Representing Roles in Formalizing Domain Ontology for Land Administration

Kean Huat SOON, Singapore

Key words: Ontology, Web Ontology Language (OWL), User Roles, Land Administration Domain Model (LADM), Automation

SUMMARY

In Land Administration, web portals have been developed to support various customers on property transactions, applications for registration of land titles, submission of survey plans for authority's approval, etc. The user groups of these portals are huge and range from various parties, such as surveyors, government authorities, landowners, member of the public, and lawyers. A formalized ontology that emphasizes user roles in Land Administration will help identify user roles by reasoning about the documents/information submitted. This ability will allow the system to serve customers more proactively. The paper formalizes domain ontology for Land Administration from natural language texts in the standard ISO 19152 Land Administration Domain Model (LADM) using Web Ontology Language (OWL). OWL is chosen because it supports reasoning and inference of new knowledge. Comparing with the existing UML (Unified Modeling Language) model, natural language texts are a good source to provide a neutral stance for developing the ontology without a prior assumption.

In the existing LADM model, however, user roles are rather represented rigidly. This has confined the way to model roles as context dependent. Also, role is involved in the association between BAUnit and Party through the relation baunitAsParty. BaunitAsParty is semantically different but the existing model has treated it like other general types of association relationship. Lastly, the conceptual structure of roles is more complex. Relationships such as generalization can exist between roles, for example a CertifiedSurveyor as specialization of a Surveyor. But the existing model represents roles in a flat code list.

To develop the domain ontology that focuses on user roles, the paper introduces three new concepts, RolePlayer, Role, and Context, and two relations hasRole and dependsOn. The introduction brings the following three benefits:

i. Treating roles as a first class concept. Treating roles as concept allows the definition of role more specifically and flexibly.
ii. BAUnit is treated as RolePlayer to relate to Party, which is a subclass of Role, through hasRole relationship.
iii. Roles (as well as RolePlayer and Context) are allowed to be represented in hierarchy or ontology in its own.

With the formalized ontology in place, it allows a system to reason about and infer new knowledge using rule language such as Semantic Web Rule Language (SWRL) and Rule Interchange Format (RIF) for handling complex user conditions.

Representing Roles in Formalizing Domain Ontology for Land Administration

Kean Huat SOON, Singapore

1. INTRODUCTION

With the current Land Administration Domain Model (LADM) standard (ISO, 2012) that is modeled in Unified Modeling Language (UML) and additional explanatory natural text and tables, it will facilitate the software development and database design for the proper implementation of land administration systems. The use of UML supports generating a database schema or exchange format. However, UML is not meant to support machine reasoning.

To support machine reasoning, Web Ontology Language (OWL) is chosen to develop ontology. UML/OCL (Object Constraint Language) uses Closed World Assumption (CWA) while OWL works under Open World Assumption (OWA) (Zedlitz et. al., 2012). CWA treats all statements that are not mentioned as false, but OWA considers missing information as undecided and new knowledge can be inferred through reasoning. A formalized ontology in Web Ontology Language (OWL) supports inference and reasoning for information integration and automation. For example, supported with OWL ontology, a party who submits a survey plan can be inferred as a surveyor if he/she does not show any right, restriction, and responsibility (RRR) obligations on an administrative entity. (For other examples of how OWL supports information integration and automation in the geographic domain, the readers are referred to Lutz and Klien (2006), Fonseca et. al. (2002), Visser et. al. (2002) and Zhao et. al. (2008)).

The paper formalizes domain ontology for Land Administration from the natural language definitions in the standard. The natural texts are a good source to provide a neutral stance for developing the ontology without a prior assumption like CWA or OWA.

The ontology attempts to support land administration systems (e.g. web portal) that prioritize user roles in order to serve customers more proactively. A domain ontology is resulted from the paper to emphasize user roles in Land Administration.

The paper has two objectives. The first is to formalize the existing LADM model to ontology in OWL. The second is to enhance the formalized OWL ontology with role representation, which emphasizes user roles.

1.1 Motivation

Research on role representation in the Knowledge Representation domain has been a long tradition (e.g. Guarino, 1992; Mizoguchi et al, 2012; Sowa, 2000; Steimann, 2000). The representation of roles involves dynamics. Roles are context dependent and need individuals to play the roles, and an individual plays/unplays roles unknowingly in different times. Roles

are different from *functions*, *state* and they should not be merged with the concept of *interface* in programming (Loebe, 2003).

Currently however roles are rather represented statically in the existing LADM model. For instance in the existing model, role is considered as attribute in the Party class (see Figure 1). The Role attribute is defined by the Code List, LA_PartyRoleType. Such a definition has confined the way to model roles as context dependent. For example, Certified Surveyor and Surveyor are defined as two role types in the LA_PartyRoleType Code List. Although they both are similar, they can be differentiated through surveyor's certification for instance. The role of Certified Surveyor can only be played under the governance of certain legal institution like the Land Surveyors Board, and the person should have obtained the registered surveyor certificate. To relate only the role of Certified Surveyor to other concepts like Surveyor certificate or Land Surveyors Board is not part of the existing model.

Because BAUnit may play the role of Party, in the existing model Party and BAUnit are associated with relationship baunitAsParty (Figure 1). This association is treated the same as other general association relationships such as suBAUnit and unitRRR. Given that role is context dependent, this association is semantically different and should be treated in its own.

Defining roles as a code list assumes that the conceptual structure of roles is relatively flat. But the relationships between roles themselves are much more complex. For example, some have generalization relationship, such as bank and money provider, surveyor and certified surveyor. Such relationships unfortunately are not yet modeled in the current model. Research has been done on developing the ontology for roles or user actions (Hoekstra, 2010; Mizoguchi, et. al., 2012; Soon and Kuhn, 2004).

