
SWAP/home	SWAP/Person	SWAP/ContactLocation
SWAP/mobile	SWAP/Person	SWAP/ContactLocation
SWAP/office	SWAP/Person	SWAP/ContactLocation
SWAP/vacationHome	SWAP/Person	SWAP/ContactLocation
SWAP/motherTongue	SWAP/Person	SWAP/LanguageCode
SWAP/emailAddress	SWAP/SocialEntity	SWAP/_EmailAddress
SWAP/mailboxURI	SWAP/SocialEntity	SWAP/_URI
SWAP/mailbox	SWAP/SocialEntity	SWAP/Mailbox
GEO/lat	GEO/SpatialThing	
GEO/long	GEO/SpatialThing	
GEO/alt	GEO/SpatialThing	

In terms of outside superclass properties, basic person properties (SWAP/middleInitial, SWAP/middleName, SWAP/firstName, SWAP/lastName), email properties (SWAP/emailAddress, SWAP/mailboxURI, SWAP/mailbox) and geographical properties (GEO/lat, GEO/long, GEO/alt) are defined as properties of SWAP/Person, SWAP/SocialEntity and GEO/SpatialThing respectively. Looking at the above superclass-related class and property information, however, do not provide enough hints to determine what domain this ontology covers. The best guess may come from knows, a direct property of the Person Class, since it reflects the social networking aspect.

In summary, the following conclusions are drawn from the analysis of superclass, subclass, and property information:

- Understanding of the domain is derived at the subclass level. In particular, knowledge of the domain can be obtained by looking at subclasses and subclass properties.
- Metadata defined at superclass level may be too abstract to provide concrete description about the domain, but superclass properties may give ideas regarding the ontology usage, since properties are inherited.

3.2 Investigating Class Hierarchies

Understanding the class hierarchy defined in an ontology is a minimum requisite for correctly understanding the properties of individual classes. This is especially true when classes have superclasses, and these superclasses have properties. Because all superclass properties are inherited and passed onto its child classes, to completely understand the properties of a given class with superclasses, all superclass properties, if they exist, must be called into account. With this point in mind, we now look at class hierarchies.

Figure 3.1 shows two Person Class hierarchies represented in graphs. The left and right graphs each represent Person Class superclass hierarchies in CIDOC CRM ontology [32] and FOAF ontology [14] respectively. Refer to Tables 3.4 and 3.5 for detailed Person Class metadata information. The ovals represent classes and directed edges represent subClassOf (subsumption) relation between two classes.

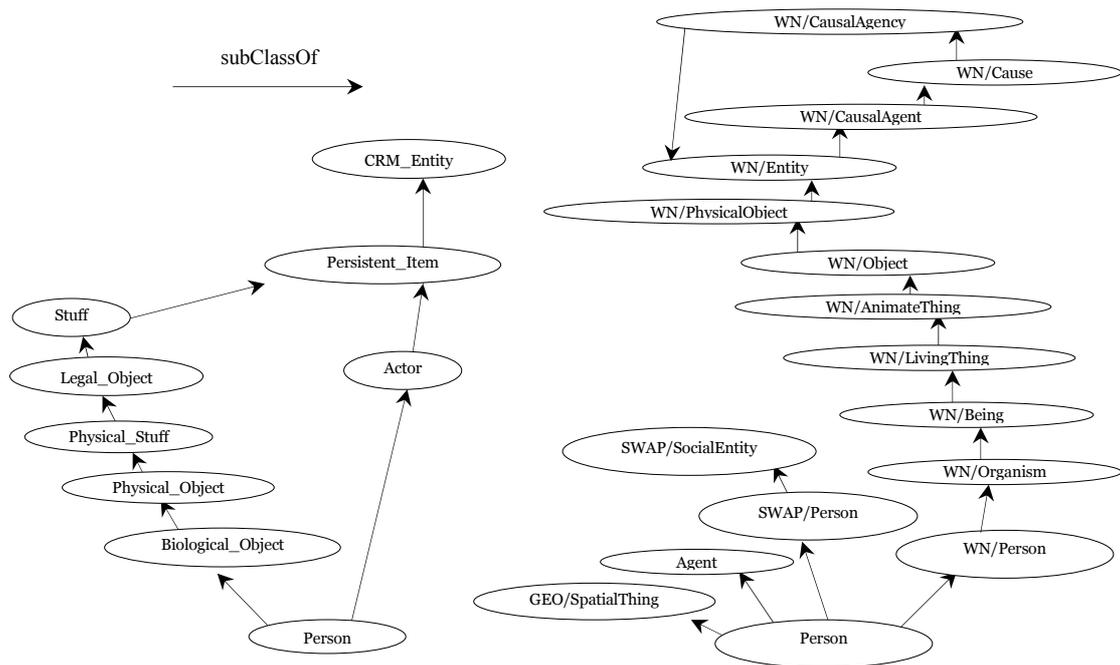


Figure 3.1: Person Class superclass hierarchy in CIDOC CRM ontology (left) and FOAF ontology (right)

CIDOC CRM: http://simile.mit.edu/repository/ontologies/official/cidoc_crm.rdfs
 FOAF: <http://simile.mit.edu/repository/ontologies/official/foaf.owl>
 GEO/SpatialThing: http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing
 SWAP/Person: <http://www.w3.org/2000/10/swap/pim/contact#Person>
 WORDNET/Person: <http://xmlns.com/wordnet/1.6/Person>

Examining the graph on the left, all Person Class superclasses are defined within the same ontology “http://simile.mit.edu/repository/ontologies/official/cidoc_crm.rdfs.” Here, Actor Class and Biological_Object (Class is omitted from here on) are defined as direct superclasses of Person Class. Similarly, starting with Biological_Object, Biological_Object has Physical_Object as superclass, Physical_Object -> Physical_Stuff, Physical_Stuff -> Legal_Object, Legal_Object -> Stuff, and Stuff -> Persistent_Item. Starting with Actor, Actor has Persistent_Item as superclass. Finally, Persistent_Item has CRM_Entity as superclass. As a result, all subclasses of CRM_Entity inherit CRM_Entity properties. It is interesting to see that Person Class is defined as a subclass of Persistent_Item.

Shifting to the graph on the right, Person Class has four direct superclasses, GEO/SpatialThing, Agent, SWAP/Person, and WN/Person. Save for Agent,

the three superclasses come from the outside ontology. Note that “/ (slash)” included in the class name indicates that the class comes from the outside ontology. Since `Person Class` inherits all superclass properties, it inherits properties defined both from within (`Agent`) the ontology and from the outside (`GEO/SpatialThing`, `SWAP/Person` and `SWAP/SocialEntity`) of the ontology. However, no properties are defined in `WN/Person`.

One caveat is that `WN/Person`’s superclass hierarchy contains an infinite loop. `WN/Person` has ten superclasses starting with `WN/Organism`, but `WN/CausalAgency`, placed at the highest level of the class hierarchy, is defined both as a superclass and a subclass. Here, an infinite loop is created. This is a good example of tangled class hierarchy, and such definition should be reused with care.

Up until now, two superclass hierarchies were examined. The main difference between these two class hierarchies is that all classes on the left are defined within the same ontology while some of the classes on the right are defined from several different ontologies. In other words, the graph on the left represents class hierarchy with all class definitions to be “closed” within a single ontology, whereas the graph on the right represents class hierarchy with some class definitions to be “open” to other ontologies. As a result, property definitions are domain-specific and consistent in the case of the former, while overlapping (e.g. name and email) or complementary (e.g. geographical information) in the case of the latter. To elaborate on the latter case, the following can be said. When outside classes are incorporated into the class hierarchy, it is done so for two reasons: 1) to align properties defined locally with similar properties defined in other ontologies, and 2) to complement classes or properties missing in the local ontology.

3.3 Remaining Five Ontologies

Table 3.6: A short ontology on organization domain

URI	http://www.ontotext.com/otk/2002/05/enterprise.rdfs	
CLASS	SUBCLASSOF	
Person	LegalAgent	
PROPERTIES	DOMAIN	RANGE
hasPosition	Person	Position
isBossOf	Person	Person

Table 3.7: An ontology with general person properties

URI	http://daml.umbc.edu/ontologies/ittalks/person	
CLASS	SUBCLASSOF	
Person		
PROPERTIES	DOMAIN	RANGE
firstName	Person	http://www.w3.org/2001/XMLSchema#string
lastName	Person	http://www.w3.org/2001/XMLSchema#string
title	Person	http://www.w3.org/2001/XMLSchema#string
gender	Person	http://www.w3.org/2001/XMLSchema#string
birthday	Person	http://daml.umbc.edu/ontologies/ittalks/event#Instant
homeAddress	Person	http://daml.umbc.edu/ontologies/ittalks/address#Address
officeAddress	Person	http://daml.umbc.edu/ontologies/ittalks/address#Address
email	Person	http://www.w3.org/2001/XMLSchema#string
homePhone	Person	http://www.w3.org/2001/XMLSchema#string
officePhone	Person	http://www.w3.org/2001/XMLSchema#string
cellPhone	Person	http://www.w3.org/2001/XMLSchema#string
fax	Person	http://www.w3.org/2001/XMLSchema#string
pager	Person	http://www.w3.org/2001/XMLSchema#string
homepage	Person	http://www.w3.org/2001/XMLSchema#anyURI

Table 3.8: A short ontology with general person properties

URI	http://dublincore.org/2000/06/07-org	
CLASS	SUBCLASSOF	
Person	http://rdf.dev.oclc.org/eor/2000/03/13-dctype#Resource	
PROPERTIES	DOMAIN	RANGE
name	Person	
affurl	Person	
affname	Person	
email	Person	

Table 3.9: Another ontology with general person properties

URI	http://www.isi.edu/webscripser/person.o.daml	
CLASS	SUBCLASSOF	
Person		
PROPERTIES	DOMAIN	RANGE
givenName	Person	
familyName	Person	
employer	Person	
emailAddress	Person	
phoneNumber	Person	

Table 3.10: An online newsfeed ontology

URI	http://dannayayers.com/2004/02/atom/atom.owl	
CLASS	SUBCLASSOF	
Person	AtomConstruct	
PROPERTIES	DOMAIN	RANGE
author	Entry, Feed	Person
contributor	Entry, Feed	Person
name	Person	http://www.w3.org/2001/XMLSchema#string
email	Person	http://www.w3.org/2001/XMLSchema#string
url	Person	http://www.w3.org/2001/XMLSchema#string

Chapter 4 Metadata Design Experiment

Metadata design process relies on the expertise of human metadata designer. Domain knowledge and design skill are required on the part of the human designer to design adequate metadata. Such requirement places a hurdle on metadata creation. To overcome this hurdle, designing difficulties caused by designer's lack of knowledge or skill must be addressed. In order to do this, difficulties in metadata design process, especially those that arise from lack of knowledge or expertise, must be clarified.

4.1 Objective

The goal of this experiment is to clarify what information needs arise during the design process. Difficulties in the design process partly arise from designer's lack of knowledge, so by supplementing this lack of knowledge with appropriate outside knowledge (information), difficulties in metadata design can be mitigated. In this experiment the author seeks to identify 1) what information is available at hand regardless of designer's knowledge and skill, and 2) what information is not available. The 2) is set complement of 1). The union knowledge is defined as knowledge required for designing a set of metadata.

A trial metadata design experiment was conducted by the author. A set of metadata for some target document was designed. Detailed information needs that arise during the design process were observed and documented. For the ontology description language, OWL DL [24] was used.

4.2 Data and Method

4.2.1 Target Document

A user input form of an online social networking service called Orkut [23] was selected as the target document to be designed. Orkut allows registered users to input various information about themselves. The user information is received through six different user input pages: general page (receives name, gender, etc.), contact page (email, phone, etc.), photo page (photo), professional page (education, occupation, etc.), and personal page (personal

traits like eye and hair color). In this experiment, general page was chosen as the sample document (Figure 4.1). The author designed a set of metadata that describes each input/selection items contained in the sample document. The entire general page items are listed in Table 4.1. Items on the left such as first name and last name are located in the first column of the table; detailed items on the right and HTML form control types like text boxes and radio buttons are located in the second column.

The screenshot shows a 'general' profile page with the following fields and values:

- first name: Heeryon
- last name: Cho
- gender: female male
- relationship status: no answer
- birth day: jun 26
- birth year: 1972
- zip / postal code: (US or Canada only)
- country: Korea (Republic of)
- primary language: English
- i'm interested in:
 - friends
 - activity partners
 - business networking
 - dating choose
- children: no answer
- ethnicity: asian
- religion: Christian/Protestant
- political view: no answer

Figure 4.1: Partial screenshot of Orkut user input form

4.2.2 What Information Is Given

Table 4.1 shows detailed items contained in the target document. These items or explicit information, as is, constitute the “information available” defined in section 4.1. Regardless of author’s background knowledge or design skill, this information is given to the author.

Table 4.1: User input items in Orkut general page

Left Side Items	Right Side Items and/or HTML Form Control Types <small>◦ Radio button (choose one),[▼] Menu (choose one),☑ checkbox (multiple choice),[Textbox]</small>
first name	[Textbox]
last name	[Textbox]
gender	◦ female, male
relationship status	[▼] single, married, committed, open marriage, open relationship
birth day	[▼] jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec [▼] 1 to 31
birth year	[▼] 1920 to 1986
zip / postal code	[Textbox]
country	[▼] United States, Canada, Afghanistan, ..., Zimbabwe (Total of 227 countries)
primary language	[▼] English, Afrikaans, Ainu, Albanian, ..., Zulu (Total of 115 languages)
i'm interested in	☑ friends, activity partners, business net-working dating(men&women, men, women)
children	[▼] no answer, no, yes-at home full time, yes-at home part time, yes-not at home
ethnicity	[▼] no answer, african american (black), asian, caucasian (white), east indian, his-panic/latino, middle eastern, native american, pacific islander, multi-ethnic, other
religion	[▼] no answer, Agnostic, Atheist, Buddhist/Taoist, Christian/Catholic, Christian/LDS, Christian/Protestant, Christian/Other, Hindu, Jewish, Islam, Spiritual but not religious, Religious humanism, other
political view	[▼] no answer, right-conservative, very right-conservative, centrist, left-liberal, very left-liberal, libertarian, very libertarian, authoritarian, very authoritarian, depends, not political
sense of humor	☑ campy/cheesy, goofy/slapstick, dry/sarcastic, obscure, clever/quick witted, raunchy, friendly
sexual orientation	[▼] no answer, straight, gay, bisexual, bi-curious
fashion	☑ alternative, casual, classic, contemporary, designer, minimal, natural, outdoorsy, smart, trendy, urban
smoking	[▼] no answer, no, socially, occasionally, regularly, heavily, trying to quit, quit
drinking	[▼] no answer, no, socially, occasionally, regularly, heavily
pets	[▼] no answer, I love my pet(s), I like them at the zoos, I like pet(s), I don't like pets
living	☑ alone, with kid(s), with roommate(s), with parents, with partner, friends visit often, with pet(s), party every night
home town	[Textbox] (in [▼] no answer, Alabama, Alaska, ..., Wyoming (50 States in the U.S.))
web page	[Textbox]
describe yourself	[Textbox]

Apart from explicit information in Table 4.1, the following structural information is inherent in the target document.

Overall Layout

The target document is vertically divided into two areas, left side area and right side area. The left side area contains attributes of person, such as name, country and gender, and the right side area contains detailed values for these attributes. Moreover, right side area items and values are recorded using HTML control types. These HTML control types place datatype, cardinal or structural restrictions on the values of person attributes.

HTML Control Types

Four kinds of HTML control types are used in the target document: text inputs, radio buttons, menus, and check boxes. These control types place specific restrictions on the items in the target document. These restrictions can in turn be expressed in OWL. Using the target document examples, each control type's restriction is explained in detail.

① Text input

Text input or text field is an HTML control type that receives some string value as user input. In most cases, inputted data's datatype is string. Taking this into account, a left column item with text input on the right column can be interpreted as datatype property having string data as its value. Figure 4.2 shows three text input examples contained in the target document. Following this figure is an OWL description that expresses this datatype assignment. RDF triples that are generated from the description are given in Table 4.2.

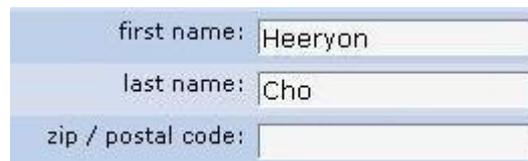


Figure 4.2: Examples of text input item contained in the target document

```
<owl:DatatypeProperty rdf:ID="first_name">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="last_name">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="zip-postal_code">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
</owl:DatatypeProperty>
```

Table 4.2: RDF triples that can be generated from the description above

Subject	Predicate	Object
first_name	rdf:type	owl:DatatypeProperty
first_name	rdfs:range	xsd:string
last_name	rdf:type	owl:DatatypeProperty
last_name	rdfs:range	xsd:string
zip-postal_code	Rdf:type	owl:DatatypeProperty
zip-postal_code	rdf:range	xsd:string

② Radio button

Radio button is an HTML control type that receives one user-selected item which is among mutually exclusive multiple items. Because only a single item can be selected, the cardinality of the selected value will always be one. Figure 4.3 shows a radio button example contained in the target document. The left side item, gender, can be interpreted as having either one of the values, female or male. The OWL description of this cardinality constraint and RDF triples generated from the OWL description (Table 4.3) are provided below.



Figure 4.3: An example of radio button item included in the target document

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#gender" />
  <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
</owl:Restriction>
```

Table 4.3: RDF triples that can be generated from the description above

Subject	Predicate	Object
SomeClass	rdf:type	owl:Restriction
SomeClass	owl:cardinality	1
SomeClass	owl:onProperty	gender

③ Menu

Menu is an HTML control type that receives one use-selected item that has been selected from a list of multiple items. In the target document, two different cases of menu items are given: The first case is a menu containing a “no answer” item (Figure 4.3 left). Here, the user has the option to withhold the item selection by choosing “no answer”. Hence, the item selected can be zero or one, which in turn can be interpreted as cardinality constraint of maximum cardinality equals one. The second case is a menu without a “no answer” item (Figure 4.3 right). Here, one item always must be selected, so the cardinality constraint of the selected value becomes exactly one. The OWL description of these examples and RDF triples generated (Table 4.3) from the description are provided below the Figure 4.3.



Figure 4.3: A menu with (left) and without (right) a “no answer” item

```

<owl:Restriction>
  <owl:onProperty rdf:resource="#relationship_status" />
  <owl:maxCardinality rdf:datatype="&xsd;positiveInteger">
    1</owl:maxCardinality>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#country" />
  <owl:cardinality rdf:datatype="&xsd;positiveInteger ">1</owl:cardinality>
</owl:Restriction>

```

Table 4.3: RDF triples that can be generated from the description above

Subject	Predicate	Object
SomeClass	rdf:type	owl:Restriction
SomeClass	owl:maxCardinality	1
SomeClass	owl:onProperty	relationship_status
SomeClass	rdf:type	owl:Restriction
SomeClass	owl:cardinality	1
SomeClass	owl:onProperty	country

④ Check box

Check box is an HTML control type that receives multiple user-selected item values. In the target document, check box items on the right are usually enumerated examples of the left side item. These right side multiple items can be defined as individuals of the left side item using the owl:oneOf element.



Figure 4.4: An example of check box items contained in the target document

Below is an OWL description of the check box example given in Figure 4.4. RDF triples for this example are omitted.

```

<owl:Class rdf:ID="SenseOfHumor">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Campy_or_cheesy_humor" />
    <owl:Thing rdf:about="#Goofy_or_slapstic_humor" />
    <owl:Thing rdf:about="#Dry_or_sarcastic_humor" />
    <owl:Thing rdf:about="#Obscure_humor" />
    <owl:Thing rdf:about="#Clever_or_quick_witted_humor" />
    <owl:Thing rdf:about="#Raunchy_humor" />
    <owl:Thing rdf:about="#Friendly_humor" />
  </owl:oneOf>
</owl:Class>

<SenseOfHumor rdf:ID="Campy_or_cheesy_humor" />
<SenseOfHumor rdf:ID="Goofy_or_slapstic_humor" />
<SenseOfHumor rdf:ID="Dry_or_sarcastic_humor" />
<SenseOfHumor rdf:ID="Obscure_humor" />
<SenseOfHumor rdf:ID="Clever_or_quick_witted_humor" />
<SenseOfHumor rdf:ID="Raunchy_humor" />
<SenseOfHumor rdf:ID="Friendly_humor" />

