IST-Africa 2011 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2011
ISBN: 978-1-905824-26-7

Semantic-Driven E-government: A Case Study of Formal Representation of Government Domain Ontology

Jean Vincent FONOU DOMBEU[1], Magda HUISMAN[2]
[1]Vaal University of Technology, Private Bag X021, Andries Potgieter Blvd, Vanderbijlpark, 1900, South Africa
Tel: +27 (0)16 950-9831, Fax: +27 (0) 16 950-9497, Email: fonoudombeu@gmail.com
[2]North-West University, Private Bag X6001, Potchefstroom, 2520, South Africa
Tel: +27 (0)18 299-4049, Fax: +27 (0)18 293-5247, Magda.Huisman@nwu.ac.za

Abstract: Semantic Web technologies are increasingly being adopted in e-government to describe and specify e-government services, aiming at their semantic integration and interoperability. However, current research does not provide guidelines on how domain ontology describing government's domain knowledge can be formally represented to enable their automatic processing by computers. This paper presents a case study of transforming domain ontology in Sub Saharan Africa and the developing world at large, into its formal version in e-government processes, aiming at (1) providing e-government developers from developing countries as well as Sub Saharan Africa, with simple activities for practicing semantic knowledge representation in e-government projects, (2) promoting the development of e-government systems that can be easily integrated and maintained, as well as (3) strengthening the adoption of Semantic Web technologies in e-government projects in developing countries.

Keywords: E-government, Domain Ontology, Description Logics, OWL, Protégé.

1. Introduction

Semantic Web technologies are mainly based on ontology. Ontology is commonly defined as an explicit specification of a conceptualization [3]. A conceptualization is an abstract and simplified view of a domain of knowledge one wishes to represent for a certain purpose; the domain could be explicitly and formally represented using existing objects, concepts, entities and the relationship that exist between them [3]. Ontology is widely used in disciplines such as computer science, software engineering, databases, artificial intelligence, and many more [4], [5]. In these fields, developers use ontology to represent knowledge in a manner that can be automatically processed by computers.

E-government has been one of the most active areas of ontology development during the past six years [6], [7], [8], [9], [10], [11]. In e-government, ontology is being used to describe and specify e-government services (e-services) aiming at their semantic integration and interoperability. In [6], [11] ontological approaches for service integration in e-government are proposed; the ontology-based solutions are modelled with Web Ontology Language (OWL) [6], [11] and Web Service Modelling Language (WSML) [6] for semantic processing. Others ontology-based solutions for semantic interoperability in e-government are proposed in Muthaiyah and Kerschberg [7] and Sabucedo et al. [8]; these solutions are represented in OWL to allow the mapping, merging and integration of e-government services. Apostolou et al. [9], [10] present a software engineering platform for the development and management of e-government services namely ONTOGOV. The ONTOGOV platform uses Semantic Web technologies including OWL-S and Web Service Modelling Ontology (WSMO) to construct eight kinds of ontology models characterizing the e-government domain; these ontology models aims at describing and composing services provided by public administrators.

On the one hand, the above works demonstrate the interest of Semantic Web technologies based on ontology in e-government; they show how various ontology-based solutions are being used in e-government research and projects to describe and specify e-government services, aiming at their semantic integration and interoperability. However these works are highly technical and more directed to Semantic Web specialists than to the broader e-government community. Furthermore, none of these works provide detailed guidelines on how e-government system developers without any knowledge of ontology and Semantic Web technologies can construct the proposed semantic-based solutions in a government service domain.

On the other hand, a review of research [1] indicates that e-government applications currently run in Sub Saharan Africa (SSA) are at the initial state of e-government development. In other words, current SSA e-government applications provide only one way interaction with citizens; they mainly present government information in the form of simple web pages and offer few online services to citizens and stakeholders. Furthermore, these applications are still largely being developed autonomously by individual government departments; this is witnessed by the existence of autonomous government department websites in many SSA countries [2]. There is no evidence that Semantic Web technologies which have emerged in the past six years [6], [7], [8], [9], [10], [11] as promising solutions for developing e-government systems that can be easily integrated and interoperated to provide seamless service delivery to citizen in a single web portal, are being considered in the development of SSA e-government applications. This paper presents a case study of transforming a domain ontology in SSA and the developing world at large, into its formal representation in e-government processes, aiming at (1) providing e-government developers from developing countries as well as SSA, with simple activities for practicing semantic knowledge representation in e-government projects, (2) promoting the development of e-government systems that can be easily integrated and maintained, and (3) strengthening the adoption of Semantic Web technologies in e-government projects in developing countries.