1.2 User-Centric Approach

In the paper roles are treated as a first class concept (or a class in OWL term). The *Role* concept is linked with another concept called *Context* through a relation *dependsOn* (as ObjectProperty in OWL) (literally means that a role depends upon a context). As an example, Party is considered as a subclass of Role, and the attributes from LA_PartyRoleType (e.g. conveyancer, surveyor) are subclasses of Party. Administrative Source and Spatial Source are subclasses of Context. To represent that the role of surveyor depends on spatial source for instance, the surveyor role can link with spatial source through dependsOn.

There is another concept called *RolePlayer* of which BAUnit is a subclass. RolePlayer relates to Role through *hasRole* relation. As a result, this approach introduces three new concepts, RolePlayer, Role, and Context, and two relations hasRole and dependsOn in order to develop the ontology.

This role representation model, which involves RolePlayer, Role and Context, is adapted from the mainstream of role representation research in Semantic Web and Knowledge Representation (e.g. Guarino, 1992; Sowa, 2000). For example, Loebe (2003) has used similar concepts *Filler*, *Role* and *Context* in the medical domain, while Kozaki et. al. (2007) has considered *Potential Player*, *Role concept* and *Context* for developing intelligent educational systems. The paper applies the similar model in the Land Administration domain.

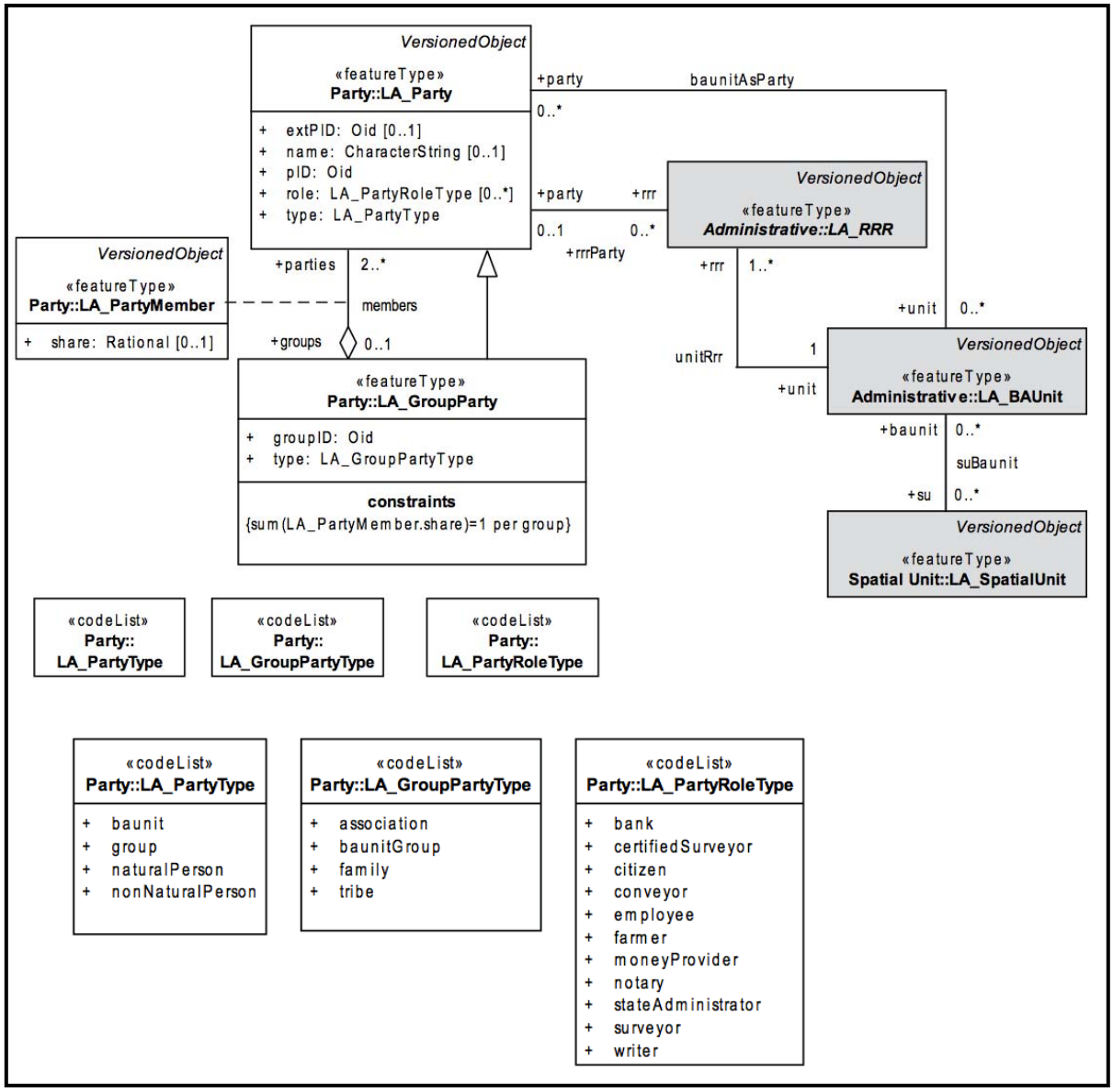

Figure 1. BAUnit is associated with Party through baunitAsParty, and roles are considered as Code List in the existing LADM model (ISO, 2012)

To sum up, the advantages of introducing role representation in the domain ontology are threefold.

i. Treating roles as a first class concept. Treating roles as concept allows the definition of role more specifically and flexibly. For example, the role of Certified Surveyor can be associated with registered surveyor certificate specifically.

ii. BAUnit is treated as RolePlayer together with other classes like NonNaturalPerson and NaturalPerson. BAUnit can be related to Party, which is a subclass of Role, through hasRole relationship.

iii. Roles (as well as RolePlayer and Context) are allowed to be represented in hierarchy or ontology in its own. For example, Money Provider and Surveyor can be defined as superclasses of Bank and Certified Surveyor respectively.