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <orkut:SenseOfHumor rdf:about="#Campy_or_cheesy_humor" />
    <orkut:SenseOfHumor rdf:about="#Goofy_or_slapstic_humor" />
    <orkut:SenseOfHumor rdf:about="#Dry_or_sarcastic_humor" />
    <orkut:SenseOfHumor rdf:about="#Obscure_humor" />
    <orkut:SenseOfHumor rdf:about="#Clever_or_quick_witted_humor" />
    <orkut:SenseOfHumor rdf:about="#Raunchy_humor" />
    <orkut:SenseOfHumor rdf:about="#Friendly_humor" />
  </owl:distinctMembers>
</owl:AllDifferent>

```

In sum, the following information is provided in the target document.

- Person attributes (left side area items) and detailed selection items (right side area items) for those attributes.
- HTML control types which can be interpreted as cardinal, datatype, structural constraints.

The former constitute the explicit information while the latter include the implicit information. In any case, both are present in the target document, and the human designer can utilize both information regardless of his or her domain knowledge and design skill.

4.2.3 Metadata Design Step

Based on the careful analysis of the target document, the metadata design experiment was carried out in the following steps. Since the author did not have prior experience in metadata design, the following documents were referred to: OWL Language Specification [5], a user's guide to creating ontology [25], and a wine ontology in OWL specification [26]. Other than these documents, no other resources, including any existing ontologies on the WWW, were referred. This restriction was placed deliberately on the author to highlight the information needs that arise during the design process. As for the actual design steps, six of the seven steps in [25] were performed. Note that the second step, which recommends ontology reuse, was skipped because reusing other ontologies will violate the restriction placed on the experiment. Below are steps 1 through 7 given in [25]. Actual design steps that the author performed are explained in the steps 1e through 7e.

- Step 1: Determine the domain and scope of the ontology.
 - > Step 1e: Set the domain of the ontology to social networking services, and limit the scope to the Orkut general page.
- Step 2: Consider reusing existing ontologies.
 - > Step 2e: Skip.
- Step 3: Enumerate important terms in the ontology.
 - > Step 3e: Let important terms be equal to the user input items given by the Orkut general page. Enumerate the input items as important terms.
- Step 4: Define the classes and the class hierarchy.
 - > Step 4e: If items are defined as classes, define the class hierarchy.
- Step 5: Define the properties of classes-slots.
 - > Step 5e: Define properties of the classes.
- Step 6: Define the facets of the slots.
 - > Step 6e: Define cardinality, datatype, domain, range of the properties.
- Step 7: Create instances.
 - > Step 7e: Create instances.

4.3 Result

After executing design steps 1e through 7e, a sample ontology (see Appendix A2) containing metadata of the target document was created. The syntax of the resulting ontology was checked using an OWL Validator [27]. The following difficulties were observed during the design process.

- Determining whether an item should be designed as a class or a property is difficult.
- Determining what subsumption relation holds between two classes is difficult.
- Designing object property that defines the relationship of two classes is difficult.
- Being confident that most major properties of the target are designed is difficult.

Each of the difficulties can be regarded as information needs problem.

4.3.1 Class/Property Uncertainty

Determining whether an item should be designed as a class or simply as a property was difficult. Classes provide an abstraction mechanism for grouping resources with similar characteristics. A class description describes an OWL class, either by a class name or by specifying the class extension of an unnamed anonymous class [28]. During the trial design process, it was difficult to determine whether items like first name, last name, gender, and web page should be modeled as classes.

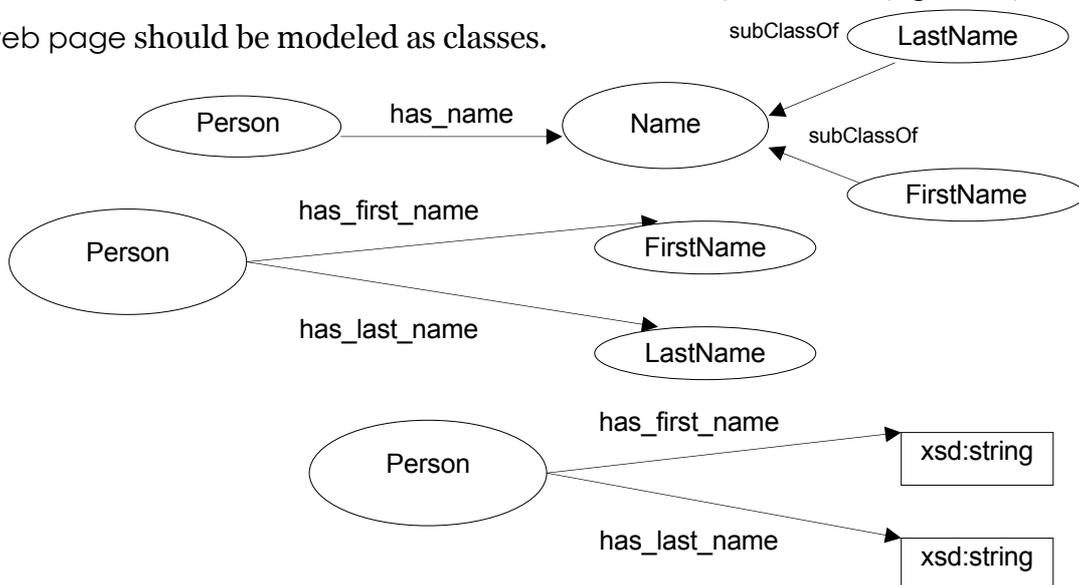


Figure 4.5: Name item designed as class (top, center) and property (bottom)

Figure 4.5 shows three different ways of designing the name item. The top graph shows Name Class having FirstName and LastName as its subclasses,

and linked to Person Class through has_name property. The center graph shows FirstName Class and LastName Class existing on its own, and each linked to Person Class through has_first_name and has_last_name property respectively. The bottom graph shows first name and last name items each designed as has_first_name and has_last_name property respectively without any classes.

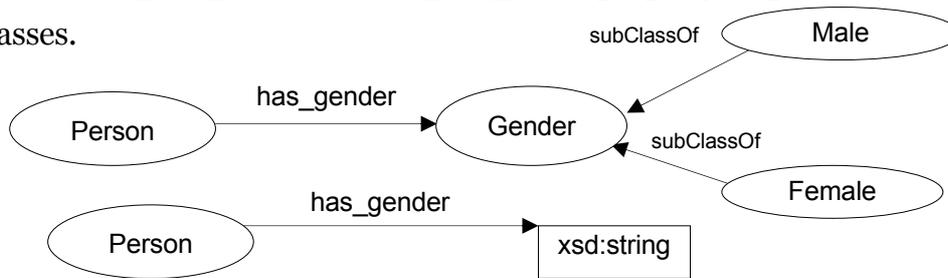


Figure 4.6: Gender item designed as class (top) and as property (bottom)

Similar design dilemma exists in the case of the gender item. In Figure 4.6, the top graph shows gender item designed as Gender Class subsuming Male and Female Class, whereas the bottom graph shows it designed as has_gender property without any classes.

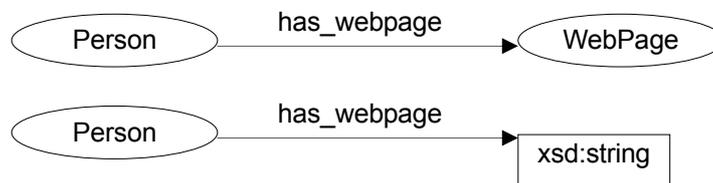


Figure 4.7: Web page item designed as class (top) and as property (bottom)

And also for the web page item. Figure 4.7 shows two variations for designing a web page item, one designed as a class (top) and the other designed as a property (bottom). Such variations to design can raise uncertainty in the designer since it is difficult to know which might be the better design. If the author had a chance to refer to existing ontologies for actual examples of name, gender and webpage definitions, some design hints might have been obtained. The design difficulty observed in these examples can be summarized as follows.

Design Difficulty 1. Determining whether an item should be designed as a class or a property is difficult.

4.3.2 Subsumption Relation Uncertainty

Determining what subsumption relation holds between two classes was difficult. Subsumption relation, which is the basis of a taxonomy, is an extremely useful tool for imparting structure on an ontology [29]. Subsumption relation can be defined by using “`rdfs:subClassOf`” element. Determining whether one class subsumes another, however, is difficult to determine, and [29] also point out that subsumption relation is often misused. For example, in the trial design experiment, the author did not know what class hierarchy to design for the Female and Male Class.

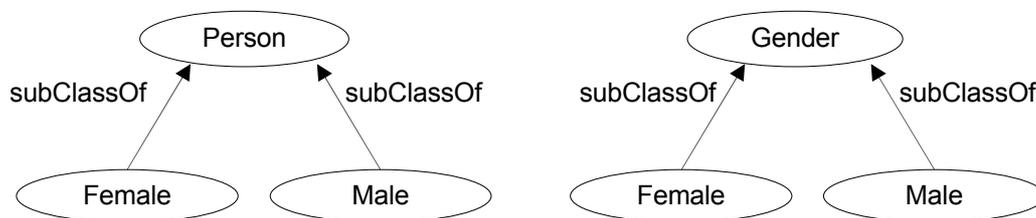


Fig. 4.7: Person Class subsuming Female and Male Class (left) and Gender Class subsuming Female and Male Class (right)

At first, male and female items were designed as individuals of the Gender Class by simply noticing the target document’s layout and HTML control type (radio button). On second thought, the author came up with two possibilities of imparting class hierarchy to the two items. Figure 4.7 shows the two possible subsumption relations. The left oval shows Person Class subsuming Female and Male Class. The right oval shows Gender Class subsuming Female and Male Class. Both, one or none may be correct. Unable to decide, the author settled with the first design. If the author were able to refer to other ontologies for class hierarchy definitions, those examples might have helped the author to make better design decision. The design difficulty observed in this example can be summarized as follows.

Design Difficulty 2. Determining what subsumption relation holds between two classes is difficult.

4.3.3 Class-to-Class Relation Uncertainty

Determining what relation holds, aside from subsumption relation, between two classes was difficult. Relationship between two classes can be defined using the “owl:ObjectProperty” element. For example, in the experiment, defining object properties of the Person Class and Smoking Class, Person Class and Pets Class, Person Class and Country Class was difficult. Each case is explained in detail.

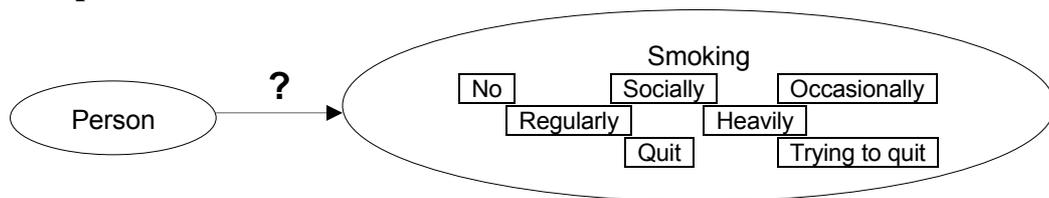


Figure 4.9: Defining relation between Person and Smoking Class is difficult

Smoking item in the target document has the following eight items as its selection values: no answer, no, socially, occasionally, regularly, heavily, trying to quit, and quit. Since no answer item is independent of the Smoking Class, the Smoking Class can be designed as a class containing the remaining seven values as its individuals (Figure 4.9). Designing relationship between Person Class and Smoking Class, however, is not easy. One way to do it is to simply define it as having has_smoking_values property. But this definition would be too specific to the target document. At the concept level, in which metadata is designed, a person does not have “smoking values”.

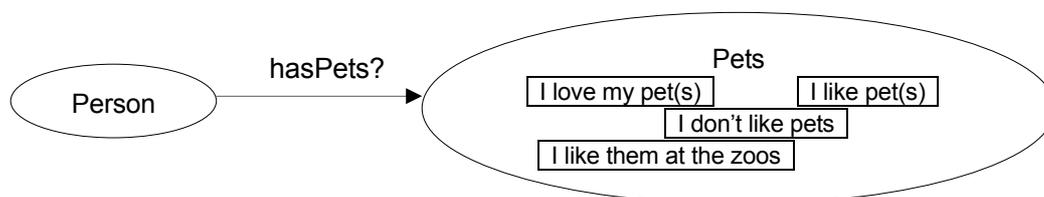


Figure 4.9: hasPets property may exist between Person and Pets Class

Pets item contains the following four selection values: I love my pet(s), I like them at the zoos, I like pet(s), and I don't like pets. Note that no answer value is omitted here also. The possible relationship holding between Person Class and Pets Class may be hasPets property (Figure 4.9). But uncertainty

remains because the selection items indicate person's "preference" for pets, not one's ownership.

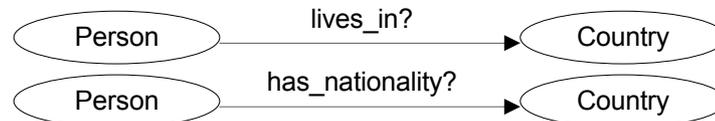


Figure 4.10: Object properties `lives_in` (top) and `has_nationality` (bottom) are possible relationships for Person and Country Class, but at the instance level, they are in conflict

The last example deals with instance level conflict. The target document contains country item, which can be interpreted either as a place where one resides in or a place where one holds one's nationality. Based on these interpretations, either `lives_in` or `has_nationality` property would be appropriate for the relationship between two classes. However, at the instance level, i.e. user input data level, these two interpretations are in conflict with each other. This conflict arises for users living outside of their home country. For example, there are two Philippines (people with Philippine nationality) living in Korea, and one has Philippine as his country value while the other has Korea as his country value. In the case of the former, Person-Country relation is interpreted as `has_nationality`, while the latter is interpreted as `lives_in`. A metadata encompassing the two conflicting interpretations would be better if it can be designed. In sum, based on the examples observed above, the following design difficulty was discovered.

Design Difficulty 3. Designing object property that defines the relationship between two classes is difficult.

In all three cases, if it was possible to refer to other ontologies for class-to-class relationship information, the author might have obtained some design ideas.

4.3.4 Uncertainty of Comprehensive Metadata Coverage

The target document used for the experiment was general page of the Orkut user input form. Apart from the general page, there are five more pages that deals with person related information (Explanation given in section 4.2.1.) Although these pages can be easily obtained in the Orkut website, it may not always be the case relevant document or resource is easily obtained in other metadata design circumstances. In such cases, the designer may not have enough relevant information to design necessary metadata. As a result, if the designer does not have enough domain knowledge, the designer has high chance of not being able to design necessary metadata. This is the comprehensive metadata coverage problem. This problem is explained in detail using a hypothetical example.

Let us imagine that metadata designer is provided with only the Orkut general page to design metadata for the social networking domain. He or she would do best to design metadata based on the information contained in the document, but would be unsure if important metadata or major metadata have all been designed. Based on the information provided in the general page, name, gender, birth day and web page properties could be designed as major properties for describing person (Figure 4.11), but it is difficult to assess whether these four properties are major properties sufficient for person. This is the comprehensive property coverage problem.

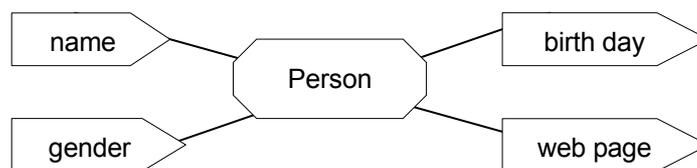


Figure 4.11: Name, gender, birth day and web page attributes can be regarded as major properties of the concept “person”

Design Difficulty 4. When relevant information about the target is insufficient, being confident that most major properties of the target are designed is difficult.

Chapter 5 Ontology as Design Material

The difficulties observed during the trial metadata design experiment can be alleviated by referring to existing ontologies on the WWW. If somewhere on the WWW there exist metadata that closely relates to the metadata the designer has difficulty is designing, then, by referring to them, the designer can know how others have designed similar metadata. These actual metadata examples may aid the metadata designer to make better design decisions.

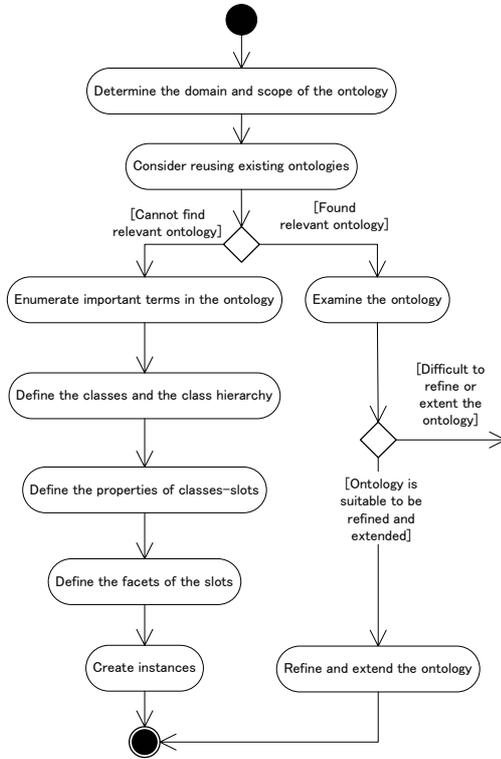
In this chapter, detailed design steps that incorporate existing ontologies is introduced. Section 5.1 discusses a step by step metadata design process that focuses on ontology reuse. The four design difficulties observed in Chapter 4 can be alleviated by following the ontology referral steps introduced. Section 5.2 explains a prototype tool which automatically generates design materials from existing ontologies. The system architecture and the graphical user interface of the tool are explained. Finally, in section 5.3, detailed content of the design material is described. Four different kinds of design materials that each meet of the information needs observed in Chapter 4 are described, three of them with real metadata examples.

5.1 Metadata Design Process Focusing On Ontology Reuse

Figure 5.1 shows a UML activity diagram of a simple metadata design process that deals with metadata design from scratch [25] (left) and a metadata design process focusing on ontology reuse proposed by the author (right, enclosed in a box). The proposed design process aims at supporting human metadata designer when information needs or uncertainties arise.

For example, it is often the case the human designer is faced with multiple design possibilities and is unsure which design direction might be correct. In such cases, the design process introduces the ontology referral step, where the designer can refer to relevant ontologies for hints. This referral step is called the “RIDE step,” and it forms a central part of the proposed design process. “RIDE” stands for “Retrieves Information” and “DEcide.” This is precisely what the designer does in this step.

A Simple Knowledge-Engineering Methodology



Designing Metadata With Existing Ontologies

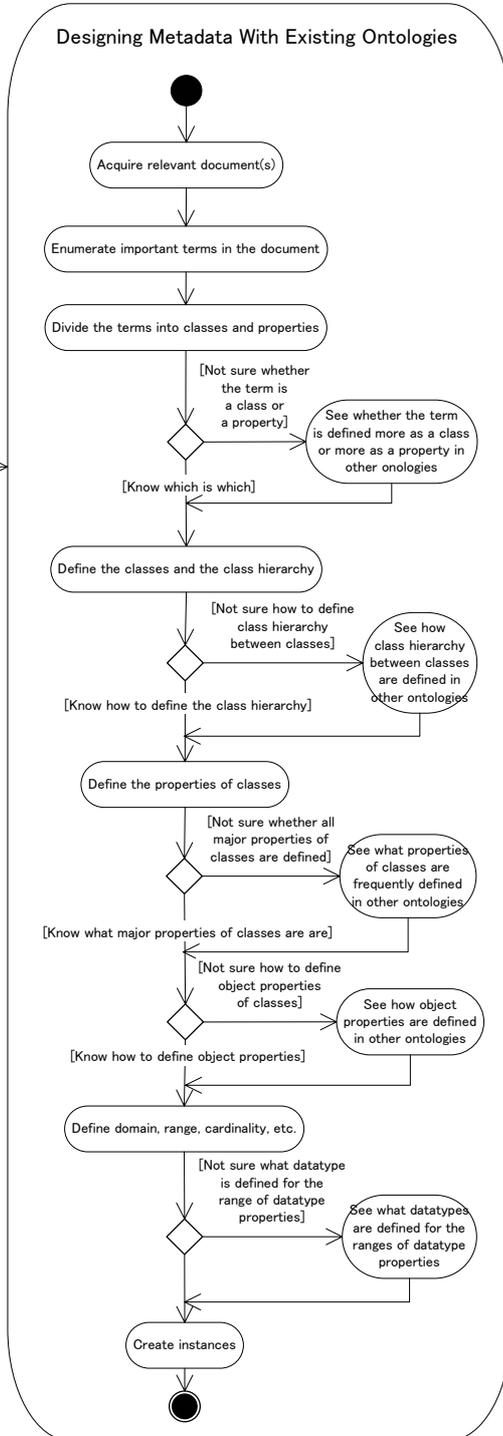


Figure 5.1: Metadata design process focusing on ontology reuse (inside box)

The RIDE step is introduced five times during the overall design process. In each RIDE step, specific metadata information needs observed in Chapter 4 are being met. The first RIDE step tackles the problem of class/property uncertainty. The designer refers to relevant class and property examples in this step. The second RIDE step tackles the problem of subsumption relation uncertainty. The designer refers to existing class hierarchy information in this step. The third RIDE step tackles the problem of comprehensive property coverage. The designer refers to multiple ontologies for key properties in this step. The fourth RIDE step tackles the problem of class-to-class relation uncertainty. The designer refers to other object properties in this step. Finally, the fifth RIDE step tackles the problem of datatype uncertainty. The designer unfamiliar with the datatype concept refers to datatype definitions in this step. In all five steps, the designer refers to the tool-generated design material for relevant metadata information.

The overall flow of the design process does not deviate greatly from the design steps introduced in [25], which focus on designing metadata from scratch. Six of the seven steps in [25], except for the reuse step, were executed in the trial design experiment, and the author found these steps to be logical and useful. However, a new step is introduced in the proposed design process, a step that divides items to be designed into either class or property. During the trial experiment, hesitation about whether to design an item as a class or a property occurred repeatedly, and the author judged it to be a key problem to the design process. Hence, this step is introduced.

In sum, the metadata design process which reuses existing ontologies leverages on the RIDE step to meet specific information needs that arise during the design process. Focusing on the specific design difficulties tackled in each RIDE step, the prototype tool was implemented.

5.2 Prototype Tool

The prototype tool receives designer's query as input to generate relevant design material. This input consists of i) a single (or two) string keyword that represents metadata item to be designed and ii) specific information need, i.e.

one of the four design difficulties observed in Chapter 4 plus a separate design difficulty (datatype uncertainty) that the designer is facing. The latter input is received in the form of radio button selection made by the user. See section 5.2.3 for details. The output design material is a comma-separated-values file, i.e. a table file.

5.2.1 System Architecture

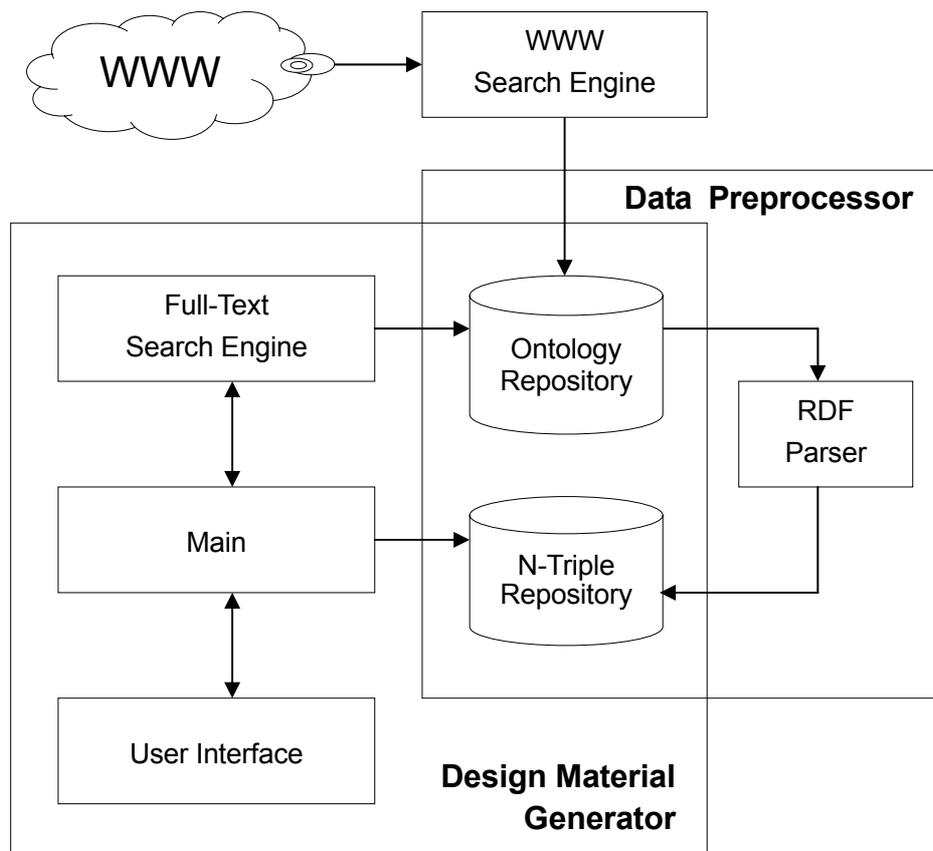


Figure 5.2: System architecture of the prototype tool

The prototype tool generates design material from RDF triples stored in the local repository. These RDF triples are produced in advance, from ontologies collected from the WWW using WWW search engine. Once the main program receives designer's query (string keyword and type of information need) via the user interface, the string keyword is handed over to the full-text search engine [namazu], and the text search engine searches the local ontology repository for

ontology files containing the matching keyword. When the search result is returned, the main program uses the search result and the type of information need thrown as query from the user to retrieve relevant RDF triples in the RDF triple repository. After relevant RDF triples are collected, relevant design material is created and the main program outputs the file.

5.2.2 Sample data

Ontologies in the form of XML files were collected in advance from the WWW using the WWW search engine. The filetype option of the search engine was exploited to collect ontology files with five filetype extensions: daml, oil, owl, rdf and rdfs. These ontology files were parsed using an RDF parser [33] to generate RDF triples. Total of 2,229 ontology files were collected (daml: 436, oil: 17, owl: 564, rdf: 932, rdfs: 280), but some were not ontology or well-formed files so these were removed.

5.2.3 Graphical User Interface

Figure 5.3 shows the graphical user interface of the prototype tool. The designer first selects one of the six radio buttons listed in the interface to address the specific information need s/he is facing. After the radio button is selected, the designer can input the keyword(s) in the text field. This keyword should represent the metadata item the designer wishes to refer to. Once the “Search” button is clicked, the main program initiates the generation of the relevant design material, and outputs the result in a table file.

5.3 Design Materials

The output of the design material differs according to the information need (or the radio button) selected by the designer. In this section, the design material, which relates to the four design difficulties observed in Chapter 4, is described in detail. For the first three design difficulties, design materials generated from real ontologies collected from WWW will be introduced as examples. The algorithms for automatically generating design material for the first two cases will be given. Note that the current algorithm deals only with lexical matching. For the third case, a limited solution is proposed. For the last case, the style of the design material is proposed.

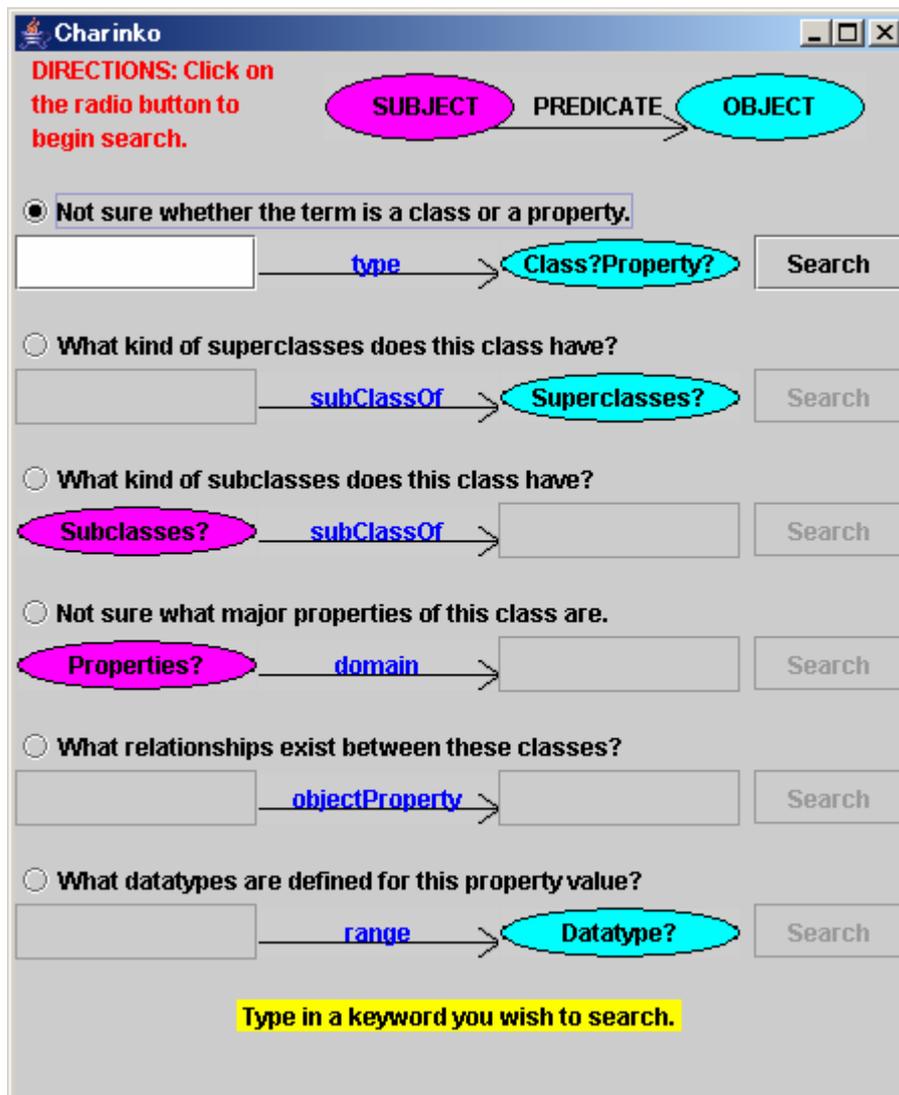


Figure 5.3: Graphical user interface of the prototype tool

5.3.1 Majority Rule as Hints to Class/Property Decision

The uncertainty of whether to design a target item as class or property can be reduced by comparing existing class and property descriptions of that item in other ontologies. By looking at the multiple class and property examples on the spot, the human designer can know which definition is the majority case. Three class/property uncertainty example problems of name, gender and web page items were observed in Chapter 4. The design material for gender and web page are given in Tables 5.1 and 5.2 respectively. The design material for name is omitted.

Table 5.1: Majority rule states that gender is a class

Metadata	Type	Full URI	OntFile
GenderValue	class	http://emeld.douglass.arizona.edu:8080/gold_1-2-2.owl#GenderValue	owl_419
gender	class	http://lonely.org/russia#gender	rdf_743*
gender	class	http://lonely.org/russia#gender	rdf_298*
GenderValue	class	http://www.emeld.org/gold.owl#GenderValue	owl_477
Gender	class	http://www.cs.wayne.edu/~shiyong/ontology/FamilyAlbum.owl#Gender	Owl_180*
Gender	class	http://www.cs.wayne.edu/~shiyong/ontology/FamilyAlbum.daml#Gender	daml_98*
Gender	class	http://www.cs.umd.edu/projects/plus/DAML/onts/personal1.o.daml#Gender	daml_46
Gender	class	http://protege.stanford.edu/plugins/owl/owl-library/koala.owl#Gender	owl_36*
Gender	class	http://protege.stanford.edu/plugins/owl/owl-library/koala.owl#Gender	owl_247*
E76.Gender	class	http://cidoc.ics.forth.gr/rdfs/cidoc_erm_v3.2.1.rdf#E76.Gender	rdf_237*
E76.Gender	class	http://139.91.183.30:9090/RDF/VRP/Examples/CIDOC1.rdf#E76.Gender	rdf_206*
Grammatical GenderValue	class	http://emeld.org/gold.owl#GrammaticalGenderValue	owl_401
gender	datatype-property	http://orlando.drc.com/daml/ontology/Person/3.1/Person-ont2.daml#gender	daml_323
gender	datatype-property	http://orl01.drc.com/daml/Ontology/Person/3.1/Person-ont2.daml#gender	daml_252
gender	datatype-property	http://daml.umbc.edu/ontologies/profile-ont#gender	owl_339
gender	datatype-property	http://xmlns.com/foaf/0.1/gender	rdf_87
has_gender	property	http://lonely.org/russia#has_gender	rdf_743*
has_gender	property	http://lonely.org/russia#has_gender	rdf_298*
gender	property	http://xmlns.com/foaf/0.1/gender	owl_157
gender	property	http://xmlns.com/foaf/0.1/#gender	owl_519
gender	property	http://xmlns.com/foaf/0.1/#gender	owl_427
P61F.has_gender	property	http://cidoc.ics.forth.gr/rdfs/cidoc_erm_v3.2.1.rdf#P61F.has_gender	rdf_237*
P61B.is_gender_of	property	http://139.91.183.30:9090/RDF/VRP/Examples/CIDOC1.rdf#P61B.is_gender_of	rdf_206*
gender	property	http://www.cs.umd.edu/~golbeck/daml/running.daml#gender	daml_43
hasGender	object-property	http://www.cs.wayne.edu/~shiyong/ontology/FamilyAlbum.owl#hasGender	owl_180*
hasGender	object-property	http://www.cs.wayne.edu/~shiyong/ontology/FamilyAlbum.daml#hasGender	daml_98*
hasGender	object-property	http://protege.stanford.edu/plugins/owl/owl-library/koala.owl#hasGender	owl_36*
hasGender	object-property	http://protege.stanford.edu/plugins/owl/owl-library/koala.owl#hasGender	owl_247*
militaryGender Restriction	object-property	http://reliant.tekknowledge.com/DAML/Military.owl#militaryGenderRestriction	owl_58
OntFile	Ontology URL		
daml_252	http://orl01.drc.com/daml/Ontology/Person/3.1/Person-ont2.daml		
daml_323	http://orlando.drc.com/daml/ontology/Person/3.1/Person-ont2.daml		
daml_43	http://www.cs.umd.edu/~golbeck/daml/running.daml		
daml_46	http://www.cs.umd.edu/projects/plus/DAML/onts/personal1.o.daml		
daml_98*	http://www.cs.wayne.edu/~shiyong/ontology/FamilyAlbum.daml		
owl_157	http://simile.mit.edu/repository/ontologies/official/foaf.owl		
owl_180*	http://www.cs.wayne.edu/~yudeng/project/owlGraph/FamilyAlbum2.owl		
owl_247*	http://www.ling.helsinki.fi/kit/2004k/ctl310semw/Protege/koala.owl		
owl_339	http://daml.umbc.edu/ontologies/profile-ont.owl		
owl_36*	http://protege.stanford.edu/plugins/owl/owl-library/koala.owl		
owl_401	http://www.u.arizona.edu/~farrar/gold.owl		
owl_419	http://www.cs.wayne.edu/~yudeng/project/owlGraph/gold_brian.owl		
owl_427	http://bblfish.net/work/atom-owl/2004-08-12/Atom.old2.owl		
owl_477	http://coli.lili.uni-bielefeld.de/~felix/lehre/ws04_05/ontologischeRessourcen/beispiele/gold/gold.owl		
owl_519	http://bblfish.net/work/atom-owl/2004-08-12/Atom.old2.owl		
owl_58	http://reliant.tekknowledge.com/DAML/Military.owl		
rdf_206*	http://139.91.183.30:9090/RDF/VRP/Examples/CIDOC1.rdf		
rdf_237*	http://cidoc.ics.forth.gr/rdfs/cidoc_erm_v3.2.1.rdf		
rdf_298*	http://www.atl.external.lmco.com/projects/ontology/ontologies/russia/russiaB.rdf		
rdf_743*	http://www.aifb.uni-karlsruhe.de/WBS/meh/mapping/data/russia1a.rdf		
rd_87	http://svn.mindswap.org/psychinko/allotests/mindswapRealized.rdf		

Table 5.2: Using keyword “webpage” returns few results

Metadata	Type	Full URI	OntFile
webpage	class	http://wizard.imsa.edu/searchhistory#webpage	rdfs_179
webpage	object-property	http://mycampus.cs.cmu.edu/ontology/foa#webpage	owl_51
webpage	object-property	http://mycampus.cs.cmu.edu/ontology/foa#webpage	owl_400
OntFile	Ontology URL		
rdfs_179	http://alumni.imsa.edu/~vader21/searchhistory.rdfs		
owl_51	http://www-2.cs.cmu.edu/~sadeh/MyCampusMirror/ROWL/foa.owl		
owl_400	http://mycampus.sadehlab.cs.cmu.edu/public_pages/ROWL/foa.owl		

Table 5.3: Majority rules that homepage is a property

Metadata	Type	Full URI	OntFile
PersonalHomepage	class	http://www.atl.external.lmco.com/projects/ontology/ontologies/comsci/csa.rdf#PersonalHomepage	rdf_51
PersonalHomepage	class	http://www.cs.umd.edu/projects/plus/DAML/ont/docmnt1.o.daml#PersonalHomepage	daml_157
Homepage	class	http://www.iwi-iuk.org/material/RDF/1.1/Schema/Class/mn#Homepage	daml_197
Homepage	class	http://www.iwi-iuk.org/material/RDF/1.1/Schema/Class/mn#Homepage	daml_210
homepage	datatype-property	http://daml.umbc.edu/ontologies/wob/foaf-lite#homepage	owl_511
homepage	datatype-property	http://xmlns.com/foaf/o.1/#homepage	owl_508
homepage	datatype-property	http://xmlns.com/foaf/o.1/#homepage	owl_433
homepage	datatype-property	http://xmlns.com/foaf/o.1/#homepage	owl_435
homepage	datatype-property	http://daml.umbc.edu/ontologies/wob/foaf-lite#homepage	owl_438
schoolHomepage	property	http://xmlns.com/foaf/o.1/schoolHomepage	rdf_410
workInfoHomepage	property	http://xmlns.com/foaf/o.1/#workInfoHomepage	owl_519
workInfoHomepage	property	http://xmlns.com/foaf/o.1/#workInfoHomepage	owl_427
schoolHomepage	property	http://xmlns.com/foaf/o.1/schoolHomepage	owl_157
schoolHomepage	property	http://xmlns.com/foaf/o.1/schoolHomepage	rdf_150
homepage	property	http://xmlns.com/foaf/o.1/homepage	rdf_35
course_Homepage	property	http://ainge.cs.uga.edu/gis/publications#course_Homepage	rdfs_65
homepage	property	http://swrc.org/swrc#homepage	rdf_341
homepage	property	http://lithwww.epfl.ch/teaching/docmul/seance17/people.rdfs#homepage	rdfs_88
homepage	property	http://xmlns.com/foaf/o.1/homepage	owl_326
homepage	property	http://lithwww.epfl.ch/teaching/docmul/seance17/labPeople.rdfs#homepage	rdfs_87
schoolHomepage	object-property	http://xmlns.com/foaf/o.1/schoolHomepage	owl_528
schoolHomepage	object-property	http://xmlns.com/foaf/o.1/schoolHomepage	rdf_87
OntFile	Ontology URL		
daml_157	http://www.cs.umd.edu/projects/plus/DAML/ont/docmnt1.o.daml		
daml_197	http://www.cs.man.ac.uk/~lopatena/certif/certif.daml		
daml_210	http://www.cs.man.ac.uk/~lopatena/certif.daml		
owl_157	http://simile.mit.edu/repository/ontologies/official/foaf.owl		
owl_326	http://www.swed.org.uk/swed/data/swed/organisation_v1.2.owl		
owl_427	http://bblfish.net/work/atom-owl/2004-08-12/Atom.old2.owl		
owl_433	http://daml.umbc.edu/ontologies/webofbelief/Foaf.owl		
owl_435	http://daml.umbc.edu/ontologies/webofbelief/o.81/foaf.owl		
owl_438	http://daml.umbc.edu/ontologies/webofbelief/o.8/foaf-lite.owl		
owl_508	http://daml.umbc.edu/ontologies/webofbelief/o.81/foaf.owl		
owl_511	http://daml.umbc.edu/ontologies/webofbelief/o.8/foaf-lite.owl		
owl_519	http://bblfish.net/work/atom-owl/2004-08-12/Atom.old2.owl		
owl_528	http://www.cs.vu.nl/~pmika/foaf/foaf.owl		
rdf_150	http://gemini.doosh.net/foaf/index.rdf		
rdf_341	http://www.aifb.uni-karlsruhe.de/WBS/meh/mapping/data/swrc1a.rdf		
rdf_35	http://ilrt.org/discovery/2001/05/ical/index.rdf		
rdf_410	http://www.cs.umd.edu/~hendler/2003/MindPeople4-30.rdf		
rdf_51	http://www.atl.external.lmco.com/projects/ontology/ontologies/comsci/csa.rdf		
rdf_87	http://svn.mindswap.org/pychinko/allogtests/mindswapRealized.rdf		
rdfs_65	http://www.arches.uga.edu/~vstaub/GlobalInfoSys/project/ontology/Could_have_been.rdfs		
rdfs_87	http://lithwww.epfl.ch/teaching/docmul/seance17/labpeople.rdfs		
rdfs_88	http://lithwww.epfl.ch/teaching/docmul/seance17/people.rdfs		

The upper half of the Tables 5.1, 5.2 and 5.3 shows actual metadata examples of gender, webpage and homepage respectively. In the first column, the metadata name is provided. In the second column, the type of metadata, whether it's a class or a property, is indicated. The third column shows the full URI of the metadata, and the fourth column shows the ontology file (filenames are arbitrarily given by the prototype tool) in which the metadata has been retrieved. The actual URL of the ontology file is provided in the second column of the lower half of the table, with the filename as pointer.

Table 5.1 shows that gender is defined more as a class than property. Notice that “object property” and “property” definitions (“datatype property” is excluded) implicate class definition when it has Gender Class as its domain. Filenames with asterisk at the end represent this case.

Table 5.2 shows webpage example. Because examples generated from “webpage” keyword were insufficient, another search was conducted using a synonym keyword “homepage”. Table 5.3 shows the result. Majority rules that homepage is a property. Notice that in here, the object property examples do not implicate Homepage Class definitions. The following steps are executed inside the main program to generate this table.

1. Search ontology file(s) that contain the keyword.
2. Retrieve RDF triples of that ontology file(s).
3. Select triples containing the “keyword” in the subject.
4. Of the triples selected in 3, further select triples containing “rdfs:type” ontology element in the predicate.
5. Sort 4 by “class”, “property”, “datatype property” and “object property” ontology element by looking at the object of each triple.

5.3.2 Seeing How Others Define Subsumption Relations

Subsumption relation can be defined using “rdfs:subClassOf.” In the prototype system, some older or experimental versions of “subClassOf” elements are handled. In particular, the following four are handled:

```
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.w3.org/TR/1999/PR-rdf-schema-19990303#subClassOf
http://www.daml.org/2000/10/daml-ont#subClassOf
http://www.daml.org/2001/03/daml+oil#subClassOf
```

Table 5.4: Male Class is mostly defined as subclass of Animal Class

Subclass	Superclass	Ontology URL
Male	Individual	http://www.tt.cs.titech.ac.jp/~fukatani/kadai/yasuda/genealogy.owl
Male	Animal	http://www.daml.org/validator/examples/ont3.daml
female	animal	http://www.cs.man.ac.uk/~horrocks/ESSLLI2003/Ontologies/sane_cows.daml
Male	Animal	http://www.atl.external.lmco.com/projects/ontology/ontologies/animals/animalsB.owl
Male	Animal	http://www.atl.external.lmco.com/projects/ontology/ontologies/animals/animalsA.owl
Female	Animal	http://www.daml.org/2002/06/webont/owl-ex.owl
Male	Animal	http://www.w3.org/2000/10/swap/test/dpo/daml+oil-ex.daml
Female	Animal	http://cvs.sourceforge.net/viewcvs.py/jena/jena2/testing/ontology/daml/daml_oil_2001_03/daml+oil-ex.daml?rev=1.2
Male	Animal	http://www.srdc.metu.edu.tr/~yildiray/example.daml
Female	Animal	http://www.daml.org/2000/12/daml+oil-ex.daml
Male	Animal	http://www.daml.org/validator/examples/ont1.daml
Female	Animal	http://www.cs.man.ac.uk/~horrocks/DAML+OIL/Datatypes/daml+oil+dt-ex.daml
Female	Animal	http://www.cs.vu.nl/~mcaklein/onto/example11.daml
Male	Animal	http://www.daml.org/2000/10/daml-ex.daml
Female	Sex	http://www.csc.fi/kielipankki/puhe/schemas/official/recording.rdfs
MalePerson	Person	http://www.cs.umd.edu/~evren/cmssc828y/hw1/ontology.daml

The sample class subsumption problem observed in Chapter 4 was that of designing superclass of Male and Female Class. Table 5.4 shows the output of the design material generated by the prototype tool upon the keyword input “male”. The first column shows the subclass metadata. In this case, Male and Female Class are the subclass metadata. The second column shows the superclass metadata. And the third column shows the ontology URLs where the metadata are defined. Based on the result presented in the table, it is revealed that many ontologies define Animal Class as the superclass of Male and Female Class.

Notice that a new design direction, which was unthought-of, was presented by the tool through reusing existing ontologies. The initial design possibilities that the author had in mind were Person Class and Gender Class. The following steps are executed inside the main program to generate this table. The first run on steps 1 through 5 return direct superclasses of the searched metadata. To obtain more abundant design material, step 6 may be executed recursively to obtain indirect superclasses of the metadata. Moreover, by replacing the “subject” in step 3 with the “object”, the algorithm will return subclasses of the targeted metadata. As mentioned in Chapter 2, examining subclass level metadata increases the understanding of the domain covered by

the ontology. Applying this algorithm will generate abundant subclass examples which the designer can utilize to find relevant ontology.

1. Search ontology file(s) that contain the keyword.
2. Retrieve RDF triples of that ontology file(s).
3. Select triples containing the "keyword" in the subject.
4. Of the triples selected in 3, further select triples containing "subclassof" ontology element in the predicate.
5. Of the triples obtained in 4, take out the object of the triple as the superclass.
6. To further and use the metadata name as a new keyword to initiate new search starting with step 1.

5.3.3 Seeing How Others Define Object Property

One way of to express a class-to-class relation is through the “object property” ontology element. Defining the object property and specifying the domain and range classes of that property will define the relationship between those two classes. In the prototype tool, the following object property elements are handled:

<http://www.daml.org/2001/03/daml+oil#ObjectProperty>
<http://www.w3.org/2002/07/owl#ObjectProperty>

In Chapter 4, the object property design difficulties observed were those between Person Class and Smoking Class, Person and Pet, and Person and Country. Tables 5.5, 5.6 and 5.7 shows design material generated for each case. The input keywords were “smoking”, “pet” and “country” respectively. The first column shows the domain class of the object property. The second column shows the object property definition. The third column shows the ontology URLs where the object properties are defined.

Table 5.5 shows object property examples for “smoking”. The domain of the object property, however, is not Person Class, but Room Class and HotelRoom Class. Although some hints about the object property definition and the range class are obtained, the examples provided are insufficient.

Table 5.5: Output of input keyword “smoking”

Domain	Object Property	Range	Ontology URL
Room	SmokingOrNon	SmokingAttribute	http://www.atl.external.lmco.com/projects/ontology/ontologies/hotel/hotelB.owl
HotelRoom	hasSmokingPreference	SmokingPreference	http://www.atl.external.lmco.com/projects/ontology/ontologies/hotel/hotelA.owl

Table 5.6: Output of input keyword “pet”

Domain	Object Property	Range	Ontology URL
person	has_pet	animal	http://wonderweb.semanticweb.org/owl/whatSpeciesAmI.rdf
person	has_pet	animal	http://www.cs.man.ac.uk/~horrocks/Slides/Innsbruck-tutorial/people+pets.rdf
person	has_pet	animal	http://www.cs.man.ac.uk/~horrocks/ISWC2003/Tutorial/people+pets.owl.rdf
person	has_pet	animal	http://owl.man.ac.uk/2003/why/latest/ontology.rdf
person	has_pet	animal	http://protege.stanford.edu/plugins/owl/owl-library/people+pets.owl
person	has_pet	animal	http://www.cs.man.ac.uk/~horrocks/Teaching/cs646/Labs/people+pets.daml
person	has_pet	animal	http://www.cs.man.ac.uk/~horrocks/Teaching/cs646/Labs/PeopleAndPets.owl

Table 5.7: Output of input keyword “country”

Domain	Object Property	Range	Ontology URL
ISO3166 DefinedCountry	referencesCountry	Country	http://lstdis.cs.uga.edu/Projects/METEOR-S/MWSAF/Ontologies/Geo/Countries.owl
ISO3166 DefinedCountry	referencesCountry	Country	http://www.w3.org/Consortium/Offices/Presentations/RDFTutorial/rdfs/Countries.owl
PoliticalParty	politicalPartyOfCountry	GeopoliticalArea	http://coli.lili.uni-bielefeld.de/~felix/lehre/ws04_05/ontologischeRessourcen/beispiele/Government.owl
PoliticalParty	politicalPartyOfCountry	GeopoliticalArea	http://reliant.tekknowledge.com/DAML/Government.owl
Locus	locusCountry	Country	http://ontologies.isx.com/onts/isx_basic_onts/isxlocusont.daml
Commodity Online	fromCountry	Country	http://ebusiness.pku.edu.cn/staff/yangyong/Research/DownLoad/CommodityOnline.owl
state	state_located_in_country	country	http://lstdis.cs.uga.edu/proj/traks/ontologies/ontology_3.owl
SymbolicString	internetCountryCode	GeopoliticalArea	http://reliant.tekknowledge.com/DAML/Communications.owl
SymbolicString	internetCountryCode	GeopoliticalArea	http://km.aifb.uni-karlsruhe.de/kaon2/Members/usovmv/myArticle/Communications.owl

Table 5.6 shows object property examples for keyword “pet”. Although in Chapter 4, difficulty in defining person’s preference for pets was the problem, existing definitions for person-pet relation seem to focus on ownership (object property has_pet). Ownership may implicate preference, and designing metadata as has_pet to define preference may be a possible design decision.

Table 5.7 shows object property examples for keyword “country”. In this case, no useful metadata hint is obtained. The domain classes presented are mostly country related classes and none of them are Person Class. Three reasons may be at work. First, not many object properties are defined in existing ontologies. Because the original data is sparse, the resulting design material is also sparse. Secondly, the simple lexical matching algorithm is limited in its ability.

Table 5.8: Output of person class properties in first five ontologies

CATEGORY	ONT1	ONT2	ONT3	ONT4	ONT5
NAME				name	name
	family_name surname	lastName	lastName		
	first_name	firstName	firstName		
			middleName displayName alias nickName		
TITLE		title	title		
HOME PAGE	homepage	homepage		homePage	
	workInfoHomepage workplaceHomepage schoolHomepage weblog				affURL
EMAIL	mbox mbox_sha1sum	email	hasEmail		email
PHONE			hasTelephoneNumber		
		homePhone officePhone cellPhone			
		fax			
		pager			
ADDRESS	(lat) (long) (alt) (lat_long)				
			hasAddress		
		homeAddress officeAddress			
BIRTHDAY /AGE		birthday			
GENDER	gender	gender	gender		
ID	icqChatID aimChatID jabberID msnChatID yahooChatID		identificationNumber ssn		
ORGANI- ZATION	member				
					affname
WORK/ PROJECT	currentProject pastProject				
PUBLI- CATION/ CREATION	publications				
	made				

Table 5.9: Output of person class properties in last five ontologies

CATEGORY	ONT6	ONT7	ONT8	ONT9	ONT10
NAME	name_Person			person.name	name
		familyName			
		givenName			
TITLE	job_title		hasPosition		
HOME PAGE					url
EMAIL	email	emailAddress			email
PHONE		phoneNumber			
ADDRESS					
				address	
	office				
BIRTHDAY /AGE	age				
GENDER	sex				
ID				id	
ORGANIZATION	has_employees		hasMember		
		employer	isBossOf		
WORK/PROJECT	associated_with_project				
PUBLICATION/CREATION	has_publications				

A new algorithm leveraging the structure of the ontology should be implemented. Thirdly, “property” ontology element should also be considered as class-to-class relation design material candidate, since domain and range classes can also be defined using the “property” element.

5.3.4 Comparing Properties Through Table

The last design difficulty discussed in Chapter 4 is the problem of comprehensive property coverage. During the design process, the metadata designer may be unsure if important or major properties of the target subject have been defined. One way to supplement this lack of knowledge is to provide sorted properties of the target subject defined in multiple ontologies. The target subject in this case is defined as class.

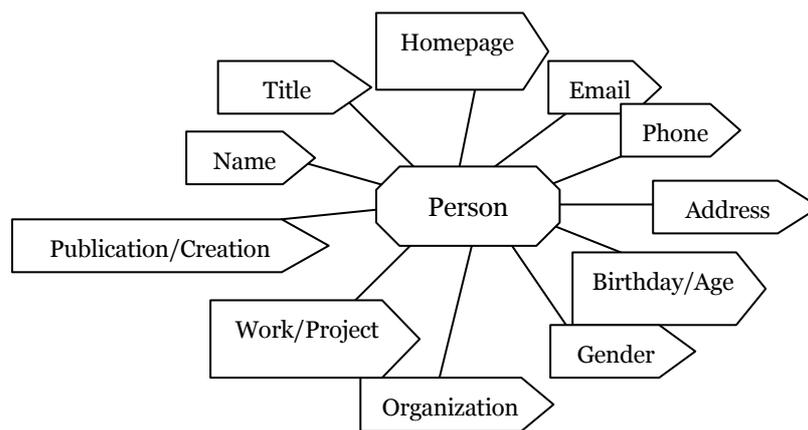


Figure 5.4: Person Class general properties created from existing metadata

Table 5.8 and 5.9 show a sample output of how this design material should be displayed. These tables show Person Class properties defined in multiple ontologies. Notice that similar properties are organized and placed in proximity to each other. Each column starting with the second column from ONT1 to ONT 5 in Table 5.8 and ONT6 to ONT 10 in Table 5.9 are ten actual ontologies containing the Person Class. The properties of the Person Class, that is, properties having Person Class for the domain, are retrieved from each ontology and grouped manually according to the similar properties. Once the similar properties are grouped, the general category for the similar properties is assigned and placed in the first column. For example, for name property,

general category NAME is assigned. As a result, the design material provides what are the general categories, or what are major categories often defined for the given class. In the case of the Person Class, the following twelve property categories were assigned as major properties: Name, Title, Homepage, Email, Phone, Address, Birthday/Age, Gender, ID, Organization, Work/Project, and Publication/Creation. Figure 5.4 shows the twelve person properties. Compare this figure with Figure 4.11 in Chapter 4. Notice that more key properties of Person Class are covered in this figure.

Chapter 6 Conclusion

This research seeks to address the difficulty in metadata design process by reusing existing ontologies as design material. To do this, the following steps were taken. Firstly, existing ontologies on the WWW were analyzed. This was necessary because these ontologies later constitute the design material. Unless the ontologies on the WWW are abundant in variety and in quantity, referring to them would not be useful. Secondly, problems in metadata design process were clarified. In particular, information needs that arise within a restricted setting, which prohibits the human designer from referring to existing ontologies, was investigated to highlight what information support may be useful. Thirdly, based on the understanding obtained from first and second, a step by step metadata design process which incorporates existing ontologies as design material was introduced along with detailed design material that addresses specific design difficulties. A prototype tool that automatically generates the design materials from existing ontologies was implemented.

To see what ontologies actually exist on the WWW, ten ontologies containing Person Class were analyzed with a focus on Person Class related class and property definitions and class hierarchies. Analysis showed that Person Class depictions varied according to ontologies. Even though many shared similar person properties like name and email, different properties (e.g. id, has_employees, knows), superclasses (e.g. Agent, Persistent_Item, SpatialThing), and subclasses (e.g. Employee, Faculty, Nurse) appeared in different ontologies. Moreover, the following were discovered.

- Understanding of the target domain is gained at the subclass level.
- Two kinds of class hierarchy patterns exist: one incorporating outside classes (open) and the other using classes in the same ontology (closed).

To clarify what problems arise in the design process, metadata design experiment was conducted by the author. As data, a social networking service's user input form was selected. A restriction was placed on the author by prohibiting the subject from referring to existing ontologies. This was done

to highlight information needs that arise during the design process. During the design process, the following difficulties were observed.

- Determining whether an item should be designed as a class or a property is difficult.
- Determining what subsumption relation holds between two classes is difficult.
- Designing object property that defines the relationship between two classes is difficult.
- Being confident that most major properties of the target are designed is difficult.

Based on the information needs observed in the experiment, metadata design process that utilizes existing ontologies was modeled, and detailed design material which may aid the designer's decision making was proposed. Also, a prototype tool that generates design material useful to the designer was implemented.

The metadata design process that incorporates existing ontologies leverages the "RIDE" loop, where the human designer retrieves relevant metadata information from existing ontologies to obtain design hints and ideas. As for the design material, for class/property uncertainty problem, both class and property metadata examples are sorted and listed to show which is the majority definition. For defining class relationship problem, existing relationships are displayed. For comprehensive property coverage problem, similar properties in multiples ontologies are displayed in table format.

Acknowledgments

I am indebted to the following people. I am grateful to Professor Toru Ishida at Graduate School of Informatics, Kyoto University, for his discussion, advice, guidance, and support. I express my sincere gratitude to my research adviser, Professor Hiroyuki Yoshihara at Department of Medical Informatics, Kyoto University Hospital and Professor Masahiro Hori at Department of Informatics, Kansai University for their worthy advice and comment.

I also thank all the members of Ishida Laboratory at Kyoto University for their support. Particularly, I deeply appreciate the two geniuses at the Ishida Laboratory, Kosuke Nakatsuka and Yuki Sugimoto for their wisdom, kindness and help.

Finally, I express my gratitude to Japanese Ministry of Education, Culture, Sports, Science and Technology for financially supporting me through Monbukagakusho Scholarship program. Without this scholarship, I would have never had the chance to meet all the wonderful people at Ishida Laboratory and be a part of an exciting research community.

References

- [1] Berners-Lee, T., Hendler, J., and Lassila, O.: The Semantic Web, *Scientific American*, pp. 34-43 (May 2001).
- [2] World Wide Web Consortium (W3C): Metadata and Resource Description, <http://www.w3.org/Metadata/>.
- [3] Gruber, T.: A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, Vol. 5, No. 2, pp. 199-220 (June 1993).
- [4] World Wide Web Consortium (W3C): Resource Description Framework (RDF), <http://www.w3.org/RDF/>.
- [5] World Wide Web Consortium (W3C): Web Ontology Language (OWL), <http://www.w3.org/2004/OWL/>.
- [6] Dublin Core Metadata Initiative (DCMI), <http://dublincore.org/>.
- [7] Kalyanpur, A., Hashmi, N., Golbeck, J. and Parsia, B.: Lifecycle of a Casual Web Ontology Development Process, WWW2004: Workshop on Application Design, Development and Implementation Issues in the Semantic Web, (May 2004).
- [8] Browse SchemaWeb Directory, <http://www.schemaweb.info/schema/BrowseSchema.aspx>.
- [9] Protege OWL Library, <http://protege.stanford.edu/ontologies/ontologies.html>.
- [10] KSL Interactive Ontology Server, <http://www-ksl-svc.stanford.edu:5915/>.
- [11] DAML Ontology Library, <http://www.daml.org/ontologies/>.
- [12] Search SchemaWeb Directory, <http://www.schemaweb.info/search/Search.aspx>.
- [13] Semantic Web Search, <http://www.semanticwebsearch.com/>.
- [14] FOAF (Friend of a Friend), <http://www.foaf-project.org/>.
- [15] Bechhofer, S., et al.: Tackling the Ontology Acquisition Bottleneck: An Experiment in Ontology Re-Engineering, (October 2003).
- [16] FoaF Explorer, <http://xml.mfd-consult.dk/foaf/explorer/>.
- [17] RSS (RDF Site Summary), <http://web.resource.org/rss/1.0/>.
- [18] RDF Site Summary (RSS) 1.0, <http://web.resource.org/rss/1.0/spec>.

- [19] FeedReader, <http://www.feedReader.com/>.
- [20] SharpReader, <http://www.sharpreader.net/>.
- [21] del.icio.us, <http://del.icio.us/>.
- [22] flickr, <http://www.flickr.com/>.
- [23] orkut, <http://www.orkut.com/>.
- [24] OWL DL,
http://www.w3.org/TR/2004/REC-owl-guide-20040210/#term_OWLDL.
- [25] Noy, F. and McGuinness, D. L.: Ontology Development 101: A Guide to Creating Your First Ontology, *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05* (March 2001). Available at:
http://protege.stanford.edu/publications/ontology_development/ontology_101-noy-mcguinness.html.
- [26] Wine Ontology, <http://www.w3.org/TR/owl-guide/wine.rdf>.
- [27] WonderWeb OWL Ontology Validator,
<http://phoebus.cs.man.ac.uk:9999/OWL/Validator>.
- [28] OWL Web Ontology Language Reference,
<http://www.w3.org/TR/owl-ref/#Class>.
- [29] Guarino N. and Welty C.: Evaluating Ontological Decisions with ONTOCLEAN, *Communications of the ACM* Vol. 45(2), pp. 61-65 (February 2002).
- [30] Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing, *International Journal of Human-Computer Studies* 43(5-6), pp.907 (November 1995).
- [32] CIDOC Conceptual Reference Model (CRM), <http://cidoc.ics.forth.gr/>.
- [33] Jena, A Semantic Web Framework for Java, <http://jena.sourceforge.net/>.
- [34] Mukherjee, S., Yang, G. and Ramakrishnan, V.: Automatic Annotation of Content-Rich HTML Documents: Structural and Semantic Analysis, In *Proceedings of the 2nd International Semantic Web Conference* (October 2003).
- [35] Kiryakov A. et al: Semantic Annotation, Indexing and Retrieval, In *Proceedings of the 2nd International Semantic Web Conference* (October 2003).

- [36] Handschuh, S., Staab, S. and Ciravegna, F.: S-CREAM – Semi-automatic CREATION of Metadata, In *Proceedings of EKAW2002*, LNCS, pp. 358-372, (September 2002).
- [37] Cimiano, P., Handschuh, S. and Staab, S.: Towards the Self-Annotating Web, In *Proceedings of the 13th International World Wide Web Conference* (May 2004).
- [38] Kalyanpur, A., Hendler, J., Parsia, B. and Golbeck, J.: SMORE – Semantic Markup, Ontology, and RDF Editor. Available at:
<http://www.mindswap.org/papers/SMORE.pdf>.
- [39] Vargas-Vera, M. et al: MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup, In *Proceedings of the 13th International Conference on Knowledge Engineering and Management* (September 2002).
- [40] Handschuh, S. And Staab, S.: Authoring and Annotation of Web Pages in CREAM, In *Proceedings of the 11th International World Wide Web Conference*, (May 2002).
- [41] Kahan, J., Koivunen, M., Prud’Hommeaux, E. And Swick, R.: Annotea: An Open RDF Infrastructure for Shared Web Annotations, In *Proceedings of the 10th International World Wide Web Conference* (May 2001).
- [42] Fernandez-Lopez M. and Gomez-Perez, A.: Overview and Analysis of Methodologies for Building Ontologies, *The Knowledge Engineering Review* Vol. 17(2), pp. 129-156, (June 2002).
- [43] Farquhar, A., Fikes, R. and Rice, J.: The Ontolingua Server: A Tool for Collaborative Ontology Construction, *International Journal of Human-Computer Studies*, Vol. 46(6), pp.707-727, (June 1997).

Appendix:

A.1 Practical Ontologies Available on the WWW

Listed here are several practical ontologies divided into the following categories: General, Reference, Planning, Enterprise/Business, Health/Medical, Agriculture, and Bio. Note that the content surveyed may not be up to date since it reflects the time of the investigation which is March 2004.

*** General Ontology

SUMO (Suggested Upper Merged Ontology)	
Content	Ontology that promote data interoperability, information search and retrieval, automated inference, etc.
Creator	IEEE Standard Upper Ontology Working Group
Project URL	http://ontology.tekknowledge.com/
Last Update	16-Mar-04
Download	http://reliant.tekknowledge.com/DAML/SUMO.owl
OWL Format	Yes.
Other	<p>Several domain ontologies can be also be downloaded from project website. But in order to download these ontologies, you need to give one's name and affiliation.</p> <p>Computing Services: http://einstein.tekknowledge.com:8080/download/register.jsp?fileType=.zip&fileName=OntologyOfServices.zip</p> <p>Financial: http://einstein.tekknowledge.com:8080/download/register.jsp?fileType=.tar&fileName=FinancialOnt.tar</p> <p>WMD: http://einstein.tekknowledge.com:8080/download/register.jsp?fileType=.zip&fileName=WMD.zip</p> <p>Terrorism: http://einstein.tekknowledge.com:8080/download/register.jsp?fileType=.zip&fileName=terrorism.zip</p> <p>Government: http://einstein.tekknowledge.com:8080/GovernmentOnt/GovernmentOnt_register.jsp?fileType=.zip&fileName=government.zip</p> <p>Economy: http://einstein.tekknowledge.com:8080/download/register.jsp?fileType=.zip&fileName=Economy.zip</p> <p>Geography: http://einstein.tekknowledge.com:8080/download/register.jsp?fileType=.zip&fileName=geography.zip</p> <p>Transportation: http://einstein.tekknowledge.com:8080/download/register.jsp?fileType=.zip&fileName=transportation.zip</p> <p>Atomic Elements: http://einstein.tekknowledge.com:8080/download/register.jsp?fileType=.zip&fileName=elements.zip</p> <p>Biological Virus: http://einstein.tekknowledge.com:8080/download/register.jsp?fileType=.tar&fileName=VirusProteinCellPart.tar</p>

OpenCyc	
Content	General knowledge base. 6,000 concepts and 60,000 assertions.
Creator	OpenCyc.org (Cycorp)
Project URL	http://www.opencyc.org/
Last Update	17-Dec-02
Download	Beta version 0.7 available. http://sourceforge.net/project/showfiles.php?group_id=27274
OWL Format	No.
Other	<p>Mapping ontologies into Cyc (white paper). http://www.cyc.com/doc/white_papers/mapping-ontologies-into-cyc_v31.pdf</p> <p>Diagram of upper cyc ontology. http://www.cyc.com/cycdoc/vocab/upperont-diagram.html</p>

*** General Ontology

DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering)	
Content	Foundational ontology
Creator	Laboratory for Applied Ontology, ISTC-CNR
Project URL	http://www.loa-cnr.it/DOLCE.html
Last Update	May-03
Download	Document. http://wonderweb.semanticweb.org/deliverables/documents/D17.pdf Another document. http://wonderweb.semanticweb.org/deliverables/documents/D18.pdf
OWL Format	No.
Other	WonderWeb Project Page. http://wonderweb.semanticweb.org/index.shtml

SENSUS	
Content	Taxonomies for natural language processing. 70,000 concepts as of May 2000.
Creator	USC/ISI, Natural Language Group
Project URL	http://www.isi.edu/natural-language/resources/sensus.html
Last Update	May-00
Download	No. Instead interactive website. http://mozart.isi.edu:8003/sensus2/
OWL Format	No.
Other	http://www.isi.edu/natural-language/projects/ONTOLOGIES.html

BRICO	
Content	Combination of WordNet, Roget's Thesaurus, and Cyc top level ontology. Works on Framerd platform.
Creator	Originally by MIT Media Lab. Now maintained at http://www.beingmeta.com .
Project URL	http://www.framerd.org/news/brico10.html
Last Update	?
Download	http://www.framerd.org/download.html
OWL Format	No.
Other	http://www.beingmeta.com/

WordNet	
Content	Lexical reference system.
Creator	Cognitive Science Laboratory, Princeton University
Project URL	http://www.cogsci.princeton.edu/~wn/
Last Update	?
Download	http://www.cogsci.princeton.edu/~wn/wn2.o.shtml
OWL Format	No.
Other	

*** Reference Ontology

AKT Reference Ontology (Advanced Knowledge Technologies Reference Ontology)	
Content	Represent the knowledge used in the CS AKTive Portal testbed.
Creator	AKT Consortium (Southampton, Edinburgh, Aberdeen, Sheffield, The Open University)
Project URL	http://www.aktors.org/publications/ontology/
Last Update	7-Mar-03
Download	Support Ontology: http://www.aktors.org/ontology/support Portal Ontology: http://www.aktors.org/ontology/portal Extensions Ontology: http://www.aktors.org/ontology/extension RDF Compatibility Ontology: http://www.aktors.org/ontology/rdfcompat INCA Ontology: http://www.aktors.org/ontology/inca
OWL Format	Yes.
Other	

*** Planning Ontology

CPR (Core Plan Representation)	
Content	Model that supports the representation needs of many different planning systems. I.e. planning ontology.
Creator	Tecknowledge
Project URL	http://projects.tecknowledge.com/CPR2/
Last Update	20-Feb-03
Download	http://projects.tecknowledge.com/CPR2/cprprotege.pon
OWL Format	No.
Other	Mappings from CPR to SUMO. http://projects.tecknowledge.com/CPR2/CPR-SUMO.html

*** Enterprise/Business Ontology

Enterprise Ontology	
Content	Collection of term and definitions relevant to businesses enterprises.
Creator	Artificial Intelligence Applications Institute, University of Edinburgh
Project URL	http://www.ai.ai.ed.ac.uk/project/enterprise/enterprise/ontology.html
Last Update	19-Aug-03
Download	http://www-ksl-svc.stanford.edu:5915/FRAME-EDITOR/UID-215&sid=ANONYMOUS&user-id=ALIEN#ENTERPRISE-ONTOLOGY
OWL Format	Probably no.
Other	Related website: http://www-ksl-svc.stanford.edu:5915/doc/ontology-server-projects.html#enterprise

BPMO (Business Process Management Ontology)	
Content	Ontology to integrate business process design, project management, business performance management, etc.
Creator	Jenz & Partner
Project URL	http://www.jenzundpartner.de/Resources/RE_OSSOnt/re_ossont.htm
Last Update	Feb-04
Download	http://www.jenzundpartner.de/Resources/RE_OSSOntOWL/re_ossontowl.htm
OWL Format	Yes.
Other	Requires Protege to download.

TOVE Ontology	
Content	Ontologies for modeling commercial and public enterprises.
Creator	Enterprise Integration Laboratory, University of Toronto
Project URL	http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html
Last Update	1998
Download	Several links in the project homepage. http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html
OWL Format	No.
Other	

Process Specification Language Ontology	
Content	Describes fundamental concepts of manufacturing processes.
Creator	National Institute of Standards and Technology (NIST), U.S.
Project URL	http://www.mel.nist.gov/psl/ontology.html
Last Update	4-Mar-04
Download	Many links provided in the project website. http://www.mel.nist.gov/psl/ontology.html
OWL Format	No.
Other	

*** Health/Medical Ontology

SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms)	
Content	Universal healthcare terminology. More than 344,000 concepts.
Creator	SNOMED International
Project URL	http://www.snomed.org/
Last Update	25-Mar-04
Download	Brief description in pdf. http://www.snomed.org/snomedct/documents/Jan04_release_print.pdf
OWL Format	No.
Other	To be added to UMLS.

Drug Ontology	
Content	Structured drug knowledge base for software applications.
Creator	Medical Informatics Group, University of Manchester
Project URL	http://www.cs.man.ac.uk/mig/projects/old/drugontology/
Last Update	2002
Download	Bits of examples are provided. http://www.cs.man.ac.uk/mig/projects/old/drugontology/example_mono.html
OWL Format	No.
Other	Related site: http://www.opengalen.org/

MeSH (Medical Subject Headings)	
Content	National Library of Medicine's controlled vocabulary thesaurus.
Creator	U.S. National Library of Medicine
Project URL	http://www.nlm.nih.gov/mesh/meshhome.html
Last Update	5-Dec-03
Download	http://www.nlm.nih.gov/mesh/filelist.html
OWL Format	No.
Other	

FMA (Foundational Model of Anatomy)	
Content	Concepts and relationships necessary for the symbolic modeling of the structure of the human body.
Creator	Structural Informatics Group, University of Washington
Project URL	http://sig.biostr.washington.edu/projects/fm/index.html
Last Update	2004
Download	http://depts.washington.edu/ventures/UW_Technology/Express_Licenses/FMA.php
OWL Format	No.
Other	

NCI Thesaurus (National Cancer Institute Thesaurus)	
Content	Public domain description logic-based terminology produced by the National Cancer Institute.
Creator	MINDSWAP Group, University of Maryland
Project URL	http://www.mindswap.org/2003/CancerOntology/
Last Update	1-Aug-03
Download	http://www.mindswap.org/2003/CancerOntology/nciOncology.owl.gz
OWL Format	Yes.
Other	

*** Agriculture Ontology

AGROVOC	
Content	A multilingual structured and controlled vocabulary designed to cover the terminology of all subject fields of agriculture, forestry, fisheries, food and related domains (e.g. environment).
Creator	Food and Agriculture Organization of the United Nations
Project URL	http://www.fao.org/agrovoc/
Last Update	Oct-03
Download	Send email to fao-agris-caris@fao.org to request download. Or use interactive web interface.
OWL Format	No.
Other	

FAO Terminology (Food and Agriculture Organization Terminology)	
Content	Multilingual (Arabic, Chinese, English, French and Spanish) terminology database related to agriculture etc.
Creator	Food and Agriculture Organization of the United Nations
Project URL	http://www.fao.org/faoterm/
Last Update	?
Download	Interactive website: http://www.fao.org/faoterm/search/index.jsp?lang=EN
OWL Format	No.
Other	

*** BIO Ontology

OBO (Open Biological Ontologies)	
Content	Provides access to controlled vocabularies for shared use across different domains, like MESH and Gene Ontology.
Creator	Hosted by Sourceforge.net
Project URL	http://obo.sourceforge.net/
Last Update	19-Mar-03
Download	http://obo.sourceforge.net/cgi-bin/table.cgi
OWL Format	Some.
Other	39 ontologies listed.

GO (Gene Ontology)	
Content	A controlled vocabulary that can be applied to all organisms even as knowledge of gene and protein roles in cells is accumulating and changing.
Creator	Gene Ontology Consortium
Project URL	http://www.geneontology.org/
Last Update	24-Mar-04
Download	Molecular Function: http://www.geneontology.org/ontology/function.ontology Biological Process: http://www.geneontology.org/ontology/process.ontology Cellular Component: http://www.geneontology.org/ontology/component.ontology
OWL Format	No.
Other	

MGED (Microarray Gene Expression Data)	
Content	Ontology for describing samples used in microarray experiments.
Creator	Microarray Gene Expression Data Society
Project URL	http://mged.sourceforge.net/ontologies/index.php
Last Update	2-Mar-04
Download	http://mged.sourceforge.net/ontologies/MGEDontology.php
OWL Format	No. (But daml file is available.)
Other	

*** BIO Ontology

SO (Sequence Ontology)	
Content	A set of terms used to describe features on a nucleotide or protein sequence.
Creator	Joint effort by the following organizations: WormBase: http://www.wormbase.org/ Berkeley Drosophila Genome Project: http://www.fruitfly.org/ FlyBase: http://www.flybase.org/ Mouse Genome Informatics: http://www.informatics.jax.org/ Sanger Institute: http://www.sanger.ac.uk/
Project URL	http://song.sourceforge.net/
Last Update	23-Mar-04
Download	http://cvs.sourceforge.net/viewcvs.py/song/ontology/so.ontology
OWL Format	No.
Other	

SOFG (Standards and Ontologies for Functional Genomics)	
Content	Integrate ontologies for human and mouse anatomy.
Creator	Joint effort by multiple organizations
Project URL	http://www.sofg.org/
Last Update	3-Mar-04
Download	human: http://www.sofg.org/resources/human.html mouse: http://www.sofg.org/resources/mouse.htm
OWL Format	No.
Other	

PO (Plant Ontology)	
Content	Structured controlled ontologies that describe plant structures and growth/developmental stages.
Creator	Plant Ontology Consortium
Project URL	http://www.plantontology.org/index.html
Last Update	10-Jan-04
Download	http://www.plantontology.org/ontology/index.html
OWL Format	No.
Other	Gramene: http://www.gramene.org/plant_ontology/ TAIR: http://godot.ncgr.org/info/ontologies/ Maize Mapping Project: http://www.maizemap.org/ontology.htm

A.2 Experimental Social Networking Ontology

The following is a social networking ontology created during the metadata design experiment conducted by the author.

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf          = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs        = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl         = "http://www.w3.org/2002/07/owl#"
  xmlns:dc          = "http://purl.org/dc/elements/1.1/"
  xmlns:xsd         = "http://www.w3.org/2001/XMLSchema#"
  xmlns             = "http://www.lab7.kuis.kyoto-u.ac.jp/~cho/ontology/orkut_person#"
  xmlns:orkut       = "http://www.lab7.kuis.kyoto-u.ac.jp/~cho/ontology/orkut_person#"
  xml:base          = "http://www.lab7.kuis.kyoto-u.ac.jp/~cho/ontology/orkut_person#"

  <owl:AnnotationProperty rdf:about="http://purl.org/dc/elements/1.1/description"/>
  <owl:AnnotationProperty rdf:about="http://purl.org/dc/elements/1.1/creator"/>
  <owl:AnnotationProperty rdf:about="http://purl.org/dc/elements/1.1/rights"/>
  <owl:AnnotationProperty rdf:about="http://purl.org/dc/elements/1.1/date"/>

  <owl:Ontology rdf:about="">
    <owl:versionInfo>orkut_merged.owl</owl:versionInfo>
    <rdfs:label>Orkut Person Ontology, OWL DL</rdfs:label>
    <dc:description>This is an experimental ontology for a social networking service website
      called Orkut.com. The website URL is http://www.orkut.com/ . Although the site also provides
      networking services for communities, this ontology only deals with
      information concerning people. Later on, I plan to merge, or create a unified ontology about social
      networking services in general. (eg. friendster.com, gree.jp, mixi.jp, classmates.com,
      cyworld.nate.com, etc.)
      Notable changes: Eliminated owl:Class PersonDescriptor.</dc:description>
    <dc:creator>Heeryon Cho</dc:creator>
    <dc:rights>Copyright (c) 2004 Heeryon Cho</dc:rights>
    <rdfs:comment>Updated on Nov. 24th, 2004.
      Changes on Class:PoliticalView, Class:Religion, Class:Ethnicity.
      Class:Ethnicity->eliminated owl:oneOf since sameAs construct does not work well
      on OWL DL. Ethnicity are redefined as classes (not individuals.)
    </rdfs:comment>
    <rdfs:comment>Unified naming style: that is, capitalization and delimiters.
      For classes, when multiple words must be used, they are connected without
      using any delimiters. eg) PoliticalView, AfricanAmerican
      For properties, first word always starts with lower-case letter, and second word
      is connected to the first word without any delimiters. New words after the first word start with
      upper-case letters. eg) hasGender, hasPoliticalView
      For individuals, first word and all proper nouns begin with capital letters.
      Other words and unimportant words start with lower-case letters.
      Examples of unimportant words are "and" and "of", etc.
      Words are connected using delimiters. eg) New_York, District_of_Columbia, Child_not_at_home
    </rdfs:comment>
    <dc:date>November 25, 2004</dc:date>
  </owl:Ontology>

  <owl:Class rdf:ID="Person">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#hasGender" />
        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:cardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#hasRelationshipStatus" />
          <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:maxCardinality>
          </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasCountry" />
            <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:cardinality>
            </owl:Restriction>
          </rdfs:subClassOf>
        <rdfs:subClassOf>
          <owl:Restriction>
```

```

        <owl:onProperty rdf:resource="#hasPrimaryLanguage" />
        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasOffspringStatus" />
            <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasEthnicity" />
                <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:maxCardinality>
                </owl:Restriction>
            </rdfs:subClassOf>
            <rdfs:subClassOf>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#hasReligion" />
                    <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:maxCardinality>
                    </owl:Restriction>
                </rdfs:subClassOf>
                <rdfs:subClassOf>
                    <owl:Restriction>
                        <owl:onProperty rdf:resource="#hasPoliticalView" />
                        <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:maxCardinality>
                        </owl:Restriction>
                    </rdfs:subClassOf>
                    <rdfs:subClassOf>
                        <owl:Restriction>
                            <owl:onProperty rdf:resource="#hasSexualOrientation" /> <!--ACCESS RESTRICTED-->
                            <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:maxCardinality>
                            </owl:Restriction>
                        </rdfs:subClassOf>
                        <rdfs:subClassOf>
                            <owl:Restriction>
                                <owl:onProperty rdf:resource="#hasPreference" />
                                <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">3</owl:maxCardinality>
                                </owl:Restriction>
                            </rdfs:subClassOf>
                        </owl:Class>
    <!--A list of owl:DatatypeProperties.-->
    <owl:DatatypeProperty rdf:ID="hasName">
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
        <rdfs:comment>Can be any name. Name of person, pet, city, building, etc.</rdfs:comment>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="hasFirstName">
        <rdfs:subPropertyOf rdf:resource="#hasName" />
        <rdfs:domain rdf:resource="#Person" />
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
        <rdfs:label>First name of the person</rdfs:label>
        <rdfs:comment>First name of the person. ? Can a person have multiple first names?</rdfs:comment>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="hasLastName">
        <rdfs:subPropertyOf rdf:resource="#hasName" />
        <rdfs:domain rdf:resource="#Person" />
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
        <rdfs:label>last name of the person</rdfs:label>
        <rdfs:comment>Last name of the person. ? Can a person have multiple last names?</rdfs:comment>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="hasBirthDate">
        <rdfs:domain rdf:resource="#Person" />
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date" />
        <rdfs:comment>How do you express June 26, 1972, using Birth_day and Birth_year?</rdfs:comment>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="hasPostalCode">