The rest of the paper is organized as follows. Section 2 presents the government service domain considered in this study. Then, the techniques employed to develop the domain ontology and those applied to transform it into its semi-formal and formal representations are described. Section 3 presents a discussion and a conclusion ends the paper.

2. Methodology

2.1 Create Domain Ontology

The idea of building the domain ontology used in this paper was motivated by the fact that, in developing countries as well as in SSA, almost every government department is somehow involved in the implementation of a programme aiming at improving the welfare of its people. These programmes are commonly called development projects and include infrastructure development, water supply and sanitation, education, rural development, health care, ICT infrastructure development and so forth. Thus, an application that could interface all the activities related to development projects implementation in a SSA country could bring tremendous advantages; particularly, such a web application would improve the monitoring and evaluation of projects and provide transparency, efficiency and better delivery to populations. In [1] we have proposed an ontology support model for such a web-based e-government application. We evaluated case studies of development projects implementation, consulted domain experts including municipalities and non-governmental organizations employees and academic members, and reviewed publications in related fields including project management, project monitoring and evaluation, and capacity building [1]. Thus, a conceptual/domain ontology of development projects monitoring (OntoDPM) in a developing country was developed [1].

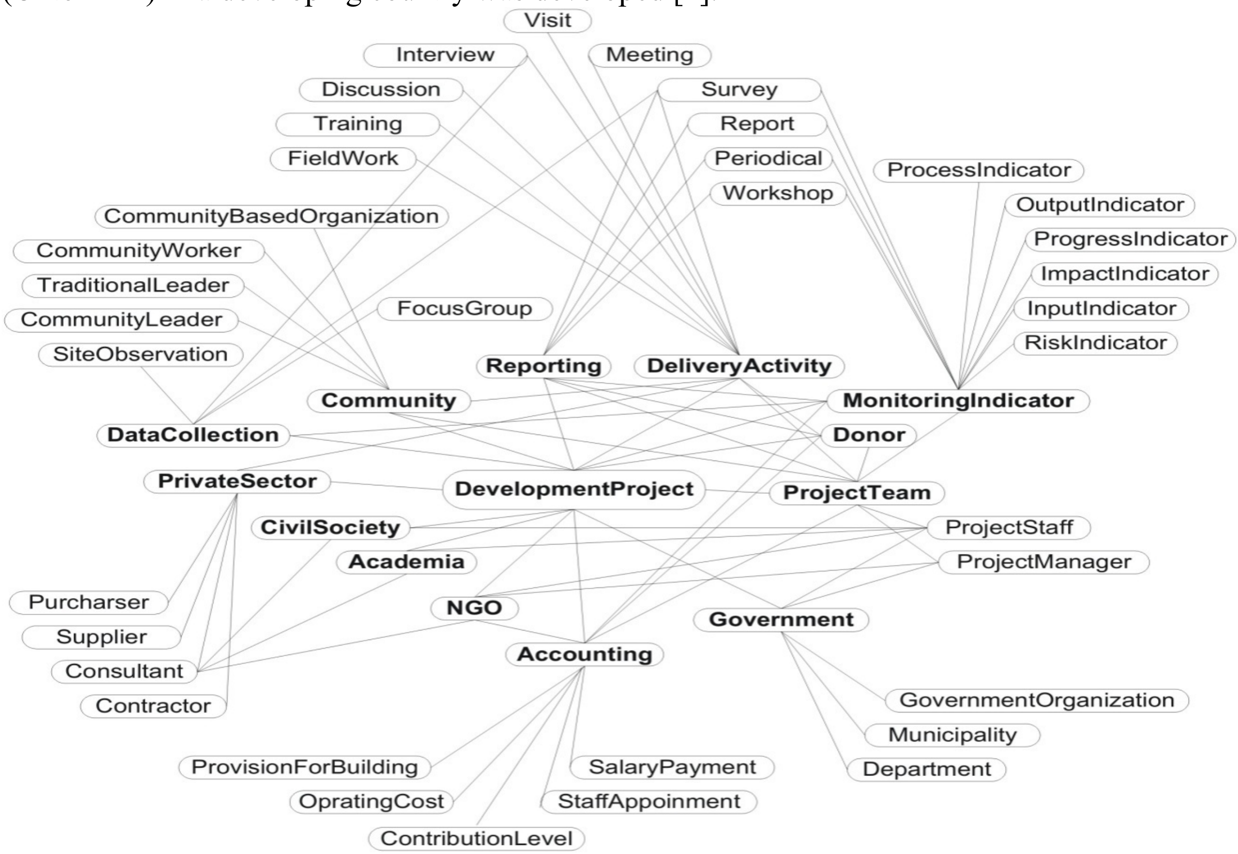

Figure 1: OntoDPM Domain Ontology

The ontology engineering field has established various kinds of ontologies; an exhaustive list of these ontologies could be found in [5]. One of the most commonly used of these ontologies is the conceptual/domain ontology. A domain ontology characterizes domains such as medicine, geology, e-government, and so on; it provides vocabulary about the objects and concepts within a domain and their relationships, the activities that take place in that domain, and theories and elementary principles governing the domain [21]. In [1] we followed a five step framework adopted from the Uschold and King [12] ontology building methodology to represent the OntoDPM domain ontology as in Figure 1. The OntoDPM shows the key concepts of the domain (people, stakeholder, financier, monitoring indicator, reporting technique, etc.), the activities carried out in the domain (training, discussion, fieldwork, visit, meeting, etc.) and the relationships between the constituents of the domain. This study does not expand on the framework used to build the OntoDPM; instead, the study focuses on the process for creating the formal representation of the OntoDPM to facilitate its automatic processing by computers.

The steps we used to generate the formal version of the OntoDPM are described as follows. Firstly, the OntoDPM domain ontology is analysed and its axioms, class hierarchy, properties and instances are represented in description logic [14], [15], [16]. Thereafter, the class hierarchy, properties and instances are created with the ontology knowledge based editor Protégé [18] and saved onto the disc. Finally, the formal version of the OntoDPM is imported. The next subsections present the abovementioned steps in detail.