In what follows, Section 2 provides background information. Section 3 illustrates the process of formalizing the ontology from natural language texts. Section 4 demonstrates the addition of role representation in the ontology. Section 5 discusses the potential use of the ontology and Section 6 concludes the paper with future work.

2. FORMAL ONTOLOGY

2.1 Domain ontology

The term *ontology* is originated in philosophy to refer to the science of what is, i.e. the kinds and structures of objects, properties, events, processes, and relations in every area of reality (Agarwal, 2005; Mark et al., 2004). To construct an ontology for the geographic domain, the understanding for the ontological foundations of geographic data (Soon, 2010) is crucial.

Generally ontology can be classified into Top Level, Domain and Application ontologies (Boskovic et. al, 2010; Guarino, 1998; Sladić et. al, 2013). Top Level ontology depicts concepts at the highest level of a domain of discourse. It includes concepts like Space, Time, Process and Event. Meanwhile, Domain Ontology describes concepts that are commonly used within a particular domain such as Land Administration. Domain ontology facilitates automation, sharing and integration of information in a domain (Van Oosterom and Zlatanova, 2008). Lastly, Application ontology focuses on a particular application and concepts contained within this type of ontology are application specific. The ontology developed in the paper is a domain ontology.

Ontology is used to explicitly describe semantics by using OWL. OWL is enriched with axioms for semantic definitions to build ontology. By interpreting the knowledge in the ontology, a reasoner with Description Logics (Baader et. al., 2010) is able to make inference. OWL has been evolved from OWL 1 to OWL 2 with significant improvements. Two examples of the improvements are firstly one is able to define a class to be related to the class itself through `ObjectHasSelf`, and secondly OWL 2 supports user-defined data types, which is not possible in OWL 1. The paper is based on OWL 2, which will be described in the following section.

2.2 Web Ontology Language (OWL 2)

OWL is a World Wide Web Consortium standard and a "knowledge representation language, designed to formulate, exchange and reason with knowledge about a domain of interest" (W3C OWL Working Group, 2012). This section introduces some basic notions of OWL 2 to provide fundamental background. For more details of OWL 2, the readers are referred to OWL 2 Web Ontology Language (http://www.w3.org/TR/owl2-overview/) and subsequent links from the web site.

2.2.1 Basic Notions

OWL[1] has three basic entities to represent knowledge. These entities are *classes*, *properties*, and *individuals*. *Classes* refer to categories, such as Party. *Properties* refer to relationships or attributes, such as `hasPartyName`, which relates a party to a name. In this case, Party is the *domain* of the property `hasPartyName` or `DataPropertyDomain (:hasPartyName :Party)`, and string is the *range* or `DataPropertyRange (:hasPartyName xsd:string)`. There are two types of properties: ObjectProperty and DataProperty. ObjectProperty refers to the relationship between classes or between individuals. For instance, `hasPartyMembers` is an ObjectProperty, which can relate GroupParty to PartyMember which both are classes. DataProperty is used to relate a class (or individual) to a value (e.g. number). `HasPartyName`, as mentioned previously is a DataProperty. Different from OWL 1, OWL 2 supports user-defined DataProperty in addition to the existing built-ins such as integer and string. *Individuals* are instances of classes. An example of individuals is *Kean*, which is an instance of class Party.

Classes and properties can have hierarchy. To form a hierarchy, classes and properties can respectively use `subClassOf` and `subObjectPropertyOf`. For example, `subObjectPropertyOf (:hasRight :hasRRR)` means that whenever A has `hasRight` relationship with a basic administration unit, A also has `hasRRR` relationship.

`SubClassOf` and `subObjectPropertyOf` are called axioms. An axiom is a truth statement or proposition. For example, `subClassOf (:GroupParty :Party)` is a statement that says GroupParty is a Party. All necessary statements should be explicitly defined to ensure the completeness of the ontology.

There are more complex axioms supported in OWL. Examples are `EquivalentClasses`, `FunctionalObjectProperty`, `FunctionalDataProperty`, `ObjectIntersectionOf`, `ObjectUnionOf`, `ObjectAllValuesFrom` and `ObjectSomeValuesFrom`. `EquivalentClasses` is to state that two classes are equivalent, e.g. `EquivalentClasses (:Human :Person)`. `FunctionalObjectProperty` or `FunctionalDataProperty` refers to one and no more than one relationship/attribute. For example, `FunctionalDataProperty (:hasPartyName)` means Party can have one and only one name.

`ObjectIntersectionOf` and `ObjectUnionOf` respectively refer to *intersection* and *union* in Set Theory or Boolean operators *AND* and *OR*. For example, being father means he is a man and a parent or in OWL, `EquivalentClasses (:Father ObjectIntersectionOf(:Man :Parent))`. For `ObjectUnionOf`, one can use it to specify for instance being parent, it is `EquivalentClasses (:Parent ObjectUnionOf(:Mother :Father))`.

`ObjectAllValuesFrom` and `ObjectSomeValuesFrom` are treated as *universal quantification* (literally means "only" or "all") and *existential quantification* (literally means "at least one" or "some") respectively. `ObjectHasSelf` allows defining a class to be related to itself. For example, `(:BAUnit ObjectHasSelf (:hasRequiredRelationshipBAUnit))` means that

[1] For the sake of simplicity, OWL 2 is simply referred as OWL in the paper.

BAUnit is related to another BAUnit through `ObjectProperty hasRequiredRelationshipBAUnit.`

All classes, properties and individuals are called *resources* in OWL. Each of the resources has a unique Uniform Resource Identifier (URI)[2]. An example of URI for the LADM ontology is http://wiki.tudelft.nl/pub/Research/ISO19152/ImplementationMaterial/LADMOntology.owl.