```

```

        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>

    <owl:DatatypeProperty rdf:ID="hasZipCode">
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
        <owl:equivalentProperty rdf:resource="#hasPostalCode" />
    </owl:DatatypeProperty>

    <owl:DatatypeProperty rdf:ID="hasWebpage">
        <rdfs:domain rdf:resource="#Person" />
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI" />
    </owl:DatatypeProperty>

    <owl:DatatypeProperty rdf:ID="hasSelfDescription">
        <rdfs:domain rdf:resource="#Person" />
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>

<!--/A list of owl:DatatypeProperties.-->

<!--A person's gender.-->

<owl:Class rdf:ID="Gender">
    <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#Male" />
        <owl:Thing rdf:about="#Female" />
    </owl:oneOf>
</owl:Class>

<Gender rdf:ID="Male" />
<Gender rdf:ID="Female" />

<owl:AllDifferent>
    <owl:distinctMembers rdf:parseType="Collection">
        <orkut:Gender rdf:about="#Male" />
        <orkut:Gender rdf:about="#Female" />
    </owl:distinctMembers>
</owl:AllDifferent>

<owl:ObjectProperty rdf:ID="hasGender">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
    <rdfs:domain rdf:resource="#Person" />
    <rdfs:range rdf:resource="#Gender" />
</owl:ObjectProperty>

<!--/A person's gender.-->

<!--A person's relationship status.-->

<owl:Class rdf:ID="RelationshipStatus">
    <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#Single" />
        <owl:Thing rdf:about="#Married" />
        <owl:Thing rdf:about="#Committed" />
        <owl:Thing rdf:about="#Open_marriage" />
        <owl:Thing rdf:about="#Open_relationship" />
    </owl:oneOf>
</owl:Class>

<RelationshipStatus rdf:ID="Single" />
<RelationshipStatus rdf:ID="Married" />
<RelationshipStatus rdf:ID="Committed" />
<RelationshipStatus rdf:ID="Open_marriage" />
<RelationshipStatus rdf:ID="Open_relationship" />

<owl:AllDifferent>
    <owl:distinctMembers rdf:parseType="Collection">
        <orkut:RelationshipStatus rdf:about="#Single" />
        <orkut:RelationshipStatus rdf:about="#Married" />
        <orkut:RelationshipStatus rdf:about="#Committed" />
        <orkut:RelationshipStatus rdf:about="#Open_marriage" />
        <orkut:RelationshipStatus rdf:about="#Open_relationship" />
    </owl:distinctMembers>
</owl:AllDifferent>

<owl:ObjectProperty rdf:ID="hasRelationshipStatus">
    <rdfs:domain rdf:resource="#Person" />
    <rdfs:range rdf:resource="#RelationshipStatus" />
</owl:ObjectProperty>

```

```

<!--/A person's relationship status.-->

<!--Are you mom or dad? Do you have children?-->

<owl:Class rdf:ID="Parent">
  <rdfs:subClassOf rdf:resource="#Person" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild" />
      <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasOffspringStatus" />
      <owl:allValuesFrom rdf:resource="#YesChild" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Mother" />
    <owl:Class rdf:about="#Father" />
  </owl:unionOf>
</owl:Class>

<owl:Class rdf:ID="Mother">
  <rdfs:subClassOf rdf:resource="#Parent" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasGender" />
      <owl:hasValue rdf:resource="#Female" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Father" />
</owl:Class>

<owl:Class rdf:ID="Father">
  <rdfs:subClassOf rdf:resource="#Parent" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasGender" />
      <owl:hasValue rdf:resource="#Male" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="NonParent">
  <rdfs:comment>Refers to people who do not have offspring.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Person" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasOffspringStatus" />
      <owl:allValuesFrom rdf:resource="#NoChild" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild" />
      <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">0</owl:cardinality>
      </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Parent" />
</owl:Class>

<owl:ObjectProperty rdf:ID="hasChild">
  <rdfs:domain rdf:resource="#Parent" />
  <rdfs:range rdf:resource="#Person" />
</owl:ObjectProperty>

<owl:Class rdf:ID="OffspringStatus">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#YesChild" />
    <owl:Class rdf:about="#NoChild" />
  </owl:unionOf>
</owl:Class>

<owl:Class rdf:ID="YesChild">