2.2 Semi-formal Representation of the OntoDPM in Description Logic

The ontology engineering field prescribes three layers of ontology development according to the formalisms used [12]. From a form that can be understood by human beings to one that can be processed by computers, these ontology layers are: informal ontology, semi-formal ontology and formal ontology [12]. Two formalisms are commonly used to represent a semi-formal ontology; they include UML class diagram [13] and description logic [14], [15]. In this research, we have chosen the description logic formalism. Description Logic is a formal language for knowledge representation. Its syntax uses basic mathematical logic symbols such as subset, union, intersection, universal and existential quantifications, etc. to represent the relationships between the constituents of a domain. The description logic version of the OntoDPM domain ontology in Figure 1 is obtained by analysing its semantic and logical structures, identifying its classes, class hierarchy and class instances, and defining relationships between classes. The relationships include inheritance and association/composition relationships. A relationship is also called property or slot. Thereafter, the mathematical logic symbols mentioned above are used to represent the class hierarchy, relationships between classes (inheritance and properties), constraints on properties, etc. For instance, in the class hierarchy of the OntoDPM, community worker, community leader, traditional leader, and project staff are people (person class) involved in the development project implementation. Therefore, community worker, community leader, traditional leader, and project staff are subclasses of the person class, representing an inheritance relationship. This relationship is represented in description logic formalism using the subset and existential quantification symbols, and the isA property as follows:

$$ProjectStaff \subseteq \exists\ isA.Person$$
$$CommunityWorker \subseteq \exists\ isA.Person$$
$$CommunityLeader \subseteq \exists\ isA.Person$$
$$TraditionalLeader \subseteq \exists\ isA.Person$$

The isA property represents the inheritance relationship between classes. Similarly, the class hierarchy of the OntoDPM shows that department, agency and municipality are division of government. This relationship between government and its divisions can be represented in description logic with a hasDivision property, the subset, existential quantification, and union symbols as follows:

$$Government \subseteq \exists\ hasDivision.(Department \cup Agency \cup Municipality)$$

A part of description logic representation of the OntoDPM in Figure 1 is provided in Appendix A. More information on description logic formalism could be found in [14], [15]. The, description logic representation of domain ontology is useful as it provides strong logical structure for the description and specification of domain knowledge [15], [16], facilitating the detection of semantic inconsistencies in domain ontology [14] and enabling semantic reasoning over the resulting ontology model. Furthermore, OWL standard which is widely used in the field of Semantic Web is based on description logic [16]. The OWL formal representation of the OntoDPM is described in the next section.