2.3 Open World Assumption versus Closed World Assumption

Open World Assumption (OWA) assumes that the world has incomplete information. The statements that are not explicitly defined or cannot be inferred are not false, but undecided. Contrary, Closed World Assumption (CWA) assumes the world is complete; information that does not exist must be false (Zedlitz et. al, 2012). Semantic Web (Kolas et. al., 2005) and knowledge representation follow Open World Assumption, while software and database modelling supports Closed World Assumption. Because of the characteristics of OWA of being open, OWA has the capability to reveal new knowledge. In contrast, CWA supports consistency checking through constraints.

To illustrate the difference further, consider that a statement is explicitly defined in a system as "serving parcel 123 is a basic administration unit." If one was to ask "is serving parcel 123 a party?", under CWA, the answer is "no", but under OWA, it is "do not know". The "no" answer is owing to the fact that serving parcel 123 has been defined as a basic administration unit and not party. Other than basic administration unit, all are considered false or violate the statement in the system. For OWA, so long as the relationship between party and basic administration unit is not defined, the answer will always be "do not know", and possibly the system will infer that both party and basic administration unit are the same. Until when party and basic administration unit are defined as two different entities, the answer is then "no".

UML/OCL follows CWA while OWL applies OWA. Taking the example from the existing LADM model, an invariant such as `{Party can only have 0 RRR in case the party has specific role}` has been defined. If a database that has applied this invariant will be violated if the data that contain party information do not have related information about RRR and Role and that attempt to load in to the database. In contrast, if an ontology has defined a person that has spatial source and that does not have RRR as a surveyor, then when a person is detected to have spatial source and no related RRR, a reasoner would automatically infer that person as a surveyor.

3. FORMALIZING ONTOLOGY FROM NATURAL LANGUAGE

The domain ontology in OWL is created using Protégé 4.3 (http://protégé.stanford.edu/), an open source ontology editor from Stanford University. This section will describe how the ontology is created from natural language texts. The complete ontology is made available at

[2] Strictly speaking, OWL supports Internationalized Resource Identifier (IRI), which can contain universal character set including Chinese and Japanese in addition to the ASCII character set (e.g. A-Z Latin alphabets), which URI can only contain. Owing to the paper does not involve universal character set like Chinese, we simply use URI here.

www.isoladm.org (hosted under the ImplementationMaterial link). Next section will illustrate the addition of role representation in the ontology. In the following sections, the extracted texts from the standard are highlighted in italics, while the formalizations are described in functional syntax for human readability (highlighted in `Courier` font).

3.1 All Classes are Versioned Objects

Like the existing LADM model, VersionedObject is also defined in the ontology as the top-level class from where all classes in the ontology are connected directly or indirectly. This means that all classes in the domain ontology are versioned objects, which have data properties hasBeginLifeSpanVersion and hasEndLifeSpanVersion.

```
Declaration ((DataProperty (:hasBeginLifeSpanVersion)))
ObjectPropertyDomain (:hasBeginLifeSpanVersion :VersionedObject)
ObjectPropertyRange (:hasBeginLifeSpanVersion xsd:dateTime)

Declaration ((DataProperty (:hasEndLifeSpanVersion)))
ObjectPropertyDomain (:hasEndLifeSpanVersion :VersionedObject)
ObjectPropertyRange (:hasEndLifeSpanVersion xsd:dateTime)
```

The direct classes to VersionedObject are Party, PartyMember, RRR, BAUnit, SpatialUnit, SpatialUnitGroup, BoundaryFace, BoundaryFaceString, Point and Level, which are shown below. The rest of classes are indirectly connected to VersionedObject.

```
SubClassOf (:Party :VersionedObject)
SubClassOf (:PartyMember :VersionedObject)
SubClassOf (:RRR :VersionedObject)
SubClassOf (:BAUnit :VersionedObject)
SubClassOf (:SpatialUnit :VersionedObject)
SubClassOf (:SpatialUnitGroup :VersionedObject)
SubClassOf (:BoundaryFace :VersionedObject)
SubClassOf (:BoundaryFaceString :VersionedObject)
SubClassOf (:Point :VersionedObject)
SubClassOf (:Level :VersionedObject)
```

3.2 Formalizations

To develop the ontology, the natural language texts are extracted based on the definitions on the four basic classes: Party, RRR, BAUnit and SpatialUnit. Note that due to the space limit, not all extracted texts are shown here, this section only illustrates the prominent ones.

To get started, let us begin with Party class,

LA_Party has a specialization: LA_GroupParty (with group party as an instance). Between LA_Party and LA_GroupParty there is an optional association class: LA_PartyMember (with party member as an instance). A group party, being a specialization of party, is also a party. Every party, being a constituent of a group party, may then be registered as a party member of class LA_PartyMember

The corresponding formalization in OWL with cardinality is as follows,

```
Declaration ((Class (:Party)))
Declaration ((Class (:GroupParty)))
```

```
Declaration ((Class (:PartyMember)))

SubClassOf (:GroupParty :Party)

Declaration ((ObjectProperty (:hasPartyMembers)))
ObjectPropertyDomain (:hasPartyMembers :GroupParty)
ObjectPropertyRange (:hasPartyMembers :PartyMember)
ObjectMinCardinality (2 :hasPartyMembers :PartyMember)

Declaration ((ObjectProperty (:isRegisteredAs)))
ObjectPropertyDomain (:isRegisteredAs :Party)
ObjectPropertyRange (:isRegisteredAs :PartyMember)
```