```

```

    <rdfs:subClassOf rdf:resource="#OffspringStatus" />
    <owl:oneOf rdf:parseType="Collection">
      <owl:Thing rdf:about="#Child_at_home_full_time"/>
      <owl:Thing rdf:about="#Child_at_home_part_time"/>
      <owl:Thing rdf:about="#Child_not_at_home"/>
    </owl:oneOf>
  </owl:Class>

  <YesChild rdf:ID="Child_at_home_full_time"/>
  <YesChild rdf:ID="Child_at_home_part_time"/>
  <YesChild rdf:ID="Child_not_at_home"/>

  <owl:AllDifferent>
    <owl:distinctMembers rdf:parseType="Collection">
      <orkut:YesChild rdf:about="#Child_at_home_full_time"/>
      <orkut:YesChild rdf:about="#Child_at_home_part_time"/>
      <orkut:YesChild rdf:about="#Child_not_at_home"/>
    </owl:distinctMembers>
  </owl:AllDifferent>

  <owl:Class rdf:ID="NoChild">
    <rdfs:subClassOf rdf:resource="#OffspringStatus" />
    <owl:disjointWith rdf:resource="#YesChild" />
  </owl:Class>

  <owl:ObjectProperty rdf:ID="hasOffspringStatus">
    <rdfs:domain rdf:resource="#Person" />
    <rdfs:range rdf:resource="#OffspringStatus" />
  </owl:ObjectProperty>

  <!--Are you mom or dad? Do you have children?-->

  <!--Ethnicity of a person.-->

  <owl:Class rdf:ID="Ethnicity" />

  <owl:Class rdf:ID="AfricanAmerican">
    <rdfs:subClassOf rdf:resource="#Ethnicity" />
    <owl:disjointWith rdf:resource="#Asian" />
    <owl:disjointWith rdf:resource="#Caucasian" />
    <owl:disjointWith rdf:resource="#EastIndian" />
    <owl:disjointWith rdf:resource="#Hispanic" />
    <owl:disjointWith rdf:resource="#MiddleEastern" />
    <owl:disjointWith rdf:resource="#NativeAmerican" />
    <owl:disjointWith rdf:resource="#PacificIslander" />
    <owl:disjointWith rdf:resource="#OtherEthnicity" />
  </owl:Class>

  <owl:Class rdf:ID="Black">
    <owl:equivalentClass rdf:resource="#AfricanAmerican" />
  </owl:Class>

  <owl:Class rdf:ID="Asian">
    <rdfs:subClassOf rdf:resource="#Ethnicity" />
    <owl:disjointWith rdf:resource="#Caucasian" />
    <owl:disjointWith rdf:resource="#EastIndian" />
    <owl:disjointWith rdf:resource="#Hispanic" />
    <owl:disjointWith rdf:resource="#MiddleEastern" />
    <owl:disjointWith rdf:resource="#NativeAmerican" />
    <owl:disjointWith rdf:resource="#PacificIslander" />
    <owl:disjointWith rdf:resource="#OtherEthnicity" />
  </owl:Class>

  <owl:Class rdf:ID="Caucasian">
    <rdfs:subClassOf rdf:resource="#Ethnicity" />
    <owl:disjointWith rdf:resource="#EastIndian" />
    <owl:disjointWith rdf:resource="#Hispanic" />
    <owl:disjointWith rdf:resource="#MiddleEastern" />
    <owl:disjointWith rdf:resource="#NativeAmerican" />
    <owl:disjointWith rdf:resource="#PacificIslander" />
    <owl:disjointWith rdf:resource="#OtherEthnicity" />
  </owl:Class>

  <owl:Class rdf:ID="White">
    <owl:equivalentClass rdf:resource="#Caucasian" />
  </owl:Class>

  <owl:Class rdf:ID="EastIndian">
    <rdfs:subClassOf rdf:resource="#Ethnicity" />

```

```

    <owl:disjointWith rdf:resource="#Hispanic" />
    <owl:disjointWith rdf:resource="#MiddleEastern" />
    <owl:disjointWith rdf:resource="#NativeAmerican" />
    <owl:disjointWith rdf:resource="#PacificIslander" />
    <owl:disjointWith rdf:resource="#OtherEthnicity" />
</owl:Class>

<owl:Class rdf:ID="Hispanic">
  <rdfs:subClassOf rdf:resource="#Ethnicity" />
  <owl:disjointWith rdf:resource="#MiddleEastern" />
  <owl:disjointWith rdf:resource="#NativeAmerican" />
  <owl:disjointWith rdf:resource="#PacificIslander" />
  <owl:disjointWith rdf:resource="#OtherEthnicity" />
</owl:Class>

<owl:Class rdf:ID="Latino">
  <owl:equivalentClass rdf:resource="#Hispanic" />
</owl:Class>

<owl:Class rdf:ID="MiddleEastern">
  <rdfs:subClassOf rdf:resource="#Ethnicity" />
  <owl:disjointWith rdf:resource="#NativeAmerican" />
  <owl:disjointWith rdf:resource="#PacificIslander" />
  <owl:disjointWith rdf:resource="#OtherEthnicity" />
</owl:Class>

<owl:Class rdf:ID="NativeAmerican">
  <rdfs:subClassOf rdf:resource="#Ethnicity" />
  <owl:disjointWith rdf:resource="#PacificIslander" />
  <owl:disjointWith rdf:resource="#OtherEthnicity" />
</owl:Class>

<owl:Class rdf:ID="American_indian">
  <owl:equivalentClass rdf:resource="#NativeAmerican" />
</owl:Class>

<owl:Class rdf:ID="PacificIslander">
  <rdfs:subClassOf rdf:resource="#Ethnicity" />
  <owl:disjointWith rdf:resource="#OtherEthnicity" />
</owl:Class>

<owl:Class rdf:ID="Multi-ethnic">
  <rdfs:subClassOf rdf:resource="#Ethnicity" />
  <owl:disjointWith rdf:resource="#OtherEthnicity" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasEthnicity" />
      <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">2</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="OtherEthnicity">
  <rdfs:subClassOf rdf:resource="#Ethnicity" />
</owl:Class>

<owl:ObjectProperty rdf:ID="hasEthnicity">
  <rdfs:range rdf:resource="#Ethnicity" />
</owl:ObjectProperty>

<!--/Ethnicity of a person.-->

<!--A person's religion.-->

<owl:Class rdf:ID="Religion"/>

<owl:Class rdf:ID="Agnostic">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#Atheist" />
  <owl:disjointWith rdf:resource="#Buddhist" />
  <owl:disjointWith rdf:resource="#Taoist" />
  <owl:disjointWith rdf:resource="#Christian" />
  <owl:disjointWith rdf:resource="#Hindu" />
  <owl:disjointWith rdf:resource="#Jewish" />
  <owl:disjointWith rdf:resource="#Islam" />
  <owl:disjointWith rdf:resource="#SpiritualButNotReligious" />
  <owl:disjointWith rdf:resource="#ReligiousHumanism" />
  <owl:disjointWith rdf:resource="#OtherReligion" />

```

```

</owl:Class>

<owl:Class rdf:ID="Atheist">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#Buddhist" />
  <owl:disjointWith rdf:resource="#Taoist" />
  <owl:disjointWith rdf:resource="#Christian" />
  <owl:disjointWith rdf:resource="#Hindu" />
  <owl:disjointWith rdf:resource="#Jewish" />
  <owl:disjointWith rdf:resource="#Islam" />
  <owl:disjointWith rdf:resource="#SpiritualButNotReligious" />
  <owl:disjointWith rdf:resource="#ReligiousHumanism" />
  <owl:disjointWith rdf:resource="#OtherReligion" />
</owl:Class>

<owl:Class rdf:ID="Buddhist">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#Taoist" />
  <owl:disjointWith rdf:resource="#Christian" />
  <owl:disjointWith rdf:resource="#Hindu" />
  <owl:disjointWith rdf:resource="#Jewish" />
  <owl:disjointWith rdf:resource="#Islam" />
  <owl:disjointWith rdf:resource="#SpiritualButNotReligious" />
  <owl:disjointWith rdf:resource="#ReligiousHumanism" />
  <owl:disjointWith rdf:resource="#OtherReligion" />
</owl:Class>

<owl:Class rdf:ID="Taoist">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#Christian" />
  <owl:disjointWith rdf:resource="#Hindu" />
  <owl:disjointWith rdf:resource="#Jewish" />
  <owl:disjointWith rdf:resource="#Islam" />
  <owl:disjointWith rdf:resource="#SpiritualButNotReligious" />
  <owl:disjointWith rdf:resource="#ReligiousHumanism" />
  <owl:disjointWith rdf:resource="#OtherReligion" />
</owl:Class>

<owl:Class rdf:ID="Christian">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#Hindu" />
  <owl:disjointWith rdf:resource="#Jewish" />
  <owl:disjointWith rdf:resource="#Islam" />
  <owl:disjointWith rdf:resource="#SpiritualButNotReligious" />
  <owl:disjointWith rdf:resource="#ReligiousHumanism" />
  <owl:disjointWith rdf:resource="#OtherReligion" />
</owl:Class>

<owl:Class rdf:ID="Catholic">
  <rdfs:subClassOf rdf:resource="#Christian" />
  <owl:disjointWith rdf:resource="#LatterDaySaints" />
  <owl:disjointWith rdf:resource="#Protestant" />
  <owl:disjointWith rdf:resource="#OtherTypeOfChristian" />
</owl:Class>

<owl:Class rdf:ID="LatterDaySaints">
  <rdfs:subClassOf rdf:resource="#Christian" />
  <owl:disjointWith rdf:resource="#Protestant" />
  <owl:disjointWith rdf:resource="#OtherTypeOfChristian" />
</owl:Class>

<owl:Class rdf:ID="Mormon">
  <owl:equivalentClass rdf:resource="#LatterDaySaints" />
</owl:Class>

<owl:Class rdf:ID="Protestant">
  <rdfs:subClassOf rdf:resource="#Christian" />
  <owl:disjointWith rdf:resource="#OtherTypeOfChristian" />
</owl:Class>

<owl:Class rdf:ID="OtherTypeOfChristian">
  <rdfs:subClassOf rdf:resource="#Christian" />

<!--The owl:complementOf expression is difficult to use. Kept getting errors when classifier (racer) was run. -->
<!--
  <owl:complementOf rdf:resource="#Catholic" />
  <owl:complementOf rdf:resource="#LatterDaySaints" />
  <owl:complementOf rdf:resource="#Protestant" />
-->

```

```

</owl:Class>
<owl:Class rdf:ID="Hindu">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#Jewish" />
  <owl:disjointWith rdf:resource="#Islam" />
  <owl:disjointWith rdf:resource="#SpiritualButNotReligious" />
  <owl:disjointWith rdf:resource="#ReligiousHumanism" />
  <owl:disjointWith rdf:resource="#OtherReligion" />
</owl:Class>
<owl:Class rdf:ID="Jewish">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#Islam" />
  <owl:disjointWith rdf:resource="#SpiritualButNotReligious" />
  <owl:disjointWith rdf:resource="#ReligiousHumanism" />
  <owl:disjointWith rdf:resource="#OtherReligion" />
</owl:Class>
<owl:Class rdf:ID="Islam">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#SpiritualButNotReligious" />
  <owl:disjointWith rdf:resource="#ReligiousHumanism" />
  <owl:disjointWith rdf:resource="#OtherReligion" />
</owl:Class>
<owl:Class rdf:ID="Muslim">
  <owl:equivalentClass rdf:resource="#Islam" />
  <owl:equivalentClass rdf:resource="#Islamic" />
</owl:Class>
<owl:Class rdf:ID="Islamic">
  <owl:equivalentClass rdf:resource="#Islam" />
  <owl:equivalentClass rdf:resource="#Muslim" />
</owl:Class>
<owl:Class rdf:ID="SpiritualButNotReligious">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#ReligiousHumanism" />
  <owl:disjointWith rdf:resource="#OtherReligion" />
</owl:Class>
<owl:Class rdf:ID="ReligiousHumanism">
  <rdfs:subClassOf rdf:resource="#Religion" />
  <owl:disjointWith rdf:resource="#OtherReligion" />
</owl:Class>
<owl:Class rdf:ID="OtherReligion">
  <rdfs:subClassOf rdf:resource="#Religion" />
</owl:Class>
<owl:ObjectProperty rdf:ID="hasReligion">
  <rdfs:domain rdf:resource="#Person" />
  <rdfs:range rdf:resource="#Religion" />
</owl:ObjectProperty>
<!--/A person's religion.-->
<!--A person's political view.-->
<owl:Class rdf:ID="PoliticalView">
  <rdfs:comment>I've copy-pasted the meaning of Libertarian that I found on the net.
  I was unsure whether Libertarian and Liberal meant the same thing. Here it goes:
  (http://www.angelfire.com/home/socialnarcotics/lp.html)
  Are Libertarians conservative or liberal? Neither. Unlike liberals and conservatives,
  Libertarians advocate a high degree of both personal and economic liberty.
  In a sense, Libertarians "borrow" from both sides to come up with a logical and consistant whole
  - but without the broken promises of Republican and Democratic politicians. For example,
  Libertarians agree with conservatives about freedom in economic matters; in favor of
  lowering taxes, slashing business regulations, and charitable (rather than government) welfare.
  But Libertarians agree with liberals on personal tolerance; in favor of people's right to choose
  personal habits and lifestyles without government intervention.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PoliticalViewWithSentiment">
  <rdfs:subClassOf rdf:resource="#PoliticalView" />
  <rdfs:subClassOf>
    <owl:Restriction>

```

```

                <owl:onProperty rdf:resource="#hasDegreeOfSentiment" />
                <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:cardinality>
                </owl:Restriction>
            </rdfs:subClassOf>
        </owl:Class>

        <owl:Class rdf:ID="RightConservative">
            <rdfs:subClassOf rdf:resource="#PoliticalViewWithSentiment" />
        </owl:Class>

        <owl:Class rdf:ID="Conservative">
            <owl:equivalentClass rdf:resource="#RightConservative" />
        </owl:Class>

        <owl:Class rdf:ID="LeftLiberal">
            <rdfs:subClassOf rdf:resource="#PoliticalViewWithSentiment" />
        </owl:Class>

        <owl:Class rdf:ID="Liberal">
            <owl:equivalentClass rdf:resource="#LeftLiberal" />
            <owl:disjointWith rdf:resource="#Conservative" />
        </owl:Class>

        <owl:Class rdf:ID="Libertarian">
            <rdfs:subClassOf rdf:resource="#PoliticalViewWithSentiment" />
            <owl:disjointWith rdf:resource="#Authoritarian" />
        </owl:Class>

        <owl:Class rdf:ID="Authoritarian">
            <rdfs:subClassOf rdf:resource="#PoliticalViewWithSentiment" />
        </owl:Class>

        <owl:Class rdf:ID="Centrist">
            <rdfs:subClassOf rdf:resource="#PoliticalView" />
        </owl:Class>

        <owl:Class rdf:ID="ViewDepends">
            <rdfs:subClassOf rdf:resource="#PoliticalView" />
        </owl:Class>

        <owl:Class rdf:ID="Ambivalent">
            <owl:equivalentClass rdf:resource="#ViewDepends" />
        </owl:Class>

        <owl:Class rdf:ID="NotPolitical">
            <rdfs:subClassOf rdf:resource="#PoliticalView" />
            <owl:disjointWith rdf:resource="#PoliticalViewWithSentiment" />
            <owl:disjointWith rdf:resource="#Centrist" />
            <owl:disjointWith rdf:resource="#ViewDepends" />
        </owl:Class>

        <owl:Class rdf:ID="DegreeOfSentiment">
            <owl:oneOf rdf:parseType="Collection">
                <owl:Thing rdf:about="#Average_sentiment"/>
                <owl:Thing rdf:about="#Strong_sentiment" />
            </owl:oneOf>
        </owl:Class>

        <DegreeOfSentiment rdf:ID="Average_sentiment" />
        <DegreeOfSentiment rdf:ID="Strong_sentiment" />

        <owl:AllDifferent>
            <owl:distinctMembers rdf:parseType="Collection">
                <orkut:DegreeOfSentiment rdf:about="#Average_sentiment" />
                <orkut:DegreeOfSentiment rdf:about="#Strong_sentiment" />
            </owl:distinctMembers>
        </owl:AllDifferent>

        <owl:ObjectProperty rdf:ID="hasPoliticalView">
            <rdfs:domain rdf:resource="#Person" />
            <rdfs:range rdf:resource="#PoliticalView" />
        </owl:ObjectProperty>

        <owl:ObjectProperty rdf:ID="hasDegreeOfSentiment">
            <rdfs:domain rdf:resource="#PoliticalViewWithSentiment" />
            <rdfs:range rdf:resource="#DegreeOfSentiment" />
        </owl:ObjectProperty>

```

```

<!--/A person's political view.-->

<!--A person's sexual orientation.-->

<owl:Class rdf:ID="SexualOrientation">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Straight" />
    <owl:Thing rdf:about="#Gay" />
    <owl:Thing rdf:about="#Bisexual" />
    <owl:Thing rdf:about="#Bi-curious" />
  </owl:oneOf>
</owl:Class>

<SexualOrientation rdf:ID="Straight" />
<SexualOrientation rdf:ID="Gay" />
<SexualOrientation rdf:ID="Bisexual" />
<SexualOrientation rdf:ID="Bi-curious" />

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <orkut:SexualOrientation rdf:about="#Straight" />
    <orkut:SexualOrientation rdf:about="#Gay" />
    <orkut:SexualOrientation rdf:about="#Bisexual" />
    <orkut:SexualOrientation rdf:about="#Bi-curious" />
  </owl:distinctMembers>
</owl:AllDifferent>

<owl:ObjectProperty rdf:ID="hasSexualOrientation">
  <rdfs:domain rdf:resource="#Person" />
  <rdfs:range rdf:resource="#SexualOrientation" />
</owl:ObjectProperty>

<!--/A person's sexual orientation.-->

<!--A person's fashion sense.-->

<owl:Class rdf:ID="FashionSense">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Alternative_style" />
    <owl:Thing rdf:about="#Casual_style" />
    <owl:Thing rdf:about="#Classic_style" />
    <owl:Thing rdf:about="#Contemporary_style" />
    <owl:Thing rdf:about="#Designer_style" />
    <owl:Thing rdf:about="#Minimal_style" />
    <owl:Thing rdf:about="#Natural_style" />
    <owl:Thing rdf:about="#Outdoorsy_style" />
    <owl:Thing rdf:about="#Smart_style" />
    <owl:Thing rdf:about="#Trendy_style" />
    <owl:Thing rdf:about="#Urban_style" />
  </owl:oneOf>
</owl:Class>

<FashionSense rdf:ID="Alternative_style" />
<FashionSense rdf:ID="Casual_style" />
<FashionSense rdf:ID="Classic_style" />
<FashionSense rdf:ID="Contemporary_style" />
<FashionSense rdf:ID="Designer_style" />
<FashionSense rdf:ID="Minimal_style" />
<FashionSense rdf:ID="Natural_style" />
<FashionSense rdf:ID="Outdoorsy_style" />
<FashionSense rdf:ID="Smart_style" />
<FashionSense rdf:ID="Trendy_style" />
<FashionSense rdf:ID="Urban_style" />

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <orkut:FashionSense rdf:about="#Alternative_style" />
    <orkut:FashionSense rdf:about="#Casual_style" />
    <orkut:FashionSense rdf:about="#Classic_style" />
    <orkut:FashionSense rdf:about="#Contemporary_style" />
    <orkut:FashionSense rdf:about="#Designer_style" />
    <orkut:FashionSense rdf:about="#Minimal_style" />
    <orkut:FashionSense rdf:about="#Natural_style" />
    <orkut:FashionSense rdf:about="#Outdoorsy_style" />
    <orkut:FashionSense rdf:about="#Smart_style" />
    <orkut:FashionSense rdf:about="#Trendy_style" />
    <orkut:FashionSense rdf:about="#Urban_style" />
  </owl:distinctMembers>
</owl:AllDifferent>

```

```

<!--/A person's fashion sense.-->

<!--A person's sense of humor.-->

<owl:Class rdf:ID="SenseOfHumor">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Campy_or_cheesy_humor" />
    <owl:Thing rdf:about="#Goofy_or_slapstic_humor" />
    <owl:Thing rdf:about="#Dry_or_sarcastic_humor" />
    <owl:Thing rdf:about="#Obscure_humor" />
    <owl:Thing rdf:about="#Clever_or_quick_witted_humor" />
    <owl:Thing rdf:about="#Raunchy_humor" />
    <owl:Thing rdf:about="#Friendly_humor" />
  </owl:oneOf>
</owl:Class>

<SenseOfHumor rdf:ID="Campy_or_cheesy_humor" />
<SenseOfHumor rdf:ID="Goofy_or_slapstic_humor" />
<SenseOfHumor rdf:ID="Dry_or_sarcastic_humor" />
<SenseOfHumor rdf:ID="Obscure_humor" />
<SenseOfHumor rdf:ID="Clever_or_quick_witted_humor" />
<SenseOfHumor rdf:ID="Raunchy_humor" />
<SenseOfHumor rdf:ID="Friendly_humor" />

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <orkut:SenseOfHumor rdf:about="#Campy_or_cheesy_humor" />
    <orkut:SenseOfHumor rdf:about="#Goofy_or_slapstic_humor" />
    <orkut:SenseOfHumor rdf:about="#Dry_or_sarcastic_humor" />
    <orkut:SenseOfHumor rdf:about="#Obscure_humor" />
    <orkut:SenseOfHumor rdf:about="#Clever_or_quick_witted_humor" />
    <orkut:SenseOfHumor rdf:about="#Raunchy_humor" />
    <orkut:SenseOfHumor rdf:about="#Friendly_humor" />
  </owl:distinctMembers>
</owl:AllDifferent>

<!--/A person's sense of humor.-->

<!--A person's preference including smoking, drinking, and having pets.-->

<owl:Class rdf:ID="Preference" />

<owl:ObjectProperty rdf:ID="hasPreference">
  <rdfs:domain rdf:resource="#Person" />
  <rdfs:range rdf:resource="#Preference" />
</owl:ObjectProperty>

<!--On smoking.-->

<owl:Class rdf:ID="PreferenceForSmoking">
  <rdfs:subClassOf rdf:resource="#Preference" />
</owl:Class>

<owl:Class rdf:ID="Smoking">
  <rdfs:subClassOf rdf:resource="#PreferenceForSmoking" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasRateOfConsumption" />
      <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="NotSmoking">
  <rdfs:subClassOf rdf:resource="#PreferenceForSmoking" />
  <owl:disjointWith rdf:resource="#Smoking" />
</owl:Class>

<owl:Class rdf:ID="TryingToQuitSmoking">
  <rdfs:subClassOf rdf:resource="#PreferenceForSmoking" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasRateOfConsumption" />
      <owl:hasValue rdf:resource="#Occasionally" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

```

<owl:Class rdf:ID="QuitSmoking">
  <rdfs:subClassOf rdf:resource="#PreferenceForSmoking" />
  <owl:disjointWith rdf:resource="#Smoking"/>
</owl:Class>

<!--/On smoking.-->

<!--On drinking.-->

<owl:Class rdf:ID="PreferenceForDrinking">
  <rdfs:subClassOf rdf:resource="#Preference" />
</owl:Class>

<owl:Class rdf:ID="Drinking">
  <rdfs:subClassOf rdf:resource="#PreferenceForDrinking" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasRateOfConsumption" />
      <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#positiveInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="NotDrinking">
  <rdfs:subClassOf rdf:resource="#PreferenceForDrinking" />
  <owl:disjointWith rdf:resource="#Drinking" />
</owl:Class>

<!--/On drinking.-->

<!--Rate of consumption defined for cigarettes and alcohol.-->

<owl:Class rdf:ID="RateOfConsumption">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Socially" />
    <owl:Thing rdf:about="#Occasionally" />
    <owl:Thing rdf:about="#Regularly" />
    <owl:Thing rdf:about="#Heavily" />
  </owl:oneOf>
</owl:Class>

<RateOfConsumption rdf:ID="Socially" />
<RateOfConsumption rdf:ID="Occasionally" />
<RateOfConsumption rdf:ID="Regularly" />
<RateOfConsumption rdf:ID="Heavily" />

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <orkut:RateOfConsumption rdf:about="#Socially" />
    <orkut:RateOfConsumption rdf:about="#Occasionally" />
    <orkut:RateOfConsumption rdf:about="#Regularly" />
    <orkut:RateOfConsumption rdf:about="#Heavily" />
  </owl:distinctMembers>
</owl:AllDifferent>

<owl:ObjectProperty rdf:ID="hasRateOfConsumption">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Smoking" />
        <owl:Class rdf:about="#Drinking" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#RateOfConsumption" />
</owl:ObjectProperty>

<!--/Rate of consumption defined for cigarettes and alcohol.-->

<!--On having pets.-->

<owl:Class rdf:ID="PreferenceForPet">
  <rdfs:subClassOf rdf:resource="#Preference" />
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#I_love_my_pet" />
    <owl:Thing rdf:about="#I_like_them_at_the_zoo" />
    <owl:Thing rdf:about="#I_like_pet" />
  </owl:oneOf>
</owl:Class>

```

```

        <owl:Thing rdf:about="#I_dont_like_pet" />
    </owl:oneOf>
</owl:Class>

<PreferenceForPet rdf:ID="I_love_my_pet" />
<PreferenceForPet rdf:ID="I_like_them_at_the_zoo" />
<PreferenceForPet rdf:ID="I_like_pet" />
<PreferenceForPet rdf:ID="I_dont_like_pet" />

<owl:AllDifferent>
    <owl:distinctMembers rdf:parseType="Collection">
        <orkut:PreferenceForPet rdf:about="#I_love_my_pets" />
        <orkut:PreferenceForPet rdf:about="#I_like_them_at_the_zoos" />
        <orkut:PreferenceForPet rdf:about="#I_like_pets" />
        <orkut:PreferenceForPet rdf:about="#I_dont_like_pets" />
    </owl:distinctMembers>
</owl:AllDifferent>

<!--/On having pets.-->

<!--/A person's preference including smoking, drinking, and having pets.-->

<!--Information on habitation. Who are you living with?-->

<owl:ObjectProperty rdf:ID="hasHabitationStatus">
    <rdfs:domain rdf:resource="#Person" />
    <rdfs:range rdf:resource="#HabitationStatus" />
</owl:ObjectProperty>

<owl:Class rdf:ID="HabitationStatus">
    <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#Living_alone" />
        <owl:Thing rdf:about="#Living_with_kids" />
        <owl:Thing rdf:about="#Living_with_roommates" />
        <owl:Thing rdf:about="#Living_with_parents" />
        <owl:Thing rdf:about="#Living_with_partner" />
        <owl:Thing rdf:about="#Friends_visit_often" />
        <owl:Thing rdf:about="#Living_with_pets" />
        <owl:Thing rdf:about="#Party_every_night" />
    </owl:oneOf>
</owl:Class>

<HabitationStatus rdf:ID="Living_alone" />
<HabitationStatus rdf:ID="Living_with_kids" />
<HabitationStatus rdf:ID="Living_with_roommates" />
<HabitationStatus rdf:ID="Living_with_parents" />
<HabitationStatus rdf:ID="Living_with_partner" />
<HabitationStatus rdf:ID="Friends_visit_often" />
<HabitationStatus rdf:ID="Living_with_pets" />
<HabitationStatus rdf:ID="Party_every_night" />

<owl:AllDifferent>
    <owl:distinctMembers rdf:parseType="Collection">
        <orkut:HabitationStatus rdf:about="#Living_alone" />
        <orkut:HabitationStatus rdf:about="#Living_with_kids" />
        <orkut:HabitationStatus rdf:about="#Living_with_roommates" />
        <orkut:HabitationStatus rdf:about="#Living_with_parents" />
        <orkut:HabitationStatus rdf:about="#Living_with_partner" />
        <orkut:HabitationStatus rdf:about="#Friends_visit_often" />
        <orkut:HabitationStatus rdf:about="#Living_with_pets" />
        <orkut:HabitationStatus rdf:about="#Party_every_night" />
    </owl:distinctMembers>
</owl:AllDifferent>

<!--/Information on habitation. Who are you living with?-->

<!--Preference for social networking.-->

<owl:ObjectProperty rdf:ID="hasInterest">
    <rdfs:domain rdf:resource="#Person" />
    <rdfs:range rdf:resource="#TypeOfSocialNetworking" />
</owl:ObjectProperty>

<owl:Class rdf:ID="TypeOfSocialNetworking">
    <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#Meeting_friends" />
        <owl:Thing rdf:about="#Meeting_activity_partners" />
        <owl:Thing rdf:about="#Business_networking" />
        <owl:Thing rdf:about="#Dating_men" />
    </owl:oneOf>
</owl:Class>

```