2.3 *Formal Representation of the OntoDPM Domain Ontology*

Semantic e-government entails using semantic ontology models to represent and describe government services in such a way that they can be automatically processed by computers. Therefore, the semi-formal version of the OntoDPM domain ontology presented above must be rewritten in a formal representation using an existing ontology representation language. The Semantic Web domain provides various languages for the formal representation of ontologies including XML, RDF, DAML, and OWL [17]. OWL is the most widely used of these languages because of its high expressive power and the fact that it is the W3C standard ontology language for the Semantic Web [18]. Several software tools are also used for ontology edition including WebODE, OntoEdit, KAON1, and Protégé [5]. Ontology developers prefer Protégé for its ease of use and its abstraction capabilities; it has a graphical user interface which enables ontology developers to concentrate on conceptual modeling without any knowledge of the syntax of the output language [18]. Furthermore, Protégé is open-source software which is downloadable from the Stanford Medical Informatics website.

In this research, we have used Protégé to create the formal version of the OntoDPM using the semi-formal specifications of its class hierarchy, relationships between classes and class instances. Several concepts including class, individual, slot, domain, range, etc. are employed in Protégé to create ontology. A slot is also called property and represents a relationship between classes. Each slot has a domain and a range, which are the classes involved in the relationship. Once ontology has been created with Protégé and saved as an OWL file onto the disc, it appears in the OWL syntax. The OWL syntax provides facilities for representing ontology elements such as inheritance, instance, slots, domain and range of a slot, etc. For instance in the OntoDPM, community worker, community leader, traditional leader, and project staff are subclasses of the person class (see description logic example in the previous section). This inheritance relationship is represented in OWL syntax with the following OWL code generated with Protégé.

```
<owl:Class rdf:about="#ProjectStaff">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:about="#CommunityLeader">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:about="#CommunityWorker">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:about="#TraditionalLeader">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
```

Similarly, in the OntoDPM, department, agency and municipality are divisions of government. The relationship between government and its divisions can be represented in Protégé with the slot hasDivision. Then, the domain of the hasDivision slot will be the government class and its range, the department, agency, and municipality classes. The hasDivision slot and its domain and ranges are represented in OWL with the following code generated with Protégé.

```
<owl:ObjectProperty rdf:about="#hasDivision">
  <rdfs:domain rdf:resource="#Government"/>
```

```
    <rdfs:range rdf:resource="#Agency"/>
    <rdfs:range rdf:resource="#Department"/>
    <rdfs:range rdf:resource="#Municipality"/>
  </owl:ObjectProperty>
```

A part of OWL version of the OntoDPM created with Protégé from its semi-formal representation is depicted in Appendix B. The next section discusses the usefulness of the OWL formal ontology and outlines the direction of our future research.

3. Discussion

This study describes how a domain ontology which models the semantic content of a government service domain can be represented in OWL to enable its automatic processing by computers. As highlighted in Section 1, OWL is a common language employed for semantic knowledge representation in e-government. In particular, OWL ontologies allow the composition [8], [9], searching, matching, mapping and merging [10], [11] of e-government services and facilitate their integration [6], [9], [10], maintenance [9] and interoperability [7], [8], [11]. Therefore, generating OWL ontology from a government service domain as it is done in this research is an important step towards the development of Semantic Web applications as e-government applications, which have potential to perform semantic inference and reasoning over the OWL ontology and facilitate software components integration and interoperability.

A detailed discussion on the use of the generated OWL ontology is beyond the scope of this paper and will be the focus of our future work. We intend to investigate various platforms currently being used for Semantic Web development based on OWL ontology models including Java API, .NET, ASP [20]. In particular, we will have a close look at the mapping of OWL ontology with the Java syntax [20].

4. Conclusions

On the one hand, the study has revealed that semantic ontology-based solutions are increasingly being adopted in e-government projects and research to develop e-government systems that can be easily integrated, interoperated and maintained. However, current works do not provide guidelines for repeating the proposed semantic ontology-based solutions in a government service domain. On the other hand, the study has pointed out that SSA e-government systems are still at the first stage of e-government development and there is no evidence of the use of semantic technologies in the development of these applications. Then, in order to assist e-government developers from SSA and the developing world at large in practicing semantic content representation in their e-government projects, the study has presented a case study of transforming a domain ontology describing a government service domain into its formal version, for computer processing.

The study uses various platforms including Protégé, JCreator, and JGrasp to create and import the OWL formal version of the domain ontology. These are mainly open source software; which make the study repeatable by the broader e-government community, particularly e-government developers from the developing countries as well as SSA where there is little or no practice of semantic content representation for e-government systems.