For RRR class,

Class LA_RRR is an abstract class. An instance of a subclass of LA_RRR is a right (or social tenure relationship), a restriction, or a responsibility.

```
Declaration ((Class (:RRR)))
Declaration ((Class (:Right)))
Declaration ((Class (:Restriction)))
Declaration ((Class (:Responsibility)))

SubClassOf (:Right :RRR)
SubClassOf (:Restriction :RRR)
SubClassOf (:Responsibility :RRR)
```

and,

A party is associated to zero or more [0..] instances of a subclass of LA_RRR.*

```
Declaration ((ObjectProperty (:hasRRR)))
ObjectPropertyDomain (:hasRRR :Party)
ObjectPropertyRange (:hasRRR :RRR)

Declaration ((ObjectProperty (:hasRight)))
ObjectPropertyDomain (:hasRight :Party)
ObjectPropertyRange (:hasRight :Right)

Declaration ((ObjectProperty (:hasRestriction)))
ObjectPropertyDomain (:hasRestriction :Party)
ObjectPropertyRange (:hasRestriction :Restriction)

Declaration ((ObjectProperty (:hasResponsibility)))
ObjectPropertyDomain (:hasResponsibility :Party)
ObjectPropertyRange (:hasResponsibility :Responsibility)

SubObjectPropertyOf (:hasRight :hasRRR)
SubObjectPropertyOf (:hasRestriction :hasRRR)
SubObjectPropertyOf (:hasResponsibility :hasRRR)
```

Specifically,

If it is a right or responsibility, then it is associated to exactly one [1] party, and exactly one [1] basic administrative unit. If it is a restriction, then it is associated to zero or one [0..1] parties, and exactly one [1] basic administrative unit. The latter allows for the registration of restrictions (e.g. right-of-way, right-to-harvest-fruit), with, or without an association to LA_Party.

```
InverseObjectProperties (:hasRight :hasRightParty)
ObjectExactCardinality (1 :hasRightParty :Party)

InverseObjectProperties (:hasResponsibility :hasResponsibilityParty)
ObjectExactCardinality (1 :hasResponsibilityParty :Party)

InverseObjectProperties (:hasRestriction :hasRestrictionParty)
ObjectMinCardinality (0 :hasRestrictionParty :Party)
ObjectMaxCardinality (1 :hasRestrictionParty :Party)
```

and the relationships between RRR and BAUnit,

```
Declaration ((ObjectProperty (:hasRRROnBAUnit)))
ObjectPropertyDomain (:hasRRROnBAUnit :RRR)
ObjectPropertyRange (:hasRRROnBAUnit :BAUnit)

FunctionalObjectProperty (:hasRRROnBAUnit)

Declaration ((ObjectProperty (:hasRightOnBAUnit)))
ObjectPropertyDomain (:hasRightOnBAUnit :Right)
ObjectPropertyRange (:hasRightOnBAUnit :BAUnit)

Declaration ((ObjectProperty (:hasRestrictionOnBAUnit)))
ObjectPropertyDomain (:hasRestrictionOnBAUnit :Restriction)
ObjectPropertyRange (:hasRestrictionOnBAUnit :BAUnit)

Declaration ((ObjectProperty (:hasResponsibilityOnBAUnit)))
ObjectPropertyDomain (:hasResponsibilityOnBAUnit :Responsibility)
ObjectPropertyRange (:hasResponsibilityOnBAUnit :BAUnit)

SubObjectPropertyOf (:hasRightOnBAUnit :hasRRROnBAUnit)
SubObjectPropertyOf (:hasRestrictionOnBAUnit :hasRRROnBAUnit)
SubObjectPropertyOf (:hasResponsibilityOnBAUnit :hasRRROnBAUnit)
```

Then the relationships between RRR and AdminSource (Administrative Source),

An instance of a subclass of LA_RRR shall be associated to one or more [1..] administrative sources (i.e. the right, restriction or responsibility affecting a basic administrative unit should be supported by at least one administrative source).*

```
Declaration ((Class (:AdminSource)))

Declaration ((ObjectProperty (:isSupportedBy)))
ObjectPropertyDomain (:isSupportedBy :RRR)
ObjectPropertyRange (:isSupportedBy :AdminSource)
```

```
ObjectMinCardinality (1 :isSupportedBy :AdminSource)
```

For the BAUnit Class,

A basic administrative unit is associated to zero or more [0..] spatial units;*

```
Declaration ((Class (:SpatialUnit)))

Declaration ((ObjectProperty (:hasBASpatialUnit)))
ObjectPropertyDomain (:hasBASpatialUnit :BAUnit)
ObjectPropertyRange (:hasBASpatialUnit :SpatialUnit)
```

Reversely, the relationships between BAUnit and RRR,

A basic administrative unit should be associated to one or more ([1..] instances of right, restriction or responsibility (i.e. a basic administrative unit cannot exist if there is not at least one right, restriction or responsibility associated to it).*

```
InverseObjectProperties (:hasRRROnBAUnit :hasBAUnitRRR)
ObjectMinCardinality (1 :hasBAUnitRRR :RRR)

SubObjectPropertyOf (:hasBAUnitRight :hasBAUnitRRR)
SubObjectPropertyOf (:hasBAUnitRestriction :hasBAUnitRRR)
SubObjectPropertyOf (:hasBAUnitResponsibility :hasBAUnitRRR)
```

The relationships between BAUnit themselves,

A basic administrative unit can be spatially related, through a required relationship, to zero or more [0..] other basic administrative units.*

```
SubClassOf (:BAUnit ObjectHasSelf (:hasRequiredRelationshipBAUnit))
```

The relationships between BAUnit and AdminSource,

A basic administrative unit can be associated to zero or more [0..] administrative sources (i.e. the basic administrative unit is usually described as the object affected by the right, restriction or responsibility in the administrative source).*

```
Declaration ((ObjectProperty (:hasBAUnitAdminSource)))
ObjectPropertyDomain (:hasBAUnitAdminSource :BAUnit)
ObjectPropertyRange (:hasBAUnitAdminSource :AdminSource)
```

The relationships between BAUnit and SpatialSource,

A basic administrative unit can be associated to zero or more [0..] spatial sources (i.e. the extent – part of – of a basic administrative unit can be described on a spatial source).*

```
Declaration ((Class (:SpatialSource)))

Declaration ((ObjectProperty (:hasBAUnitSpatialSource)))
```

```
ObjectPropertyDomain (:hasBAUnitSpatialSource :BAUnit)
ObjectPropertyRange (:hasBAUnitSpatialSource :SpatialSource)
```