```

        <owl:Thing rdf:about="#Dating_men_and_women" />
        <owl:Thing rdf:about="#Dating_women" />
    </owl:oneOf>
</owl:Class>

<TypeOfSocialNetworking rdf:ID="Meeting_friends" />
<TypeOfSocialNetworking rdf:ID="Meeting_activity_partners" />
<TypeOfSocialNetworking rdf:ID="Business_networking" />
<TypeOfSocialNetworking rdf:ID="Dating_men" />
<TypeOfSocialNetworking rdf:ID="Dating_men_and_women" />
<TypeOfSocialNetworking rdf:ID="Dating_women" />

<owl:AllDifferent>
    <owl:distinctMembers rdf:parseType="Collection">
        <orkut:TypeOfSocialNetworking rdf:about="#Meeting_friends" />
        <orkut:TypeOfSocialNetworking rdf:about="#Meeting_activity_partners" />
        <orkut:TypeOfSocialNetworking rdf:about="#Business_networking" />
        <orkut:TypeOfSocialNetworking rdf:about="#Dating_men" />
        <orkut:TypeOfSocialNetworking rdf:about="#Dating_men_and_women" />
        <orkut:TypeOfSocialNetworking rdf:about="#Dating_women" />
    </owl:distinctMembers>
</owl:AllDifferent>

<!--/Preference for social networking.-->

<!--Country that a person lives in. Could also be interpreted as nationality of a person.-->

<owl:ObjectProperty rdf:ID="hasCountry">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
    <rdfs:domain rdf:resource="#Person" />
    <rdfs:range rdf:resource="#Country" />
</owl:ObjectProperty>

<owl:Class rdf:ID="Country">
<rdfs:comment>A list of countries defined using owl:oneOf. It is known that the number of countries in the
world ranges between 189 and 266. Countries listed in the orkut site is 227. Although this list is not
exhaustive, we define the country as an exhaustive class consisting of exactly 227 individuals (= countries).
For more information on how many countries there are, see
http://ask.yahoo.com/ask/20010105.html</rdfs:comment>
<owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#United_States" />
    <owl:Thing rdf:about="#Canada" />
    <owl:Thing rdf:about="#Afghanistan" />
    <owl:Thing rdf:about="#Albania" />
    <owl:Thing rdf:about="#Algeria" />
    <owl:Thing rdf:about="#American_Samoa" />
    <owl:Thing rdf:about="#Andorra" />
    <owl:Thing rdf:about="#Angola" />
    <owl:Thing rdf:about="#Anguilla" />
    <owl:Thing rdf:about="#Antigua_and_Barbuda" />
    <owl:Thing rdf:about="#Argentina" />
    <owl:Thing rdf:about="#Armenia" />
    <owl:Thing rdf:about="#Ascension_Island" />
    <owl:Thing rdf:about="#Australia" />
    <owl:Thing rdf:about="#Austria" />
    <owl:Thing rdf:about="#Azerbaijan" />
    <owl:Thing rdf:about="#Bahamas" />
    <owl:Thing rdf:about="#Bahrain" />
    <owl:Thing rdf:about="#Bangladesh" />
    <owl:Thing rdf:about="#Barbados" />
    <owl:Thing rdf:about="#Belarus" />
    <owl:Thing rdf:about="#Belgium" />
    <owl:Thing rdf:about="#Belize" />
    <owl:Thing rdf:about="#Benin" />
    <owl:Thing rdf:about="#Bermuda" />
    <owl:Thing rdf:about="#Bhutan" />
    <owl:Thing rdf:about="#Bolivia" />
    <owl:Thing rdf:about="#Bosnia_and_Herzegovina" />
    <owl:Thing rdf:about="#Botswana" />
    <owl:Thing rdf:about="#Brazil" />
    <owl:Thing rdf:about="#British_Indian_Ocean_Territory" />
    <owl:Thing rdf:about="#Brunei_Darussalam" />
    <owl:Thing rdf:about="#Bulgaria" />
    <owl:Thing rdf:about="#Burkina_Faso" />
    <owl:Thing rdf:about="#Burundi" />
    <owl:Thing rdf:about="#Cambodia" />
    <owl:Thing rdf:about="#Cameroon" />
    <owl:Thing rdf:about="#Cape_Verde" />
    <owl:Thing rdf:about="#Cayman_Islands" />

```

```

<owl:Thing rdf:about="#Central_African_Republic" />
<owl:Thing rdf:about="#Chad" />
<owl:Thing rdf:about="#Chile" />
<owl:Thing rdf:about="#China" />
<owl:Thing rdf:about="#Colombia" />
<owl:Thing rdf:about="#Comoros" />
<owl:Thing rdf:about="#Congo" />
<owl:Thing rdf:about="#Cook_Islands" />
<owl:Thing rdf:about="#Costa_Rica" />
<owl:Thing rdf:about="#Cote_D_Ivoire" />
<owl:Thing rdf:about="#Croatia" />
<owl:Thing rdf:about="#Cuba" />
<owl:Thing rdf:about="#Cyprus" />
<owl:Thing rdf:about="#Czech_Republic" />
<owl:Thing rdf:about="#Denmark" />
<owl:Thing rdf:about="#Djibouti" />
<owl:Thing rdf:about="#Dominica" />
<owl:Thing rdf:about="#Dominican_Republic" />
<owl:Thing rdf:about="#Ecuador" />
<owl:Thing rdf:about="#Egypt" />
<owl:Thing rdf:about="#El_Salvador" />
<owl:Thing rdf:about="#Equatorial_Guinea" />
<owl:Thing rdf:about="#Eritrea" />
<owl:Thing rdf:about="#Estonia" />
<owl:Thing rdf:about="#Ethiopia" />
<owl:Thing rdf:about="#Falkland_Islands" />
<owl:Thing rdf:about="#Faroe_Islands" />
<owl:Thing rdf:about="#Federated_States_of_Micronesia" />
<owl:Thing rdf:about="#Fiji" />
<owl:Thing rdf:about="#Finland" />
<owl:Thing rdf:about="#France" />
<owl:Thing rdf:about="#French_Guiana" />
<owl:Thing rdf:about="#French_Polynesia" />
<owl:Thing rdf:about="#Gabon" />
<owl:Thing rdf:about="#Georgia" />
<owl:Thing rdf:about="#Germany" />
<owl:Thing rdf:about="#Ghana" />
<owl:Thing rdf:about="#Gibraltar" />
<owl:Thing rdf:about="#Greece" />
<owl:Thing rdf:about="#Greenland" />
<owl:Thing rdf:about="#Grenada" />
<owl:Thing rdf:about="#Guadeloupe" />
<owl:Thing rdf:about="#Guam" />
<owl:Thing rdf:about="#Guatemala" />
<owl:Thing rdf:about="#Guinea" />
<owl:Thing rdf:about="#Guinea_Bissau" />
<owl:Thing rdf:about="#Guyana" />
<owl:Thing rdf:about="#Haiti" />
<owl:Thing rdf:about="#Honduras" />
<owl:Thing rdf:about="#Hungary" />
<owl:Thing rdf:about="#Iceland" />
<owl:Thing rdf:about="#India" />
<owl:Thing rdf:about="#Indonesia" />
<owl:Thing rdf:about="#Iran" />
<owl:Thing rdf:about="#Iraq" />
<owl:Thing rdf:about="#Ireland" />
<owl:Thing rdf:about="#Isle_of_Man" />
<owl:Thing rdf:about="#Israel" />
<owl:Thing rdf:about="#Italy" />
<owl:Thing rdf:about="#Jamaica" />
<owl:Thing rdf:about="#Japan" />
<owl:Thing rdf:about="#Jordan" />
<owl:Thing rdf:about="#Kazakhstan" />
<owl:Thing rdf:about="#Kenya" />
<owl:Thing rdf:about="#Kiribati" />
<owl:Thing rdf:about="#Korea_(Peoples_Republic_of)" />
<owl:Thing rdf:about="#Korea_(Republic_of)" />
<owl:Thing rdf:about="#Kuwait" />
<owl:Thing rdf:about="#Kyrgyzstan" />
<owl:Thing rdf:about="#Laos" />
<owl:Thing rdf:about="#Latvia" />
<owl:Thing rdf:about="#Lebanon" />
<owl:Thing rdf:about="#Lesotho" />
<owl:Thing rdf:about="#Liberia" />
<owl:Thing rdf:about="#Libya" />
<owl:Thing rdf:about="#Liechtenstein" />
<owl:Thing rdf:about="#Lithuania" />
<owl:Thing rdf:about="#Luxembourg" />
<owl:Thing rdf:about="#Macau" />

```

```

<owl:Thing rdf:about="#Macedonia" />
<owl:Thing rdf:about="#Madagascar" />
<owl:Thing rdf:about="#Malawi" />
<owl:Thing rdf:about="#Malaysia" />
<owl:Thing rdf:about="#Maldives" />
<owl:Thing rdf:about="#Mali" />
<owl:Thing rdf:about="#Malta" />
<owl:Thing rdf:about="#Marshall_Islands" />
<owl:Thing rdf:about="#Martinique" />
<owl:Thing rdf:about="#Mauritius" />
<owl:Thing rdf:about="#Mayotte" />
<owl:Thing rdf:about="#Mexico" />
<owl:Thing rdf:about="#Moldova" />
<owl:Thing rdf:about="#Monaco" />
<owl:Thing rdf:about="#Mongolia" />
<owl:Thing rdf:about="#Montenegro" />
<owl:Thing rdf:about="#Montserrat" />
<owl:Thing rdf:about="#Morocco" />
<owl:Thing rdf:about="#Mozambique" />
<owl:Thing rdf:about="#Myanmar" />
<owl:Thing rdf:about="#Namibia" />
<owl:Thing rdf:about="#Nauru" />
<owl:Thing rdf:about="#Nepal" />
<owl:Thing rdf:about="#Netherlands" />
<owl:Thing rdf:about="#Netherlands_Antilles" />
<owl:Thing rdf:about="#New_Caledonia" />
<owl:Thing rdf:about="#New_Zealand" />
<owl:Thing rdf:about="#Nicaragua" />
<owl:Thing rdf:about="#Niger" />
<owl:Thing rdf:about="#Nigeria" />
<owl:Thing rdf:about="#Niue" />
<owl:Thing rdf:about="#Norfolk_Island" />
<owl:Thing rdf:about="#Northern_Mariana_Islands" />
<owl:Thing rdf:about="#Norway" />
<owl:Thing rdf:about="#Oman" />
<owl:Thing rdf:about="#Pakistan" />
<owl:Thing rdf:about="#Palau" />
<owl:Thing rdf:about="#Panama" />
<owl:Thing rdf:about="#Papua_New_Guinea" />
<owl:Thing rdf:about="#Paraguay" />
<owl:Thing rdf:about="#Peru" />
<owl:Thing rdf:about="#Philippines" />
<owl:Thing rdf:about="#Pitcairn" />
<owl:Thing rdf:about="#Poland" />
<owl:Thing rdf:about="#Portugal" />
<owl:Thing rdf:about="#Puerto_Rico" />
<owl:Thing rdf:about="#Qatar" />
<owl:Thing rdf:about="#Reunion" />
<owl:Thing rdf:about="#Romania" />
<owl:Thing rdf:about="#Russian_Federation" />
<owl:Thing rdf:about="#Rwanda" />
<owl:Thing rdf:about="#Saint_Vincent_and_the_Grenadines" />
<owl:Thing rdf:about="#San_Marino" />
<owl:Thing rdf:about="#Sao_Tome_and_Principe" />
<owl:Thing rdf:about="#Saudi_Arabia" />
<owl:Thing rdf:about="#Senegal" />
<owl:Thing rdf:about="#Serbia" />
<owl:Thing rdf:about="#Seychelles" />
<owl:Thing rdf:about="#Sierra_Leone" />
<owl:Thing rdf:about="#Singapore" />
<owl:Thing rdf:about="#Slovakia" />
<owl:Thing rdf:about="#Slovenia" />
<owl:Thing rdf:about="#Solomon_Islands" />
<owl:Thing rdf:about="#Somalia" />
<owl:Thing rdf:about="#South_Africa" />
<owl:Thing rdf:about="#South_Georgia" />
<owl:Thing rdf:about="#Spain" />
<owl:Thing rdf:about="#Sri_Lanka" />
<owl:Thing rdf:about="#St_Kitts_and_Nevis" />
<owl:Thing rdf:about="#St_Lucia" />
<owl:Thing rdf:about="#St_Pierre_and_Miquelon" />
<owl:Thing rdf:about="#Sudan" />
<owl:Thing rdf:about="#Suriname" />
<owl:Thing rdf:about="#Swaziland" />
<owl:Thing rdf:about="#Sweden" />
<owl:Thing rdf:about="#Switzerland" />
<owl:Thing rdf:about="#Syrian_Arab_Republic" />
<owl:Thing rdf:about="#Taiwan" />
<owl:Thing rdf:about="#Tajikistan" />

```

```

    <owl:Thing rdf:about="#Tanzania" />
    <owl:Thing rdf:about="#Thailand" />
    <owl:Thing rdf:about="#The_Gambia" />
    <owl:Thing rdf:about="#Togo" />
    <owl:Thing rdf:about="#Tokelau" />
    <owl:Thing rdf:about="#Tonga" />
    <owl:Thing rdf:about="#Trinidad_and_Tobago" />
    <owl:Thing rdf:about="#Tunisia" />
    <owl:Thing rdf:about="#Turkey" />
    <owl:Thing rdf:about="#Turkmenistan" />
    <owl:Thing rdf:about="#Turks_and_Caicos_Islands" />
    <owl:Thing rdf:about="#Tuvalu" />
    <owl:Thing rdf:about="#Uganda" />
    <owl:Thing rdf:about="#Ukraine" />
    <owl:Thing rdf:about="#United_Arab_Emirates" />
    <owl:Thing rdf:about="#United_Kingdom" />
    <owl:Thing rdf:about="#Uruguay" />
    <owl:Thing rdf:about="#Uzbekistan" />
    <owl:Thing rdf:about="#Vanuatu" />
    <owl:Thing rdf:about="#Venezuela" />
    <owl:Thing rdf:about="#Viet_Nam" />
    <owl:Thing rdf:about="#Virgin_Islands_(U.K.)" />
    <owl:Thing rdf:about="#Virgin_Islands_(U.S.)" />
    <owl:Thing rdf:about="#Wallis_and_Futuna_Islands" />
    <owl:Thing rdf:about="#Western_Samoa" />
    <owl:Thing rdf:about="#Yemen" />
    <owl:Thing rdf:about="#Yugoslavia" />
    <owl:Thing rdf:about="#Zaire" />
    <owl:Thing rdf:about="#Zambia" />
    <owl:Thing rdf:about="#Zimbabwe" />
  </owl:oneOf>
</owl:Class>

<Country rdf:ID="United_States" />
<Country rdf:ID="Canada" />
<Country rdf:ID="Afghanistan" />
<Country rdf:ID="Albania" />
<Country rdf:ID="Algeria" />
<Country rdf:ID="American_Samoa" />
<Country rdf:ID="Andorra" />
<Country rdf:ID="Angola" />
<Country rdf:ID="Anguilla" />
<Country rdf:ID="Antigua_and_Barbuda" />
<Country rdf:ID="Argentina" />
<Country rdf:ID="Armenia" />
<Country rdf:ID="Ascension_Island" />
<Country rdf:ID="Australia" />
<Country rdf:ID="Austria" />
<Country rdf:ID="Azerbaijan" />
<Country rdf:ID="Bahamas" />
<Country rdf:ID="Bahrain" />
<Country rdf:ID="Bangladesh" />
<Country rdf:ID="Barbados" />
<Country rdf:ID="Belarus" />
<Country rdf:ID="Belgium" />
<Country rdf:ID="Belize" />
<Country rdf:ID="Benin" />
<Country rdf:ID="Bermuda" />
<Country rdf:ID="Bhutan" />
<Country rdf:ID="Bolivia" />
<Country rdf:ID="Bosnia_and_Herzegovina" />
<Country rdf:ID="Botswana" />
<Country rdf:ID="Brazil" />
<Country rdf:ID="British_Indian_Ocean_Territory" />
<Country rdf:ID="Brunei_Darussalam" />
<Country rdf:ID="Bulgaria" />
<Country rdf:ID="Burkina_Faso" />
<Country rdf:ID="Burundi" />
<Country rdf:ID="Cambodia" />
<Country rdf:ID="Cameroon" />
<Country rdf:ID="Cape_Verde" />
<Country rdf:ID="Cayman_Islands" />
<Country rdf:ID="Central_African_Republic" />
<Country rdf:ID="Chad" />
<Country rdf:ID="Chile" />
<Country rdf:ID="China" />
<Country rdf:ID="Colombia" />
<Country rdf:ID="Comoros" />
<Country rdf:ID="Congo" />

```

<Country rdf:ID="Cook_Islands" />
 <Country rdf:ID="Costa_Rica" />
 <Country rdf:ID="Cote_D_Ivoire" />
 <Country rdf:ID="Croatia" />
 <Country rdf:ID="Cuba" />
 <Country rdf:ID="Cyprus" />
 <Country rdf:ID="Czech_Republic" />
 <Country rdf:ID="Denmark" />
 <Country rdf:ID="Djibouti" />
 <Country rdf:ID="Dominica" />
 <Country rdf:ID="Dominican_Republic" />
 <Country rdf:ID="Ecuador" />
 <Country rdf:ID="Egypt" />
 <Country rdf:ID="El_Salvador" />
 <Country rdf:ID="Equatorial_Guinea" />
 <Country rdf:ID="Eritrea" />
 <Country rdf:ID="Estonia" />
 <Country rdf:ID="Ethiopia" />
 <Country rdf:ID="Falkland_Islands" />
 <Country rdf:ID="Faroe_Islands" />
 <Country rdf:ID="Federated_States_of_Micronesia" />
 <Country rdf:ID="Fiji" />
 <Country rdf:ID="Finland" />
 <Country rdf:ID="France" />
 <Country rdf:ID="French_Guiana" />
 <Country rdf:ID="French_Polynesia" />
 <Country rdf:ID="Gabon" />
 <Country rdf:ID="Georgia" />
 <Country rdf:ID="Germany" />
 <Country rdf:ID="Ghana" />
 <Country rdf:ID="Gibraltar" />
 <Country rdf:ID="Greece" />
 <Country rdf:ID="Greenland" />
 <Country rdf:ID="Grenada" />
 <Country rdf:ID="Guadeloupe" />
 <Country rdf:ID="Guam" />
 <Country rdf:ID="Guatemala" />
 <Country rdf:ID="Guinea" />
 <Country rdf:ID="Guinea_Bissau" />
 <Country rdf:ID="Guyana" />
 <Country rdf:ID="Haiti" />
 <Country rdf:ID="Honduras" />
 <Country rdf:ID="Hungary" />
 <Country rdf:ID="Iceland" />
 <Country rdf:ID="India" />
 <Country rdf:ID="Indonesia" />
 <Country rdf:ID="Iran" />
 <Country rdf:ID="Iraq" />
 <Country rdf:ID="Ireland" />
 <Country rdf:ID="Isle_of_Man" />
 <Country rdf:ID="Israel" />
 <Country rdf:ID="Italy" />
 <Country rdf:ID="Jamaica" />
 <Country rdf:ID="Japan" />
 <Country rdf:ID="Jordan" />
 <Country rdf:ID="Kazakhstan" />
 <Country rdf:ID="Kenya" />
 <Country rdf:ID="Kiribati" />
 <Country rdf:ID="Korea_(Peoples_Republic_of)" />
 <Country rdf:ID="Korea_(Republic_of)" />
 <Country rdf:ID="Kuwait" />
 <Country rdf:ID="Kyrgyzstan" />
 <Country rdf:ID="Laos" />
 <Country rdf:ID="Latvia" />
 <Country rdf:ID="Lebanon" />
 <Country rdf:ID="Lesotho" />
 <Country rdf:ID="Liberia" />
 <Country rdf:ID="Libya" />
 <Country rdf:ID="Liechtenstein" />
 <Country rdf:ID="Lithuania" />
 <Country rdf:ID="Luxembourg" />
 <Country rdf:ID="Macau" />
 <Country rdf:ID="Macedonia" />
 <Country rdf:ID="Madagascar" />
 <Country rdf:ID="Malawi" />
 <Country rdf:ID="Malaysia" />
 <Country rdf:ID="Maldives" />
 <Country rdf:ID="Mali" />
 <Country rdf:ID="Malta" />

<Country rdf:ID="Marshall_Islands" />
 <Country rdf:ID="Martinique" />
 <Country rdf:ID="Mauritius" />
 <Country rdf:ID="Mayotte" />
 <Country rdf:ID="Mexico" />
 <Country rdf:ID="Moldova" />
 <Country rdf:ID="Monaco" />
 <Country rdf:ID="Mongolia" />
 <Country rdf:ID="Montenegro" />
 <Country rdf:ID="Montserrat" />
 <Country rdf:ID="Morocco" />
 <Country rdf:ID="Mozambique" />
 <Country rdf:ID="Myanmar" />
 <Country rdf:ID="Namibia" />
 <Country rdf:ID="Nauru" />
 <Country rdf:ID="Nepal" />
 <Country rdf:ID="Netherlands" />
 <Country rdf:ID="Netherlands_Antilles" />
 <Country rdf:ID="New_Caledonia" />
 <Country rdf:ID="New_Zealand" />
 <Country rdf:ID="Nicaragua" />
 <Country rdf:ID="Niger" />
 <Country rdf:ID="Nigeria" />
 <Country rdf:ID="Niue" />
 <Country rdf:ID="Norfolk_Island" />
 <Country rdf:ID="Northern_Mariana_Islands" />
 <Country rdf:ID="Norway" />
 <Country rdf:ID="Oman" />
 <Country rdf:ID="Pakistan" />
 <Country rdf:ID="Palau" />
 <Country rdf:ID="Panama" />
 <Country rdf:ID="Papua_New_Guinea" />
 <Country rdf:ID="Paraguay" />
 <Country rdf:ID="Peru" />
 <Country rdf:ID="Philippines" />
 <Country rdf:ID="Pitcairn" />
 <Country rdf:ID="Poland" />
 <Country rdf:ID="Portugal" />
 <Country rdf:ID="Puerto_Rico" />
 <Country rdf:ID="Qatar" />
 <Country rdf:ID="Reunion" />
 <Country rdf:ID="Romania" />
 <Country rdf:ID="Russian_Federation" />
 <Country rdf:ID="Rwanda" />
 <Country rdf:ID="Saint_Vincent_and_the_Grenadines" />
 <Country rdf:ID="San_Marino" />
 <Country rdf:ID="Sao_Tome_and_Principe" />
 <Country rdf:ID="Saudi_Arabia" />
 <Country rdf:ID="Senegal" />
 <Country rdf:ID="Serbia" />
 <Country rdf:ID="Seychelles" />
 <Country rdf:ID="Sierra_Leone" />
 <Country rdf:ID="Singapore" />
 <Country rdf:ID="Slovakia" />
 <Country rdf:ID="Slovenia" />
 <Country rdf:ID="Solomon_Islands" />
 <Country rdf:ID="Somalia" />
 <Country rdf:ID="South_Africa" />
 <Country rdf:ID="South_Georgia" />
 <Country rdf:ID="Spain" />
 <Country rdf:ID="Sri_Lanka" />
 <Country rdf:ID="St_Kitts_and_Nevis" />
 <Country rdf:ID="St_Lucia" />
 <Country rdf:ID="St_Pierre_and_Miquelon" />
 <Country rdf:ID="Sudan" />
 <Country rdf:ID="Suriname" />
 <Country rdf:ID="Swaziland" />
 <Country rdf:ID="Sweden" />
 <Country rdf:ID="Switzerland" />
 <Country rdf:ID="Syrian_Arab_Republic" />
 <Country rdf:ID="Taiwan" />
 <Country rdf:ID="Tajikistan" />
 <Country rdf:ID="Tanzania" />
 <Country rdf:ID="Thailand" />
 <Country rdf:ID="The_Gambia" />
 <Country rdf:ID="Togo" />
 <Country rdf:ID="Tokelau" />
 <Country rdf:ID="Tonga" />
 <Country rdf:ID="Trinidad_and_Tobago" />

```

<Country rdf:ID="Tunisia" />
<Country rdf:ID="Turkey" />
<Country rdf:ID="Turkmenistan" />
<Country rdf:ID="Turks_and_Caicos_Islands" />
<Country rdf:ID="Tuvalu" />
<Country rdf:ID="Uganda" />
<Country rdf:ID="Ukraine" />
<Country rdf:ID="United_Arab_Emirates" />
<Country rdf:ID="United_Kingdom" />
<Country rdf:ID="Uruguay" />
<Country rdf:ID="Uzbekistan" />
<Country rdf:ID="Vanuatu" />
<Country rdf:ID="Venezuela" />
<Country rdf:ID="Viet_Nam" />
<Country rdf:ID="Virgin_Islands_(U.K.)" />
<Country rdf:ID="Virgin_Islands_(U.S.)" />
<Country rdf:ID="Wallis_and_Futuna_Islands" />
<Country rdf:ID="Western_Samoa" />
<Country rdf:ID="Yemen" />
<Country rdf:ID="Yugoslavia" />
<Country rdf:ID="Zaire" />
<Country rdf:ID="Zambia" />
<Country rdf:ID="Zimbabwe" />