Despite the detailed description of the semantic development processes provided in this study, we think that it is important that developing countries and SSA governments in particular adopt sustainable solutions for developing e-government applications with Semantic Web technologies in the continent. The first medium to a long-term solution would be the inclusion of Semantic Web courses in the IT programmes at the University level in SSA. Currently, only a few computer science and IT departments at SSA universities are offering Semantic Web courses. Another solution that could be adopted at the government level would be the frequent organization of seminars and workshops for IT employees, on ontology development, from the conceptual modeling to the implementation with open-source Semantic Web ontology editing and implementing platforms.

In terms of partnership of European countries with SSA countries, governments of both parties could sign agreements for staff training in which IT employees of government departments could visit universities or agencies in Europe to study the use of semantic technologies for e-application development.

References

[1] J.V. Fonou-Dombeu and M. Huisman, Investigating E-government Knowledge Base ontology Supporting Development Project Monitoring in Sub Saharan Africa, *International Journal of Computing and ICT Research*, Special Issue, Vol. 4, No. 1, pp. 20-29, October, 2010.

[2] E.N., Asiimwe and N., Lim, Usability of Government Websites in Uganda, *Electronic Journal of e-Government*, Vol. 8, No. 1, pp. 1-12, 2010.

[3] T. R., Gruber, Toward Principles for the Design of Ontologies used for Knowledge Sharing, *International Journal of Human-Computer Studies* 43, pp.907-928, 1993.

[4] C.A., Welty, Ontology Research, AI Magazine, Vol. 24, No. 3, pp 11-12, 2003.

[5] C., Calero, F., Ruiz and M., Piattini, Ontologies for Software Engineering and Software Technology, *Calero.Ruiz.Piattini (Eds.), Springer-Verlag Berlin Heidelberg*, 2006.

[6] P., Salhofer, B., Stadlhofer and G., Tretter, Ontology Driven E-government, *Electronic Journal of E-government*, Vol. 7, No. 4, pp. 415-424, 2009.

[7] S., Muthaiyah and L., Kerschberg, Achieving Interoperability in E-government Services with two Modes of Semantic Bridging: SRS and SWRL, *Journal of Theoretical and Applied Electronic Commerce Research*, Vol. 3, No. 3, pp. 52-63, December, 2008.

[8] L.M.A., Sabucedo, L.E.A., Rifon, F., Corradini, A., Polzonetti and B., Re, Knowledge-Based Platform for E-government Agents: A Web-based Solution Using Semantic Technologies, *Journal of Expert Systems with Applications*, Elsevier Inc., Vol. 5, No. 37, pp. 3647-3656, 2010.

[9] D., Apostolou, L., Stojanovic, T.P., Lobo, J.C., Miro and A., Papadakis, Configuring E-government Services Using Ontologies, IFIP International Federation for Information Processing, *Springer Boston*, Vol. 2005, No. 189, pp. 141-155, 2005a.

[10] D., Apostolou, L., Stojanovic, T.P., Lobo and B., Thoensen, Towards a Semantically-Driven Software Engineering Environment for E-government, IFIP International Federation for Information Processing, M. Bohlen (Eds), TCGOV 2005, LNAI 3416, pp. 157-168, 2005b.

[11] F. Sanati and J. Lu, Multilevel Life-event Abstraction Framework for E-government Service Integration, *In Proceedings of the 9th European Conference on E-government 2009 (ECEG 2009)*, London, UK, pp. 550-558, 29-30 June, 2009.

[12] M., Uschold and M., King, Towards a Methodology for Building Ontologies, *In Proceedings of IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing,* Montreal, Canada, 1995.

[13] L., Ceccaroni and E.A., Kendall, Semantically-Rich, Graphical Environment for Collaborative Ontology Development in Agentcities,iD3, Barcelona, Spain, 2003.

[14] M., Fahad, M.A., Qadir and S.A.H., Shah, Evaluation of Ontologies and DL Reasoners, *In IFIP International Federation for Information Processing*, Boston: Springer, Vol. 288, pp. 17-27, 2008.

[15] F., Baader, I., Horrocks and U., Sattler, Description Logics as Ontology Languages for the Semantic Web, Mechanizing Mathematical Reasoning, *Lecture Notes in Computer Science*, 2005, Vol. 2005, No. 2605, pp. 228-248, 2005.