The relationships between Source, AdminSource and SpatialSource,

In the LADM, administrative sources and spatial sources are modelled, starting with an abstract class LA_Source. LA_Source has two subclasses: (1) LA_AdministrativeSource […], and (2) LA_SpatialSource.

```
Declaration ((Class (:Source)))
SubClassOf (:AdminSource :Source)
SubClassOf (:SpatialSource :Source)
```

Reversely, the relationships from AdminSource to Party, BAUnit and RRR,

An administrative source should be associated to one or more [1..] parties;*

```
Declaration ((ObjectProperty (:hasAdminSourceParty)))
ObjectPropertyDomain(:hasAdminSourceParty :AdminSource)
ObjectPropertyRange(:hasAdminSourceParty :Party)
ObjectMinCardinality (1 :hasAdminSourceParty :Party)
```

An administrative source may be associated to zero or more [0..] basic administrative units;*

```
Declaration ((ObjectProperty (:hasAdminSourceBAUnit)))
ObjectPropertyDomain(:hasAdminSourceBAUnit :AdminSource)
ObjectPropertyRange(:hasAdminSourceBAUnit :BAUnit)
```

An administrative source may be associated to zero or more [0..] instances of specializations (right, restriction/mortgage, and responsibility) of LA_RRR;*

```
Declaration ((ObjectProperty (:hasAdminSourceRRR)))
ObjectPropertyDomain(:hasAdminSourceRRR :AdminSource)
ObjectPropertyRange(:hasAdminSourceRRR :RRR)
```

The relationships between SpatialSource and Point,

A spatial source should be associated to one or more [1..] points (i.e. the spatial source describes in all cases one or more points.*

```
Declaration ((Class (:Point)))

Declaration ((ObjectProperty (:describesPoint)))
ObjectPropertyDomain(:describesPoint :SpatialSource)
ObjectPropertyRange(:describesPoint :Point)
ObjectMinCardinality (1 :describesPoint :Point)
```

The relationships between SpatialSource and Boundary Face String,

A spatial source may be associated to zero or more [0..] boundary face strings;*

```
Declaration ((Class (:BoundaryFaceString)))
```

```
Declaration ((ObjectProperty (:describesBFaceString)))
ObjectPropertyDomain(:describesBFaceString :SpatialSource)
ObjectPropertyRange(:describesBFaceString :BoundaryFaceString)
```

The relationships between SpatialSource and Boundary Face,

A spatial source may be associated to zero or more [0..] boundary faces;*

```
Declaration ((Class (:BoundaryFace)))

Declaration ((ObjectProperty (:describesBFace)))
ObjectPropertyDomain(:describesBFace :SpatialSource)
ObjectPropertyRange(describesBFace :BoundaryFace)
```

The relationships between SpatialSource and SpatialUnit,

A spatial source may be associated to zero or more [0..] spatial units;*

```
Declaration ((ObjectProperty (:describesSpatialExtent)))
ObjectPropertyDomain(:describesSpatialExtent :SpatialSource)
ObjectPropertyRange(:describesSpatialExtent :SpatialUnit)
```

The relationships between SpatialSource and BAUnit,

A spatial source may be associated to zero or more [0..] basic administrative units;*

```
InverseObjectProperties(:hasBAUnitSpatialSource :hasSpatialSourceBAUnit)
```

Finally, the relationships between SpatialSource and Party,

A spatial source should be associated to one or more [1..] parties;*

```
Declaration ((ObjectProperty (:hasSpatialSourceParty)))
ObjectPropertyDomain(:hasSpatialSourceParty :SpatialSource)
ObjectPropertyRange(:hasSpatialSourceParty :Party)
ObjectMinCardinality (1 :hasSpatialSourceParty :Party)
```

4. REPRESENTING ROLES IN THE ONTOLOGY

To represent roles in the ontology, three new classes, which are RolePlayer, Role, and Context, are added in the ontology.

```
Declaration ((Class (:RolePlayer)))
Declaration ((Class (:Role)))
Declaration ((Class (:Context)))
```

RolePlayer, Role and Context are also VersionedObject.

```
SubClassOf (:RolePlayer :VersionedObject)
SubClassOf (:Role :VersionedObject)
SubClassOf (:Context :VersionedObject)
```

Object properties dependsOn and hasRole have also been created to relate respectively between RolePlayer and Role, and between Role and Context as illustrated below.

```
Declaration ((ObjectProperty (:hasRole)))
ObjectPropertyDomain (:hasRole :RolePlayer)
ObjectPropertyRange (:hasRole :Role)

Declaration ((ObjectProperty (:dependsOn)))
ObjectPropertyDomain (:dependsOn :Role)
ObjectPropertyRange (:dependsOn :Context)
```

In the ontology, together with other PartyTypes (e.g. NaturalPerson), BAUnit is a subclass of RolePlayer, which can have relation hasRole with Role, which in turn has subclass Party. This literally means that BAUnit has role as Party.

```
SubClassOf (:BAUnit :RolePlayer)
SubClassOf (:NaturalPerson :RolePlayer)
SubClassOf (:NonNaturalPerson :RolePlayer)
SubClassOf (:Group :RolePlayer)
```

Subclasses of Role are defined as follows. The attributes under the PartyRoleType code list have been redefined in a hierarchical structure. For example, Surveyor and MoneyProvider have been defined as subclasses of Party, and CertifiedSurveyor and Bank are subclasses of Surveyor and MoneyProvider respectively. This definition supports inference. For example when Surveyor class is defined to have required SpatialSource, all its subclasses, such as CertifiedSurveyor, can be inferred to have SpatialSource. Explicit statements to define individual types of surveyor to have required SpatialSource are not required.

```
SubClassOf (:Party :Role)

SubClassOf (:Surveyor :Party)
SubClassOf (:MoneyProvider :Party)
SubClassOf (:Conveyancer :Party)
SubClassOf (:Writer :Party)
SubClassOf (:Citizen :Party)
SubClassOf (:Employee :Party)
SubClassOf (:StateAdministrator :Party)
SubClassOf (:Notary :Party)
SubClassOf (:Farmer :Party)

SubClassOf (:Bank :MoneyProvider)
SubClassOf (:CertifiedSurveyor :Surveyor)
```

The subclasses of Context class are as follows,

```
SubClassOf (:AdminSource :Context)
SubClassOf (:SpatialSource :Context)
```

which means that both AdminSource and SpatialSource are Context, and these contexts determine the roles.