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <orkut:Country rdf:about="United_States" />
    <orkut:Country rdf:about="Canada" />
    <orkut:Country rdf:about="Afghanistan" />
    <orkut:Country rdf:about="Albania" />
    <orkut:Country rdf:about="Algeria" />
    <orkut:Country rdf:about="American_Samoa" />
    <orkut:Country rdf:about="Andorra" />
    <orkut:Country rdf:about="Angola" />
    <orkut:Country rdf:about="Anguilla" />
    <orkut:Country rdf:about="Antigua_and_Barbuda" />
    <orkut:Country rdf:about="Argentina" />
    <orkut:Country rdf:about="Armenia" />
    <orkut:Country rdf:about="Ascension_Island" />
    <orkut:Country rdf:about="Australia" />
    <orkut:Country rdf:about="Austria" />
    <orkut:Country rdf:about="Azerbaijan" />
    <orkut:Country rdf:about="Bahamas" />
    <orkut:Country rdf:about="Bahrain" />
    <orkut:Country rdf:about="Bangladesh" />
    <orkut:Country rdf:about="Barbados" />
    <orkut:Country rdf:about="Belarus" />
    <orkut:Country rdf:about="Belgium" />
    <orkut:Country rdf:about="Belize" />
    <orkut:Country rdf:about="Benin" />
    <orkut:Country rdf:about="Bermuda" />
    <orkut:Country rdf:about="Bhutan" />
    <orkut:Country rdf:about="Bolivia" />
    <orkut:Country rdf:about="Bosnia_and_Herzegovina" />
    <orkut:Country rdf:about="Botswana" />
    <orkut:Country rdf:about="Brazil" />
    <orkut:Country rdf:about="British_Indian_Ocean_Territory" />
    <orkut:Country rdf:about="Brunei_Darussalam" />
    <orkut:Country rdf:about="Bulgaria" />
    <orkut:Country rdf:about="Burkina_Faso" />
    <orkut:Country rdf:about="Burundi" />
    <orkut:Country rdf:about="Cambodia" />
    <orkut:Country rdf:about="Cameroon" />
    <orkut:Country rdf:about="Cape_Verde" />
    <orkut:Country rdf:about="Cayman_Islands" />
    <orkut:Country rdf:about="Central_African_Republic" />
    <orkut:Country rdf:about="Chad" />
    <orkut:Country rdf:about="Chile" />
    <orkut:Country rdf:about="China" />
    <orkut:Country rdf:about="Colombia" />
    <orkut:Country rdf:about="Comoros" />
    <orkut:Country rdf:about="Congo" />
    <orkut:Country rdf:about="Cook_Islands" />
    <orkut:Country rdf:about="Costa_Rica" />
    <orkut:Country rdf:about="Cote_D_Ivoire" />
    <orkut:Country rdf:about="Croatia" />
    <orkut:Country rdf:about="Cuba" />
    <orkut:Country rdf:about="Cyprus" />
    <orkut:Country rdf:about="Czech_Republic" />
  
```

<orkut:Country rdf:about="Denmark" />
 <orkut:Country rdf:about="Djibouti" />
 <orkut:Country rdf:about="Dominica" />
 <orkut:Country rdf:about="Dominican_Republic" />
 <orkut:Country rdf:about="Ecuador" />
 <orkut:Country rdf:about="Egypt" />
 <orkut:Country rdf:about="El_Salvador" />
 <orkut:Country rdf:about="Equatorial_Guinea" />
 <orkut:Country rdf:about="Eritrea" />
 <orkut:Country rdf:about="Estonia" />
 <orkut:Country rdf:about="Ethiopia" />
 <orkut:Country rdf:about="Falkland_Islands" />
 <orkut:Country rdf:about="Faroe_Islands" />
 <orkut:Country rdf:about="Federated_States_of_Micronesia" />
 <orkut:Country rdf:about="Fiji" />
 <orkut:Country rdf:about="Finland" />
 <orkut:Country rdf:about="France" />
 <orkut:Country rdf:about="French_Guiana" />
 <orkut:Country rdf:about="French_Polynesia" />
 <orkut:Country rdf:about="Gabon" />
 <orkut:Country rdf:about="Georgia" />
 <orkut:Country rdf:about="Germany" />
 <orkut:Country rdf:about="Ghana" />
 <orkut:Country rdf:about="Gibraltar" />
 <orkut:Country rdf:about="Greece" />
 <orkut:Country rdf:about="Greenland" />
 <orkut:Country rdf:about="Grenada" />
 <orkut:Country rdf:about="Guadeloupe" />
 <orkut:Country rdf:about="Guam" />
 <orkut:Country rdf:about="Guatemala" />
 <orkut:Country rdf:about="Guinea" />
 <orkut:Country rdf:about="Guinea_Bissau" />
 <orkut:Country rdf:about="Guyana" />
 <orkut:Country rdf:about="Haiti" />
 <orkut:Country rdf:about="Honduras" />
 <orkut:Country rdf:about="Hungary" />
 <orkut:Country rdf:about="Iceland" />
 <orkut:Country rdf:about="India" />
 <orkut:Country rdf:about="Indonesia" />
 <orkut:Country rdf:about="Iran" />
 <orkut:Country rdf:about="Iraq" />
 <orkut:Country rdf:about="Ireland" />
 <orkut:Country rdf:about="Isle_of_Man" />
 <orkut:Country rdf:about="Israel" />
 <orkut:Country rdf:about="Italy" />
 <orkut:Country rdf:about="Jamaica" />
 <orkut:Country rdf:about="Japan" />
 <orkut:Country rdf:about="Jordan" />
 <orkut:Country rdf:about="Kazakhstan" />
 <orkut:Country rdf:about="Kenya" />
 <orkut:Country rdf:about="Kiribati" />
 <orkut:Country rdf:about="Korea_(Peoples_Republic_of)" />
 <orkut:Country rdf:about="Korea_(Republic_of)" />
 <orkut:Country rdf:about="Kuwait" />
 <orkut:Country rdf:about="Kyrgyzstan" />
 <orkut:Country rdf:about="Laos" />
 <orkut:Country rdf:about="Latvia" />
 <orkut:Country rdf:about="Lebanon" />
 <orkut:Country rdf:about="Lesotho" />
 <orkut:Country rdf:about="Liberia" />
 <orkut:Country rdf:about="Libya" />
 <orkut:Country rdf:about="Liechtenstein" />
 <orkut:Country rdf:about="Lithuania" />
 <orkut:Country rdf:about="Luxembourg" />
 <orkut:Country rdf:about="Macau" />
 <orkut:Country rdf:about="Macedonia" />
 <orkut:Country rdf:about="Madagascar" />
 <orkut:Country rdf:about="Malawi" />
 <orkut:Country rdf:about="Malaysia" />
 <orkut:Country rdf:about="Maldives" />
 <orkut:Country rdf:about="Mali" />
 <orkut:Country rdf:about="Malta" />
 <orkut:Country rdf:about="Marshall_Islands" />
 <orkut:Country rdf:about="Martinique" />
 <orkut:Country rdf:about="Mauritius" />
 <orkut:Country rdf:about="Mayotte" />
 <orkut:Country rdf:about="Mexico" />
 <orkut:Country rdf:about="Moldova" />
 <orkut:Country rdf:about="Monaco" />

<orkut:Country rdf:about="Mongolia" />
 <orkut:Country rdf:about="Montenegro" />
 <orkut:Country rdf:about="Montserrat" />
 <orkut:Country rdf:about="Morocco" />
 <orkut:Country rdf:about="Mozambique" />
 <orkut:Country rdf:about="Myanmar" />
 <orkut:Country rdf:about="Namibia" />
 <orkut:Country rdf:about="Nauru" />
 <orkut:Country rdf:about="Nepal" />
 <orkut:Country rdf:about="Netherlands" />
 <orkut:Country rdf:about="Netherlands_Antilles" />
 <orkut:Country rdf:about="New_Caledonia" />
 <orkut:Country rdf:about="New_Zealand" />
 <orkut:Country rdf:about="Nicaragua" />
 <orkut:Country rdf:about="Niger" />
 <orkut:Country rdf:about="Nigeria" />
 <orkut:Country rdf:about="Niue" />
 <orkut:Country rdf:about="Norfolk_Island" />
 <orkut:Country rdf:about="Northern_Mariana_Islands" />
 <orkut:Country rdf:about="Norway" />
 <orkut:Country rdf:about="Oman" />
 <orkut:Country rdf:about="Pakistan" />
 <orkut:Country rdf:about="Palau" />
 <orkut:Country rdf:about="Panama" />
 <orkut:Country rdf:about="Papua_New_Guinea" />
 <orkut:Country rdf:about="Paraguay" />
 <orkut:Country rdf:about="Peru" />
 <orkut:Country rdf:about="Philippines" />
 <orkut:Country rdf:about="Pitcairn" />
 <orkut:Country rdf:about="Poland" />
 <orkut:Country rdf:about="Portugal" />
 <orkut:Country rdf:about="Puerto_Rico" />
 <orkut:Country rdf:about="Qatar" />
 <orkut:Country rdf:about="Reunion" />
 <orkut:Country rdf:about="Romania" />
 <orkut:Country rdf:about="Russian_Federation" />
 <orkut:Country rdf:about="Rwanda" />
 <orkut:Country rdf:about="Saint_Vincent_and_the_Grenadines" />
 <orkut:Country rdf:about="San_Marino" />
 <orkut:Country rdf:about="Sao_Tome_and_Principe" />
 <orkut:Country rdf:about="Saudi_Arabia" />
 <orkut:Country rdf:about="Senegal" />
 <orkut:Country rdf:about="Serbia" />
 <orkut:Country rdf:about="Seychelles" />
 <orkut:Country rdf:about="Sierra_Leone" />
 <orkut:Country rdf:about="Singapore" />
 <orkut:Country rdf:about="Slovakia" />
 <orkut:Country rdf:about="Slovenia" />
 <orkut:Country rdf:about="Solomon_Islands" />
 <orkut:Country rdf:about="Somalia" />
 <orkut:Country rdf:about="South_Africa" />
 <orkut:Country rdf:about="South_Georgia" />
 <orkut:Country rdf:about="Spain" />
 <orkut:Country rdf:about="Sri_Lanka" />
 <orkut:Country rdf:about="St_Kitts_and_Nevis" />
 <orkut:Country rdf:about="St_Lucia" />
 <orkut:Country rdf:about="St_Pierre_and_Miquelon" />
 <orkut:Country rdf:about="Sudan" />
 <orkut:Country rdf:about="Suriname" />
 <orkut:Country rdf:about="Swaziland" />
 <orkut:Country rdf:about="Sweden" />
 <orkut:Country rdf:about="Switzerland" />
 <orkut:Country rdf:about="Syrian_Arab_Republic" />
 <orkut:Country rdf:about="Taiwan" />
 <orkut:Country rdf:about="Tajikistan" />
 <orkut:Country rdf:about="Tanzania" />
 <orkut:Country rdf:about="Thailand" />
 <orkut:Country rdf:about="The_Gambia" />
 <orkut:Country rdf:about="Togo" />
 <orkut:Country rdf:about="Tokelau" />
 <orkut:Country rdf:about="Tonga" />
 <orkut:Country rdf:about="Trinidad_and_Tobago" />
 <orkut:Country rdf:about="Tunisia" />
 <orkut:Country rdf:about="Turkey" />
 <orkut:Country rdf:about="Turkmenistan" />
 <orkut:Country rdf:about="Turks_and_Caicos_Islands" />
 <orkut:Country rdf:about="Tuvalu" />
 <orkut:Country rdf:about="Uganda" />
 <orkut:Country rdf:about="Ukraine" />

```

<orkut:Country rdf:about="United_Arab_Emirates" />
<orkut:Country rdf:about="United_Kingdom" />
<orkut:Country rdf:about="Uruguay" />
<orkut:Country rdf:about="Uzbekistan" />
<orkut:Country rdf:about="Vanuatu" />
<orkut:Country rdf:about="Venezuela" />
<orkut:Country rdf:about="Viet_Nam" />
<orkut:Country rdf:about="Virgin_Islands_(U.K.)" />
<orkut:Country rdf:about="Virgin_Islands_(U.S.)" />
<orkut:Country rdf:about="Wallis_and_Futuna_Islands" />
<orkut:Country rdf:about="Western_Samoa" />
<orkut:Country rdf:about="Yemen" />
<orkut:Country rdf:about="Yugoslavia" />
<orkut:Country rdf:about="Zaire" />
<orkut:Country rdf:about="Zambia" />
<orkut:Country rdf:about="Zimbabwe" />
</owl:distinctMembers>
</owl:AllDifferent>

<!--/Country that a person lives in. Could also be interpreted as nationality of a person.-->

<!--Primary language= native language= mother tongue of a person.-->

<owl:ObjectProperty rdf:ID="hasPrimaryLanguage">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
  <rdfs:domain rdf:resource="#Person" />
  <rdfs:range rdf:resource="#PrimaryLanguage" />
</owl:ObjectProperty>

<owl:Class rdf:ID="PrimaryLanguage"> <!--This list contains 115 languages.-->
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#English" />
    <owl:Thing rdf:about="#Afrikaans" />
    <owl:Thing rdf:about="#Ainu" />
    <owl:Thing rdf:about="#Albanian" />
    <owl:Thing rdf:about="#Amo" />
    <owl:Thing rdf:about="#Aymara" />
    <owl:Thing rdf:about="#Azerbaijani" />
    <owl:Thing rdf:about="#Azeri" />
    <owl:Thing rdf:about="#Bahasa" />
    <owl:Thing rdf:about="#Basque" />
    <owl:Thing rdf:about="#Batak" />
    <owl:Thing rdf:about="#Batak_toba" />
    <owl:Thing rdf:about="#Bengali" />
    <owl:Thing rdf:about="#Bihari" />
    <owl:Thing rdf:about="#Bosnian" />
    <owl:Thing rdf:about="#Breton" />
    <owl:Thing rdf:about="#Catalan" />
    <owl:Thing rdf:about="#Cherokee" />
    <owl:Thing rdf:about="#Cornish" />
    <owl:Thing rdf:about="#Corsican" />
    <owl:Thing rdf:about="#Cree" />
    <owl:Thing rdf:about="#Croatian" />
    <owl:Thing rdf:about="#Czech" />
    <owl:Thing rdf:about="#Danish" />
    <owl:Thing rdf:about="#Dutch" />
    <owl:Thing rdf:about="#Edo" />
    <owl:Thing rdf:about="#Esperanto" />
    <owl:Thing rdf:about="#Estonian" />
    <owl:Thing rdf:about="#Faroese" />
    <owl:Thing rdf:about="#Fijian" />
    <owl:Thing rdf:about="#Filipino" />
    <owl:Thing rdf:about="#Finnish" />
    <owl:Thing rdf:about="#French" />
    <owl:Thing rdf:about="#Frisian" />
    <owl:Thing rdf:about="#Gaelic" />
    <owl:Thing rdf:about="#Galician" />
    <owl:Thing rdf:about="#Gascon" />
    <owl:Thing rdf:about="#German" />
    <owl:Thing rdf:about="#Guarani" />
    <owl:Thing rdf:about="#Hanunoo" />
    <owl:Thing rdf:about="#Hausa" />
    <owl:Thing rdf:about="#Hawaiian" />
    <owl:Thing rdf:about="#Hmong" />
    <owl:Thing rdf:about="#Hopi" />
    <owl:Thing rdf:about="#Hungarian" />
    <owl:Thing rdf:about="#Ibibio" />
    <owl:Thing rdf:about="#Icelandic" />
    <owl:Thing rdf:about="#Indonesian" />
  
```

```

<owl:Thing rdf:about="#Ingush" />
<owl:Thing rdf:about="#Interlingua" />
<owl:Thing rdf:about="#Inuktitut" />
<owl:Thing rdf:about="#Inupiaq" />
<owl:Thing rdf:about="#Irish" />
<owl:Thing rdf:about="#Italian" />
<owl:Thing rdf:about="#Javanese" />
<owl:Thing rdf:about="#Kannada" />
<owl:Thing rdf:about="#Kanuri" />
<owl:Thing rdf:about="#Karelian" />
<owl:Thing rdf:about="#Khasi" />
<owl:Thing rdf:about="#Kirghiz" />
<owl:Thing rdf:about="#Komi" />
<owl:Thing rdf:about="#Kurdish" />
<owl:Thing rdf:about="#Lapp" />
<owl:Thing rdf:about="#Latin" />
<owl:Thing rdf:about="#Latvian" />
<owl:Thing rdf:about="#Lithuanian" />
<owl:Thing rdf:about="#Lushootseed" />
<owl:Thing rdf:about="#Luxemburgish" />
<owl:Thing rdf:about="#Macedonian" />
<owl:Thing rdf:about="#Malay" />
<owl:Thing rdf:about="#Maltese" />
<owl:Thing rdf:about="#Mari" />
<owl:Thing rdf:about="#Naga" />
<owl:Thing rdf:about="#Navajo" />
<owl:Thing rdf:about="#Norwegian" />
<owl:Thing rdf:about="#Occitan" />
<owl:Thing rdf:about="#Oriya" />
<owl:Thing rdf:about="#Polish" />
<owl:Thing rdf:about="#Portuguese" />
<owl:Thing rdf:about="#Provencal" />
<owl:Thing rdf:about="#Prussian" />
<owl:Thing rdf:about="#Punjabi" />
<owl:Thing rdf:about="#Quechua" />
<owl:Thing rdf:about="#Romanian" />
<owl:Thing rdf:about="#Romany" />
<owl:Thing rdf:about="#Sami" />
<owl:Thing rdf:about="#Scots_Gaelic" />
<owl:Thing rdf:about="#Serbo-Croatian" />
<owl:Thing rdf:about="#Sesotho" />
<owl:Thing rdf:about="#Shona" />
<owl:Thing rdf:about="#Sinhalese" />
<owl:Thing rdf:about="#Slovak" />
<owl:Thing rdf:about="#Slovenian" />
<owl:Thing rdf:about="#Somali" />
<owl:Thing rdf:about="#Spanish" />
<owl:Thing rdf:about="#Sudanese" />
<owl:Thing rdf:about="#Swahili" />
<owl:Thing rdf:about="#Swedish" />
<owl:Thing rdf:about="#Tagalog" />
<owl:Thing rdf:about="#Tagbanwa" />
<owl:Thing rdf:about="#Tahitian" />
<owl:Thing rdf:about="#Tajik" />
<owl:Thing rdf:about="#Tamazight" />
<owl:Thing rdf:about="#Turkish" />
<owl:Thing rdf:about="#Turkmen" />
<owl:Thing rdf:about="#Twi" />
<owl:Thing rdf:about="#Udmurt" />
<owl:Thing rdf:about="#Uighur" />
<owl:Thing rdf:about="#Uzbek" />
<owl:Thing rdf:about="#Vietnamese" />
<owl:Thing rdf:about="#Welsh" />
<owl:Thing rdf:about="#Xhosa" />
<owl:Thing rdf:about="#Yi" />
<owl:Thing rdf:about="#Yoruba" />
<owl:Thing rdf:about="#Zulu" />
</owl:oneOf>
</owl:Class>
<owl:Class rdf:ID="MotherTongue">
  <owl:equivalentClass>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <PrimaryLanguage rdf:ID="English" />
        <PrimaryLanguage rdf:ID="Afrikaans" />
        <PrimaryLanguage rdf:ID="Ainu" />
        <PrimaryLanguage rdf:ID="Albanian" />
        <PrimaryLanguage rdf:ID="Amo" />
      </owl:oneOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

<PrimaryLanguage rdf:ID="Aymara" />
 <PrimaryLanguage rdf:ID="Azerbaijani" />
 <PrimaryLanguage rdf:ID="Azeri" />
 <PrimaryLanguage rdf:ID="Bahasa" />
 <PrimaryLanguage rdf:ID="Basque" />
 <PrimaryLanguage rdf:ID="Batak" />
 <PrimaryLanguage rdf:ID="Batak_toba" />
 <PrimaryLanguage rdf:ID="Bengali" />
 <PrimaryLanguage rdf:ID="Bihari" />
 <PrimaryLanguage rdf:ID="Bosnian" />
 <PrimaryLanguage rdf:ID="Breton" />
 <PrimaryLanguage rdf:ID="Catalan" />
 <PrimaryLanguage rdf:ID="Cherokee" />
 <PrimaryLanguage rdf:ID="Cornish" />
 <PrimaryLanguage rdf:ID="Corsican" />
 <PrimaryLanguage rdf:ID="Cree" />
 <PrimaryLanguage rdf:ID="Croatian" />
 <PrimaryLanguage rdf:ID="Czech" />
 <PrimaryLanguage rdf:ID="Danish" />
 <PrimaryLanguage rdf:ID="Dutch" />
 <PrimaryLanguage rdf:ID="Edo" />
 <PrimaryLanguage rdf:ID="Esperanto" />
 <PrimaryLanguage rdf:ID="Estonian" />
 <PrimaryLanguage rdf:ID="Faroese" />
 <PrimaryLanguage rdf:ID="Fijian" />
 <PrimaryLanguage rdf:ID="Filipino" />
 <PrimaryLanguage rdf:ID="Finnish" />
 <PrimaryLanguage rdf:ID="French" />
 <PrimaryLanguage rdf:ID="Frisian" />
 <PrimaryLanguage rdf:ID="Gaelic" />
 <PrimaryLanguage rdf:ID="Galician" />
 <PrimaryLanguage rdf:ID="Gascon" />
 <PrimaryLanguage rdf:ID="German" />
 <PrimaryLanguage rdf:ID="Guarani" />
 <PrimaryLanguage rdf:ID="Hanunoo" />
 <PrimaryLanguage rdf:ID="Hausa" />
 <PrimaryLanguage rdf:ID="Hawaiian" />
 <PrimaryLanguage rdf:ID="Hmong" />
 <PrimaryLanguage rdf:ID="Hopi" />
 <PrimaryLanguage rdf:ID="Hungarian" />
 <PrimaryLanguage rdf:ID="Ibibio" />
 <PrimaryLanguage rdf:ID="Icelandic" />
 <PrimaryLanguage rdf:ID="Indonesian" />
 <PrimaryLanguage rdf:ID="Ingush" />
 <PrimaryLanguage rdf:ID="Interlingua" />
 <PrimaryLanguage rdf:ID="Inuktitut" />
 <PrimaryLanguage rdf:ID="Inupiaq" />
 <PrimaryLanguage rdf:ID="Irish" />
 <PrimaryLanguage rdf:ID="Italian" />
 <PrimaryLanguage rdf:ID="Javanese" />
 <PrimaryLanguage rdf:ID="Kannada" />
 <PrimaryLanguage rdf:ID="Kanuri" />
 <PrimaryLanguage rdf:ID="Karelian" />
 <PrimaryLanguage rdf:ID="Khasi" />
 <PrimaryLanguage rdf:ID="Kirghiz" />
 <PrimaryLanguage rdf:ID="Komi" />
 <PrimaryLanguage rdf:ID="Kurdish" />
 <PrimaryLanguage rdf:ID="Lapp" />
 <PrimaryLanguage rdf:ID="Latin" />
 <PrimaryLanguage rdf:ID="Latvian" />
 <PrimaryLanguage rdf:ID="Lithuanian" />
 <PrimaryLanguage rdf:ID="Lushootseed" />
 <PrimaryLanguage rdf:ID="Luxemburgish" />
 <PrimaryLanguage rdf:ID="Macedonian" />
 <PrimaryLanguage rdf:ID="Malay" />
 <PrimaryLanguage rdf:ID="Maltese" />
 <PrimaryLanguage rdf:ID="Mari" />
 <PrimaryLanguage rdf:ID="Naga" />
 <PrimaryLanguage rdf:ID="Navajo" />
 <PrimaryLanguage rdf:ID="Norwegian" />
 <PrimaryLanguage rdf:ID="Occitan" />
 <PrimaryLanguage rdf:ID="Oriya" />
 <PrimaryLanguage rdf:ID="Polish" />
 <PrimaryLanguage rdf:ID="Portuguese" />
 <PrimaryLanguage rdf:ID="Provencal" />
 <PrimaryLanguage rdf:ID="Prussian" />
 <PrimaryLanguage rdf:ID="Punjabi" />
 <PrimaryLanguage rdf:ID="Quechua" />
 <PrimaryLanguage rdf:ID="Romanian" />

```

    <PrimaryLanguage rdf:ID="Romany" />
    <PrimaryLanguage rdf:ID="Sami" />
    <PrimaryLanguage rdf:ID="Scots_Gaelic" />
    <PrimaryLanguage rdf:ID="Serbo-Croatian" />
    <PrimaryLanguage rdf:ID="Sesotho" />
    <PrimaryLanguage rdf:ID="Shona" />
    <PrimaryLanguage rdf:ID="Sinhalese" />
    <PrimaryLanguage rdf:ID="Slovak" />
    <PrimaryLanguage rdf:ID="Slovenian" />
    <PrimaryLanguage rdf:ID="Somali" />
    <PrimaryLanguage rdf:ID="Spanish" />
    <PrimaryLanguage rdf:ID="Sudanese" />
    <PrimaryLanguage rdf:ID="Swahili" />
    <PrimaryLanguage rdf:ID="Swedish" />
    <PrimaryLanguage rdf:ID="Tagalog" />
    <PrimaryLanguage rdf:ID="Tagbanwa" />
    <PrimaryLanguage rdf:ID="Tahitian" />
    <PrimaryLanguage rdf:ID="Tajik" />
    <PrimaryLanguage rdf:ID="Tamazight" />
    <PrimaryLanguage rdf:ID="Turkish" />
    <PrimaryLanguage rdf:ID="Turkmen" />
    <PrimaryLanguage rdf:ID="Twi" />
    <PrimaryLanguage rdf:ID="Udmurt" />
    <PrimaryLanguage rdf:ID="Uighur" />
    <PrimaryLanguage rdf:ID="Uzbek" />
    <PrimaryLanguage rdf:ID="Vietnamese" />
    <PrimaryLanguage rdf:ID="Welsh" />
    <PrimaryLanguage rdf:ID="Xhosa" />
    <PrimaryLanguage rdf:ID="Yi" />
    <PrimaryLanguage rdf:ID="Yoruba" />
    <PrimaryLanguage rdf:ID="Zulu" />
  </owl:oneOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="NativeLanguage">
  <owl:equivalentClass>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <PrimaryLanguage rdf:ID="English" />
        <PrimaryLanguage rdf:ID="Afrikaans" />
        <PrimaryLanguage rdf:ID="Ainu" />
        <PrimaryLanguage rdf:ID="Albanian" />
        <PrimaryLanguage rdf:ID="Amo" />
        <PrimaryLanguage rdf:ID="Aymara" />
        <PrimaryLanguage rdf:ID="Azerbaijani" />
        <PrimaryLanguage rdf:ID="Azeri" />
        <PrimaryLanguage rdf:ID="Bahasa" />
        <PrimaryLanguage rdf:ID="Basque" />
        <PrimaryLanguage rdf:ID="Batak" />
        <PrimaryLanguage rdf:ID="Batak_toba" />
        <PrimaryLanguage rdf:ID="Bengali" />
        <PrimaryLanguage rdf:ID="Bihari" />
        <PrimaryLanguage rdf:ID="Bosnian" />
        <PrimaryLanguage rdf:ID="Breton" />
        <PrimaryLanguage rdf:ID="Catalan" />
        <PrimaryLanguage rdf:ID="Cherokee" />
        <PrimaryLanguage rdf:ID="Cornish" />
        <PrimaryLanguage rdf:ID="Corsican" />
        <PrimaryLanguage rdf:ID="Cree" />
        <PrimaryLanguage rdf:ID="Croatian" />
        <PrimaryLanguage rdf:ID="Czech" />
        <PrimaryLanguage rdf:ID="Danish" />
        <PrimaryLanguage rdf:ID="Dutch" />
        <PrimaryLanguage rdf:ID="Edo" />
        <PrimaryLanguage rdf:ID="Esperanto" />
        <PrimaryLanguage rdf:ID="Estonian" />
        <PrimaryLanguage rdf:ID="Faroese" />
        <PrimaryLanguage rdf:ID="Fijian" />
        <PrimaryLanguage rdf:ID="Filipino" />
        <PrimaryLanguage rdf:ID="Finnish" />
        <PrimaryLanguage rdf:ID="French" />
        <PrimaryLanguage rdf:ID="Frisian" />
        <PrimaryLanguage rdf:ID="Gaelic" />
        <PrimaryLanguage rdf:ID="Galician" />
        <PrimaryLanguage rdf:ID="Gascon" />
        <PrimaryLanguage rdf:ID="German" />
        <PrimaryLanguage rdf:ID="Guarani" />
      </owl:oneOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