[16] J. Z., Pan, A Flexible Ontology Reasoning Architecture for the Semantic Web, IEEE Transaction on Knowledge and Data Engineering, Vol. 19, No. 2, pp. 246 – 260, February, 2007.

[17] M., Laclavik, Ontology and Agent Based Approach for Knowledge Management, *PhD Thesis*, Institute of Informatics, Slovak Academy of Sciences, June, 2005.

[18] M., Singh, S.K., Malik, Constructing Ontologies in OWL Using Protégé-2000, *In Proceedings of the 2nd National Conference on Challenges and Opportunities in Information Technology (COIT 2008)*, Mandi Gobindgarh, Punjab-India, 2008.

[19] A., Saekow and C., Boonmee, A Practical Approach to Interoperability Practical Implementation Support (IPIS) for E-government Interoperability, *Electronic Journal of E-government*, Vol. 7, No. 4, pp. 403-414, 2009.

[20] A., Kalyanpur, D., Jimenez, S., Battle and J., Padget, Automatic Mapping of OWL Ontologies into Java, *In Proceedings of the 16th International Conference on Software Engineering and Knowledge Engineering (SEKE'2004)*, June 2004, pp. 98-103, 2004.
[21] A. Gomez-Perez and V.R. Benjamins, Overview of knowledge Sharing and Reuse Components: Ontology and Problem-Solving Methods, *In Proceedings of the IJCAI-99 workshop on Ontologies and Problem- Solving Methods (KRR5)*, Stockholm, Sweden, August 2, 1999.

Appendix A: **Part of Description Logic Representation of the OntoDPM**

$DevelopmentProject \subseteq \exists\ hasFunder.Financier$

$DevelopmentProject \subseteq \exists\ hasCorporate.Stakeholder$

$DevelopmentProject \subseteq \exists\ hasPeople.Person$

$DevelopmentProject \subseteq \exists\ hasIndicator.MonitoringIndicator$

$DevelopmentProject \subseteq \exists\ hasActivity.DeliveryActivity$

$DevelopmentProject \subseteq \exists\ hasReport.ReportingTechnique$

$DevelopmentProject \subseteq \exists\ hasData.DataCollectionTechnique$

$ProjectStaff \subseteq \exists\ isA.Person$

$CommunityWorker \subseteq \exists\ isA.Person$

$CommunityLeader \subseteq \exists\ isA.Person$

$TraditionalLeader \subseteq \exists\ isA.Person$

$Government \subseteq \exists\ isA.Financier$

$Donor \subseteq \exists\ isA.Financier$

$NonGovernmentalOrganization \subseteq \exists\ isA.Financier$

$PrivateCompany \subseteq \exists\ isA.Stakeholder$

$AcademicInstitution \subseteq \exists\ isA.Stakeholder$

$CommunityBasedOrganization \subseteq \exists\ isA.Stakeholder$

$CivilSociety \subseteq \exists\ isA.Stakeholder$

$Consultant \subseteq \exists\ isA.PrivateCompany$

$Contractor \subseteq \exists\ isA.PrivateCompany$

$Supplier \subseteq \exists\ isA.PrivateCompany$

$Purcharser \subseteq \exists\ isA.PrivateCompany$

$Government \subseteq \exists\ hasDivision.(Department \cup Agency \cup Municipality)$

Appendix B: **Part of OWL Representation of the OntoDPM**

```
<owl:Class rdf:about="#Person"/>
  <owl:Class rdf:about="#ProjectStaff">
    <rdfs:subClassOf rdf:resource="#Person"/>
  </owl:Class>
  <owl:Class rdf:about="#CommunityLeader">
    <rdfs:subClassOf rdf:resource="#Person"/>
  </owl:Class>
  <owl:Class rdf:about="#CommunityWorker">
    <rdfs:subClassOf rdf:resource="#Person"/>
  </owl:Class>
  <owl:Class rdf:about="#TraditionalLeader">
    <rdfs:subClassOf rdf:resource="#Person"/>
  </owl:Class>
</owl:Class>
<owl:Class rdf:about="#Financier"/>
  <owl:Class rdf:about="#Donor">
    <rdfs:subClassOf rdf:resource="#Financier"/>
  </owl:Class>
  <owl:Class rdf:about="#Government">
    <rdfs:subClassOf rdf:resource="#Financier"/>
  </owl:Class>
  <owl:Class rdf:about="#NGO">
    <rdfs:subClassOf rdf:resource="#Financier"/>
  </owl:Class>
</owl:Class>
```