As a result, the relationships between RolePlayer, Role and Context can be demonstrated in Figure 2 using OntoGraf, a visualization plug-in in Protégé. In the figure, the dashed arrow from RolePlayer to Role represents hasRole relation, while the dashed arrow from Role to Context describes dependsOn relation. The rest of arrows depict hasSubClass relation.

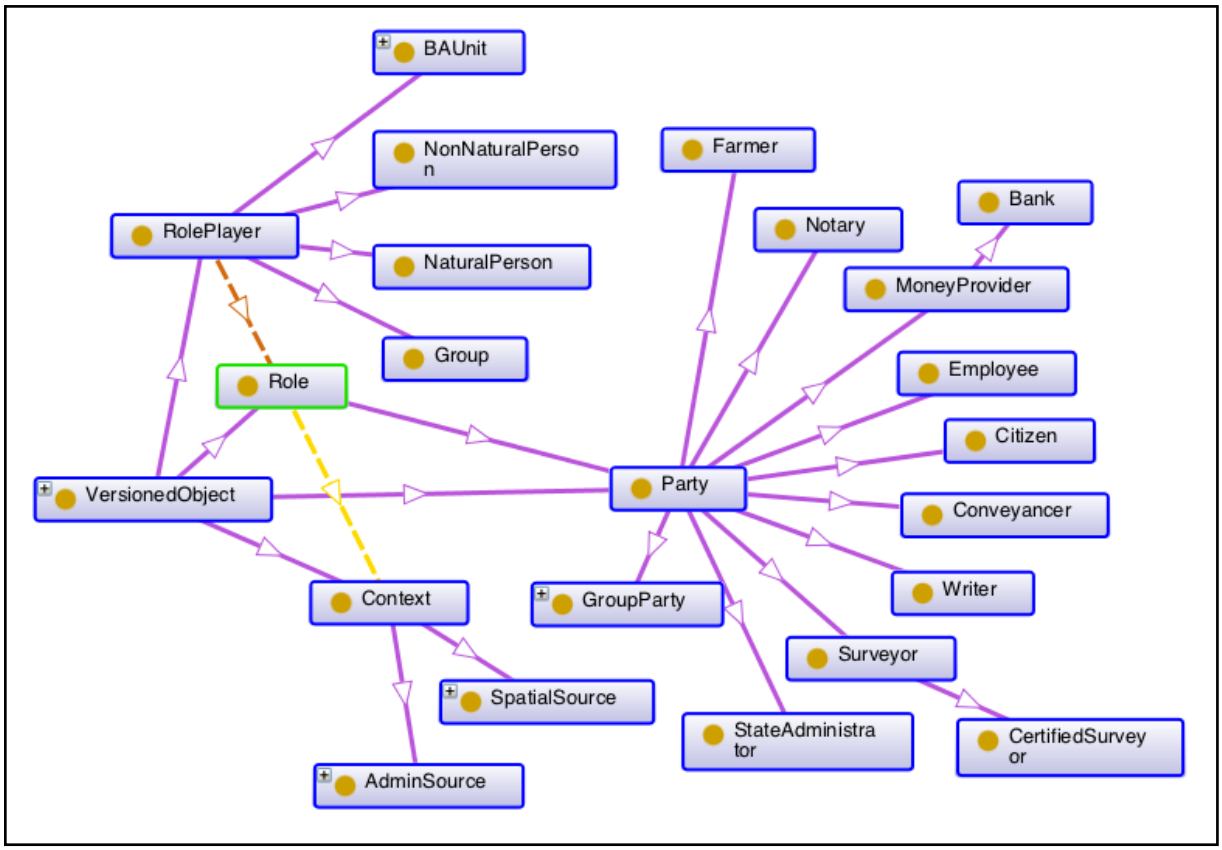

Figure 2. The interrelationships between RolePlayer, Role and Context

Overall the resulted domain ontology focusing on user roles can be visualized in Figure 3 (a). The hierarchical structure of the ontology is illustrated in Figure 3 (b) with new classes highlighted. Figure 3 (c) shows the ObjectProperties used in the ontology.

5. APPLICATIONS

With the formalized ontology in place, it allows a system to reason about and infer new knowledge using rule language such as Semantic Web Rule Language (SWRL) (Horrocks et. al., 2004) and Rule Interchange Format (RIF) (De Bruijn and Welty, 2013). SWRL is a rule extension to OWL and is developed based on the same logics used in constructing OWL. To illustrate a simple example for the potential use of the ontology, the following SWRL syntax defines the condition where if someone, who is a party and has administrative source and no any related RRR, can be inferred as a conveyancer,

```
Party(?x)^(hasRRR=0)(?x)^dependsOn(?x,?y)^AdminSource(?y)
->Conveyancer(?x)
```