```

<PrimaryLanguage rdf:ID="Hanunoo" />
<PrimaryLanguage rdf:ID="Hausa" />
<PrimaryLanguage rdf:ID="Hawaiian" />
<PrimaryLanguage rdf:ID="Hmong" />
<PrimaryLanguage rdf:ID="Hopi" />
<PrimaryLanguage rdf:ID="Hungarian" />
<PrimaryLanguage rdf:ID="Ibibio" />
<PrimaryLanguage rdf:ID="Icelandic" />
<PrimaryLanguage rdf:ID="Indonesian" />
<PrimaryLanguage rdf:ID="Ingush" />
<PrimaryLanguage rdf:ID="Interlingua" />
<PrimaryLanguage rdf:ID="Inuktitut" />
<PrimaryLanguage rdf:ID="Inupiaq" />
<PrimaryLanguage rdf:ID="Irish" />
<PrimaryLanguage rdf:ID="Italian" />
<PrimaryLanguage rdf:ID="Javanese" />
<PrimaryLanguage rdf:ID="Kannada" />
<PrimaryLanguage rdf:ID="Kanuri" />
<PrimaryLanguage rdf:ID="Karelian" />
<PrimaryLanguage rdf:ID="Khasi" />
<PrimaryLanguage rdf:ID="Kirghiz" />
<PrimaryLanguage rdf:ID="Komi" />
<PrimaryLanguage rdf:ID="Kurdish" />
<PrimaryLanguage rdf:ID="Lapp" />
<PrimaryLanguage rdf:ID="Latin" />
<PrimaryLanguage rdf:ID="Latvian" />
<PrimaryLanguage rdf:ID="Lithuanian" />
<PrimaryLanguage rdf:ID="Lushootseed" />
<PrimaryLanguage rdf:ID="Luxemburgish" />
<PrimaryLanguage rdf:ID="Macedonian" />
<PrimaryLanguage rdf:ID="Malay" />
<PrimaryLanguage rdf:ID="Maltese" />
<PrimaryLanguage rdf:ID="Mari" />
<PrimaryLanguage rdf:ID="Naga" />
<PrimaryLanguage rdf:ID="Navajo" />
<PrimaryLanguage rdf:ID="Norwegian" />
<PrimaryLanguage rdf:ID="Occitan" />
<PrimaryLanguage rdf:ID="Oriya" />
<PrimaryLanguage rdf:ID="Polish" />
<PrimaryLanguage rdf:ID="Portuguese" />
<PrimaryLanguage rdf:ID="Provençal" />
<PrimaryLanguage rdf:ID="Prussian" />
<PrimaryLanguage rdf:ID="Punjabi" />
<PrimaryLanguage rdf:ID="Quechua" />
<PrimaryLanguage rdf:ID="Romanian" />
<PrimaryLanguage rdf:ID="Romany" />
<PrimaryLanguage rdf:ID="Sami" />
<PrimaryLanguage rdf:ID="Scots_Gaelic" />
<PrimaryLanguage rdf:ID="Serbo-Croatian" />
<PrimaryLanguage rdf:ID="Sesotho" />
<PrimaryLanguage rdf:ID="Shona" />
<PrimaryLanguage rdf:ID="Sinhalese" />
<PrimaryLanguage rdf:ID="Slovak" />
<PrimaryLanguage rdf:ID="Slovenian" />
<PrimaryLanguage rdf:ID="Somali" />
<PrimaryLanguage rdf:ID="Spanish" />
<PrimaryLanguage rdf:ID="Sudanese" />
<PrimaryLanguage rdf:ID="Swahili" />
<PrimaryLanguage rdf:ID="Swedish" />
<PrimaryLanguage rdf:ID="Tagalog" />
<PrimaryLanguage rdf:ID="Tagbanwa" />
<PrimaryLanguage rdf:ID="Tahitian" />
<PrimaryLanguage rdf:ID="Tajik" />
<PrimaryLanguage rdf:ID="Tamazight" />
<PrimaryLanguage rdf:ID="Turkish" />
<PrimaryLanguage rdf:ID="Turkmen" />
<PrimaryLanguage rdf:ID="Twi" />
<PrimaryLanguage rdf:ID="Udmurt" />
<PrimaryLanguage rdf:ID="Uighur" />
<PrimaryLanguage rdf:ID="Uzbek" />
<PrimaryLanguage rdf:ID="Vietnamese" />
<PrimaryLanguage rdf:ID="Welsh" />
<PrimaryLanguage rdf:ID="Xhosa" />
<PrimaryLanguage rdf:ID="Yi" />
<PrimaryLanguage rdf:ID="Yoruba" />
<PrimaryLanguage rdf:ID="Zulu" />
</owl:oneOf>
</owl:Class>
</owl:equivalentClass>

```

</owl:Class>

<PrimaryLanguage rdf:ID="English" />
<PrimaryLanguage rdf:ID="Afrikaans" />
<PrimaryLanguage rdf:ID="Ainu" />
<PrimaryLanguage rdf:ID="Albanian" />
<PrimaryLanguage rdf:ID="Amo" />
<PrimaryLanguage rdf:ID="Aymara" />
<PrimaryLanguage rdf:ID="Azerbaijani" />
<PrimaryLanguage rdf:ID="Azeri" />
<PrimaryLanguage rdf:ID="Bahasa" />
<PrimaryLanguage rdf:ID="Basque" />
<PrimaryLanguage rdf:ID="Batak" />
<PrimaryLanguage rdf:ID="Batak_toba" />
<PrimaryLanguage rdf:ID="Bengali" />
<PrimaryLanguage rdf:ID="Bihari" />
<PrimaryLanguage rdf:ID="Bosnian" />
<PrimaryLanguage rdf:ID="Breton" />
<PrimaryLanguage rdf:ID="Catalan" />
<PrimaryLanguage rdf:ID="Cherokee" />
<PrimaryLanguage rdf:ID="Cornish" />
<PrimaryLanguage rdf:ID="Corsican" />
<PrimaryLanguage rdf:ID="Cree" />
<PrimaryLanguage rdf:ID="Croatian" />
<PrimaryLanguage rdf:ID="Czech" />
<PrimaryLanguage rdf:ID="Danish" />
<PrimaryLanguage rdf:ID="Dutch" />
<PrimaryLanguage rdf:ID="Edo" />
<PrimaryLanguage rdf:ID="Esperanto" />
<PrimaryLanguage rdf:ID="Estonian" />
<PrimaryLanguage rdf:ID="Faroese" />
<PrimaryLanguage rdf:ID="Fijian" />
<PrimaryLanguage rdf:ID="Filipino" />
<PrimaryLanguage rdf:ID="Finnish" />
<PrimaryLanguage rdf:ID="French" />
<PrimaryLanguage rdf:ID="Frisian" />
<PrimaryLanguage rdf:ID="Gaelic" />
<PrimaryLanguage rdf:ID="Galician" />
<PrimaryLanguage rdf:ID="Gascon" />
<PrimaryLanguage rdf:ID="German" />
<PrimaryLanguage rdf:ID="Guarani" />
<PrimaryLanguage rdf:ID="Hanunoo" />
<PrimaryLanguage rdf:ID="Hausa" />
<PrimaryLanguage rdf:ID="Hawaiian" />
<PrimaryLanguage rdf:ID="Hmong" />
<PrimaryLanguage rdf:ID="Hopi" />
<PrimaryLanguage rdf:ID="Hungarian" />
<PrimaryLanguage rdf:ID="Ibibio" />
<PrimaryLanguage rdf:ID="Icelandic" />
<PrimaryLanguage rdf:ID="Indonesian" />
<PrimaryLanguage rdf:ID="Ingush" />
<PrimaryLanguage rdf:ID="Interlingua" />
<PrimaryLanguage rdf:ID="Inuktitut" />
<PrimaryLanguage rdf:ID="Inupiaq" />
<PrimaryLanguage rdf:ID="Irish" />
<PrimaryLanguage rdf:ID="Italian" />
<PrimaryLanguage rdf:ID="Javanese" />
<PrimaryLanguage rdf:ID="Kannada" />
<PrimaryLanguage rdf:ID="Kanuri" />
<PrimaryLanguage rdf:ID="Karelian" />
<PrimaryLanguage rdf:ID="Khasi" />
<PrimaryLanguage rdf:ID="Kirghiz" />
<PrimaryLanguage rdf:ID="Komi" />
<PrimaryLanguage rdf:ID="Kurdish" />
<PrimaryLanguage rdf:ID="Lapp" />
<PrimaryLanguage rdf:ID="Latin" />
<PrimaryLanguage rdf:ID="Latvian" />
<PrimaryLanguage rdf:ID="Lithuanian" />
<PrimaryLanguage rdf:ID="Lushootseed" />
<PrimaryLanguage rdf:ID="Luxemburgish" />
<PrimaryLanguage rdf:ID="Macedonian" />
<PrimaryLanguage rdf:ID="Malay" />
<PrimaryLanguage rdf:ID="Maltese" />
<PrimaryLanguage rdf:ID="Mari" />
<PrimaryLanguage rdf:ID="Naga" />
<PrimaryLanguage rdf:ID="Navajo" />
<PrimaryLanguage rdf:ID="Norwegian" />
<PrimaryLanguage rdf:ID="Occitan" />
<PrimaryLanguage rdf:ID="Oriya" />

<PrimaryLanguage rdf:ID="Polish" />
 <PrimaryLanguage rdf:ID="Portuguese" />
 <PrimaryLanguage rdf:ID="Provencal" />
 <PrimaryLanguage rdf:ID="Prussian" />
 <PrimaryLanguage rdf:ID="Punjabi" />
 <PrimaryLanguage rdf:ID="Quechua" />
 <PrimaryLanguage rdf:ID="Romanian" />
 <PrimaryLanguage rdf:ID="Romany" />
 <PrimaryLanguage rdf:ID="Sami" />
 <PrimaryLanguage rdf:ID="Scots_Gaelic" />
 <PrimaryLanguage rdf:ID="Serbo-Croatian" />
 <PrimaryLanguage rdf:ID="Sesotho" />
 <PrimaryLanguage rdf:ID="Shona" />
 <PrimaryLanguage rdf:ID="Sinhalese" />
 <PrimaryLanguage rdf:ID="Slovak" />
 <PrimaryLanguage rdf:ID="Slovenian" />
 <PrimaryLanguage rdf:ID="Somali" />
 <PrimaryLanguage rdf:ID="Spanish" />
 <PrimaryLanguage rdf:ID="Sudanese" />
 <PrimaryLanguage rdf:ID="Swahili" />
 <PrimaryLanguage rdf:ID="Swedish" />
 <PrimaryLanguage rdf:ID="Tagalog" />
 <PrimaryLanguage rdf:ID="Tagbanwa" />
 <PrimaryLanguage rdf:ID="Tahitian" />
 <PrimaryLanguage rdf:ID="Tajik" />
 <PrimaryLanguage rdf:ID="Tamazight" />
 <PrimaryLanguage rdf:ID="Turkish" />
 <PrimaryLanguage rdf:ID="Turkmen" />
 <PrimaryLanguage rdf:ID="Twi" />
 <PrimaryLanguage rdf:ID="Udmurt" />
 <PrimaryLanguage rdf:ID="Uighur" />
 <PrimaryLanguage rdf:ID="Uzbek" />
 <PrimaryLanguage rdf:ID="Vietnamese" />
 <PrimaryLanguage rdf:ID="Welsh" />
 <PrimaryLanguage rdf:ID="Xhosa" />
 <PrimaryLanguage rdf:ID="Yi" />
 <PrimaryLanguage rdf:ID="Yoruba" />
 <PrimaryLanguage rdf:ID="Zulu" />

<owl:AllDifferent>
 <owl:distinctMembers rdf:parseType="Collection">
 <orkut:PrimaryLanguage rdf:about="English" />
 <orkut:PrimaryLanguage rdf:about="Afrikaans" />
 <orkut:PrimaryLanguage rdf:about="Ainu" />
 <orkut:PrimaryLanguage rdf:about="Albanian" />
 <orkut:PrimaryLanguage rdf:about="Amo" />
 <orkut:PrimaryLanguage rdf:about="Aymara" />
 <orkut:PrimaryLanguage rdf:about="Azerbaijani" />
 <orkut:PrimaryLanguage rdf:about="Azeri" />
 <orkut:PrimaryLanguage rdf:about="Bahasa" />
 <orkut:PrimaryLanguage rdf:about="Basque" />
 <orkut:PrimaryLanguage rdf:about="Batak" />
 <orkut:PrimaryLanguage rdf:about="Batak_toba" />
 <orkut:PrimaryLanguage rdf:about="Bengali" />
 <orkut:PrimaryLanguage rdf:about="Bihari" />
 <orkut:PrimaryLanguage rdf:about="Bosnian" />
 <orkut:PrimaryLanguage rdf:about="Breton" />
 <orkut:PrimaryLanguage rdf:about="Catalan" />
 <orkut:PrimaryLanguage rdf:about="Cherokee" />
 <orkut:PrimaryLanguage rdf:about="Cornish" />
 <orkut:PrimaryLanguage rdf:about="Corsican" />
 <orkut:PrimaryLanguage rdf:about="Cree" />
 <orkut:PrimaryLanguage rdf:about="Croatian" />
 <orkut:PrimaryLanguage rdf:about="Czech" />
 <orkut:PrimaryLanguage rdf:about="Danish" />
 <orkut:PrimaryLanguage rdf:about="Dutch" />
 <orkut:PrimaryLanguage rdf:about="Edo" />
 <orkut:PrimaryLanguage rdf:about="Esperanto" />
 <orkut:PrimaryLanguage rdf:about="Estonian" />
 <orkut:PrimaryLanguage rdf:about="Faroese" />
 <orkut:PrimaryLanguage rdf:about="Fijian" />
 <orkut:PrimaryLanguage rdf:about="Filipino" />
 <orkut:PrimaryLanguage rdf:about="Finnish" />
 <orkut:PrimaryLanguage rdf:about="French" />
 <orkut:PrimaryLanguage rdf:about="Frisian" />
 <orkut:PrimaryLanguage rdf:about="Gaelic" />
 <orkut:PrimaryLanguage rdf:about="Galician" />
 <orkut:PrimaryLanguage rdf:about="Gascon" />
 <orkut:PrimaryLanguage rdf:about="German" />

```

<orkut:PrimaryLanguage rdf:about="Guarani" />
<orkut:PrimaryLanguage rdf:about="Hanunoo" />
<orkut:PrimaryLanguage rdf:about="Hausa" />
<orkut:PrimaryLanguage rdf:about="Hawaiian" />
<orkut:PrimaryLanguage rdf:about="Hmong" />
<orkut:PrimaryLanguage rdf:about="Hopi" />
<orkut:PrimaryLanguage rdf:about="Hungarian" />
<orkut:PrimaryLanguage rdf:about="Ibibio" />
<orkut:PrimaryLanguage rdf:about="Icelandic" />
<orkut:PrimaryLanguage rdf:about="Indonesian" />
<orkut:PrimaryLanguage rdf:about="Ingush" />
<orkut:PrimaryLanguage rdf:about="Interlingua" />
<orkut:PrimaryLanguage rdf:about="Inuktitut" />
<orkut:PrimaryLanguage rdf:about="Inupiaq" />
<orkut:PrimaryLanguage rdf:about="Irish" />
<orkut:PrimaryLanguage rdf:about="Italian" />
<orkut:PrimaryLanguage rdf:about="Javanese" />
<orkut:PrimaryLanguage rdf:about="Kannada" />
<orkut:PrimaryLanguage rdf:about="Kanuri" />
<orkut:PrimaryLanguage rdf:about="Karelian" />
<orkut:PrimaryLanguage rdf:about="Khasi" />
<orkut:PrimaryLanguage rdf:about="Kirghiz" />
<orkut:PrimaryLanguage rdf:about="Komi" />
<orkut:PrimaryLanguage rdf:about="Kurdish" />
<orkut:PrimaryLanguage rdf:about="Lapp" />
<orkut:PrimaryLanguage rdf:about="Latin" />
<orkut:PrimaryLanguage rdf:about="Latvian" />
<orkut:PrimaryLanguage rdf:about="Lithuanian" />
<orkut:PrimaryLanguage rdf:about="Lushootseed" />
<orkut:PrimaryLanguage rdf:about="Luxemburgish" />
<orkut:PrimaryLanguage rdf:about="Macedonian" />
<orkut:PrimaryLanguage rdf:about="Malay" />
<orkut:PrimaryLanguage rdf:about="Maltese" />
<orkut:PrimaryLanguage rdf:about="Mari" />
<orkut:PrimaryLanguage rdf:about="Naga" />
<orkut:PrimaryLanguage rdf:about="Navajo" />
<orkut:PrimaryLanguage rdf:about="Norwegian" />
<orkut:PrimaryLanguage rdf:about="Occitan" />
<orkut:PrimaryLanguage rdf:about="Oriya" />
<orkut:PrimaryLanguage rdf:about="Polish" />
<orkut:PrimaryLanguage rdf:about="Portuguese" />
<orkut:PrimaryLanguage rdf:about="Provençal" />
<orkut:PrimaryLanguage rdf:about="Prussian" />
<orkut:PrimaryLanguage rdf:about="Punjabi" />
<orkut:PrimaryLanguage rdf:about="Quechua" />
<orkut:PrimaryLanguage rdf:about="Romanian" />
<orkut:PrimaryLanguage rdf:about="Romany" />
<orkut:PrimaryLanguage rdf:about="Sami" />
<orkut:PrimaryLanguage rdf:about="Scots_Gaelic" />
<orkut:PrimaryLanguage rdf:about="Serbo-Croatian" />
<orkut:PrimaryLanguage rdf:about="Sesotho" />
<orkut:PrimaryLanguage rdf:about="Shona" />
<orkut:PrimaryLanguage rdf:about="Sinhalese" />
<orkut:PrimaryLanguage rdf:about="Slovak" />
<orkut:PrimaryLanguage rdf:about="Slovenian" />
<orkut:PrimaryLanguage rdf:about="Somali" />
<orkut:PrimaryLanguage rdf:about="Spanish" />
<orkut:PrimaryLanguage rdf:about="Sudanese" />
<orkut:PrimaryLanguage rdf:about="Swahili" />
<orkut:PrimaryLanguage rdf:about="Swedish" />
<orkut:PrimaryLanguage rdf:about="Tagalog" />
<orkut:PrimaryLanguage rdf:about="Tagbanwa" />
<orkut:PrimaryLanguage rdf:about="Tahitian" />
<orkut:PrimaryLanguage rdf:about="Tajik" />
<orkut:PrimaryLanguage rdf:about="Tamazight" />
<orkut:PrimaryLanguage rdf:about="Turkish" />
<orkut:PrimaryLanguage rdf:about="Turkmen" />
<orkut:PrimaryLanguage rdf:about="Twi" />
<orkut:PrimaryLanguage rdf:about="Udmurt" />
<orkut:PrimaryLanguage rdf:about="Uighur" />
<orkut:PrimaryLanguage rdf:about="Uzbek" />
<orkut:PrimaryLanguage rdf:about="Vietnamese" />
<orkut:PrimaryLanguage rdf:about="Welsh" />
<orkut:PrimaryLanguage rdf:about="Xhosa" />
<orkut:PrimaryLanguage rdf:about="Yi" />
<orkut:PrimaryLanguage rdf:about="Yoruba" />
<orkut:PrimaryLanguage rdf:about="Zulu" />
</owl:distinctMembers>
</owl:AllDifferent>

```

```

<!--/Primary language= native language= mother tongue of a person.-->
<!--Hometown of a person. This item applies only to a person living in the U.S.-->
<owl:Class rdf:ID="US_City">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#CaliforniaCity" />
    <owl:Class rdf:about="#MassachusettsCity" />
    <owl:Class rdf:about="#OtherStatesCity" />
  </owl:unionOf>
</owl:Class>

<owl:Class rdf:ID="CaliforniaCity">
  <rdfs:subClassOf rdf:resource="#US_City" />
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Santa_Clara" />
    <owl:Thing rdf:about="#San_Hose" />
  </owl:oneOf>
</owl:Class>

<owl:Class rdf:ID="MassachusettsCity">
  <rdfs:subClassOf rdf:resource="#US_City" />
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Belmont_MA" />
    <owl:Thing rdf:about="#Boston" />
  </owl:oneOf>
</owl:Class>

<owl:Class rdf:ID="OtherStatesCity">
  <rdfs:subClassOf rdf:resource="#US_City" />
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#AndMuchMore1" />
    <owl:Thing rdf:about="#AndMuchMore2" />
  </owl:oneOf>
</owl:Class>

<MassachusettsCity rdf:ID="Belmont_MA" />
<MassachusettsCity rdf:ID="Boston" />

<CaliforniaCity rdf:ID="Santa_Clara" />
<CaliforniaCity rdf:ID="San_Hose" />

<OtherStatesCity rdf:ID="AndMuchMore1" />
<OtherStatesCity rdf:ID="AndMuchMore2" />

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <orkut:MassachusettsCity rdf:about="#Belmont_MA" />
    <orkut:MassachusettsCity rdf:about="#Boston" />
  </owl:distinctMembers>
</owl:AllDifferent>

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <orkut:CaliforniaCity rdf:about="#Santa_Clara" />
    <orkut:CaliforniaCity rdf:about="#San_Hose" />
  </owl:distinctMembers>
</owl:AllDifferent>

<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <orkut:OtherStatesCity rdf:about="#AndMuchMore1" />
    <orkut:OtherStatesCity rdf:about="#AndMuchMore2" />
  </owl:distinctMembers>
</owl:AllDifferent>

<US_State rdf:ID="Alabama" />
<US_State rdf:ID="Alaska" />
<US_State rdf:ID="Arizona" />
<US_State rdf:ID="Arkansas" />

<US_State rdf:ID="California">
  <containsUS_City rdf:resource="#Santa_Clara" />
  <containsUS_City rdf:resource="#San_Hose" />
</US_State>

<US_State rdf:ID="Colorado" />
<US_State rdf:ID="Connecticut" />

```

```

<US_State rdf:ID="Delaware" />
<US_State rdf:ID="District_of_Columbia" />
<US_State rdf:ID="Florida" />
<US_State rdf:ID="Georgia" />
<US_State rdf:ID="Hawaii" />
<US_State rdf:ID="Idaho" />
<US_State rdf:ID="Illinois" />
<US_State rdf:ID="Indiana" />
<US_State rdf:ID="Iowa" />
<US_State rdf:ID="Kansas" />
<US_State rdf:ID="Kentucky" />
<US_State rdf:ID="Louisiana" />
<US_State rdf:ID="Maine" />
<US_State rdf:ID="Maryland" />

<US_State rdf:ID="Massachusetts">
  <containsUS_City rdf:resource="#Belmont_MA" />
  <containsUS_City rdf:resource="#Boston" />
</US_State>

<US_State rdf:ID="Michigan" />
<US_State rdf:ID="Minnesota" />
<US_State rdf:ID="Mississippi" />
<US_State rdf:ID="Missouri" />
<US_State rdf:ID="Montana" />
<US_State rdf:ID="Nebraska" />
<US_State rdf:ID="Nevada" />
<US_State rdf:ID="New_Hampshire" />
<US_State rdf:ID="New_Jersey" />
<US_State rdf:ID="New_Mexico" />
<US_State rdf:ID="New_York" />
<US_State rdf:ID="North_Carolina" />
<US_State rdf:ID="North_Dakota" />
<US_State rdf:ID="Ohio" />
<US_State rdf:ID="Oklahoma" />
<US_State rdf:ID="Oregon" />
<US_State rdf:ID="Pennsylvania" />
<US_State rdf:ID="Rhode_Island" />
<US_State rdf:ID="South_Carolina" />
<US_State rdf:ID="South_Dakota" />
<US_State rdf:ID="Tennessee" />
<US_State rdf:ID="Utah" />
<US_State rdf:ID="Vermont" />
<US_State rdf:ID="Virginia" />
<US_State rdf:ID="Washington" />
<US_State rdf:ID="West_Virginia" />
<US_State rdf:ID="Wisconsin" />
<US_State rdf:ID="Wyoming" />

<US_State rdf:ID="OtherStates">
  <containsUS_City rdf:resource="#AndMuchMore1" />
  <containsUS_City rdf:resource="#AndMuchMore2" />
</US_State>

<owl:Class rdf:ID="US_State">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Alabama" />
    <owl:Thing rdf:about="#Alaska" />
    <owl:Thing rdf:about="#Arizona" />
    <owl:Thing rdf:about="#Arkansas" />
    <owl:Thing rdf:about="#California" />
    <owl:Thing rdf:about="#Colorado" />
    <owl:Thing rdf:about="#Connecticut" />
    <owl:Thing rdf:about="#Delaware" />
    <owl:Thing rdf:about="#District_of_Columbia" />
    <owl:Thing rdf:about="#Florida" />
    <owl:Thing rdf:about="#Georgia" />
    <owl:Thing rdf:about="#Hawaii" />
    <owl:Thing rdf:about="#Idaho" />
    <owl:Thing rdf:about="#Illinois" />
    <owl:Thing rdf:about="#Indiana" />
    <owl:Thing rdf:about="#Iowa" />
    <owl:Thing rdf:about="#Kansas" />
    <owl:Thing rdf:about="#Kentucky" />
    <owl:Thing rdf:about="#Louisiana" />
    <owl:Thing rdf:about="#Maine" />
    <owl:Thing rdf:about="#Maryland" />
    <owl:Thing rdf:about="#Massachusetts" />
    <owl:Thing rdf:about="#Michigan" />
  </owl:oneOf>
</owl:Class>

```

```

    <owl:Thing rdf:about="#Minnesota" />
    <owl:Thing rdf:about="#Mississippi" />
    <owl:Thing rdf:about="#Missouri" />
    <owl:Thing rdf:about="#Montana" />
    <owl:Thing rdf:about="#Nebraska" />
    <owl:Thing rdf:about="#Nevada" />
    <owl:Thing rdf:about="#New_Hampshire" />
    <owl:Thing rdf:about="#New_Jersey" />
    <owl:Thing rdf:about="#New_Mexico" />
    <owl:Thing rdf:about="#New_York" />
    <owl:Thing rdf:about="#North_Carolina" />
    <owl:Thing rdf:about="#North_Dakota" />
    <owl:Thing rdf:about="#Ohio" />
    <owl:Thing rdf:about="#Oklahoma" />
    <owl:Thing rdf:about="#Oregon" />
    <owl:Thing rdf:about="#Pennsylvania" />
    <owl:Thing rdf:about="#Rhode_Island" />
    <owl:Thing rdf:about="#South_Carolina" />
    <owl:Thing rdf:about="#South_Dakota" />
    <owl:Thing rdf:about="#Tennessee" />
    <owl:Thing rdf:about="#Utah" />
    <owl:Thing rdf:about="#Vermont" />
    <owl:Thing rdf:about="#Virginia" />
    <owl:Thing rdf:about="#Washington" />
    <owl:Thing rdf:about="#West_Virginia" />
    <owl:Thing rdf:about="#Wisconsin" />
    <owl:Thing rdf:about="#Wyoming" />
  </owl:oneOf>
</owl:Class>

<owl:ObjectProperty rdf:ID="hasUS_Hometown">
  <rdfs:domain rdf:resource="#Person" />
  <rdfs:range rdf:resource="#US_City" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="containsUS_City">
  <rdfs:domain rdf:resource="#US_State" />
  <rdfs:range rdf:resource="#US_City" />
</owl:ObjectProperty>

<!--/Hometown of a person. This item only applies to people living in the U.S.-->
</rdf:RDF>

```