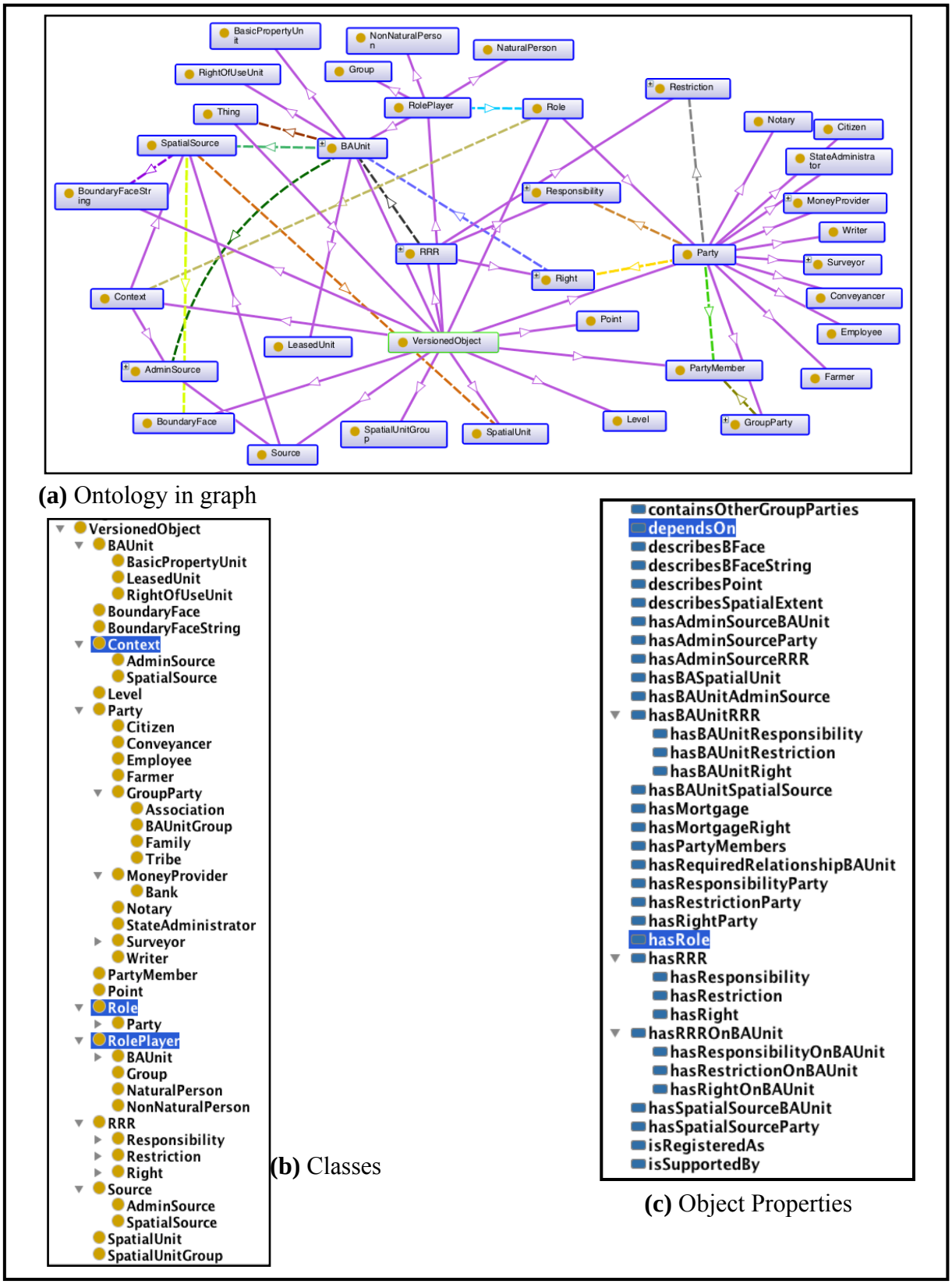

Figure 3. The formalized domain ontology focusing on user roles for land administration

Or else, if he/she has spatial source, the system will then consider him/her as a surveyor.

```
Party(?x)^(hasRRR=0)(?x)^dependsOn(?x,?y)^SpatialSource(?y)->Surveyor(?x)
```

More rules can be defined to address more complicated conditions, which can include more classes and relationships from the formalized ontology. The above mentioned just illustrates a simple one.

As mentioned earlier, the formalized domain ontology attempts to be used for supporting automation in the Land Administration domain. In Land Administration, web portals have been developed to support various customers on property transactions, applications for registration of land titles, submission of survey plans for authority's approval, etc. Examples of these web portals include SPEAR (Surveying and Planning through Electronic Applications and Referrals, http://www.spear.land.vic.gov.au/) from Victoria, Australia, LandOnline (http://www.landonline.govt.nz) from LINZ (Land Information New Zealand), New Zealand, and STARS (Singapore Titles Automated Registration System, https://www.stars.gov.sg) from Singapore. The user groups of these portals are huge and range from various parties, such as surveyors, government authorities, land owners, member of the public, and lawyers. The formalized ontology will help to identify the role of a user by reasoning about the documents/information the user submitted. The ability of intelligently inferring user roles will allow the system to serve customers more proactively.
Or else, if he/she has spatial source, the system will then consider him/her as a surveyor.

```
Party(?x)^(hasRRR=0)(?x)^dependsOn(?x,?y)^SpatialSource(?y)->Surveyor(?x)
```

More rules can be defined to address more complicated conditions, which can include more classes and relationships from the formalized ontology. The above mentioned just illustrates a simple one.

As mentioned earlier, the formalized domain ontology attempts to be used for supporting automation in the Land Administration domain. In Land Administration, web portals have been developed to support various customers on property transactions, applications for registration of land titles, submission of survey plans for authority's approval, etc. Examples of these web portals include SPEAR (Surveying and Planning through Electronic Applications and Referrals, http://www.spear.land.vic.gov.au/) from Victoria, Australia, LandOnline (http://www.landonline.govt.nz) from LINZ (Land Information New Zealand), New Zealand, and STARS (Singapore Titles Automated Registration System, https://www.stars.gov.sg) from Singapore. The user groups of these portals are huge and range from various parties, such as surveyors, government authorities, land owners, member of the public, and lawyers. The formalized ontology will help to identify the role of a user by reasoning about the documents/information the user submitted. The ability of intelligently inferring user roles will allow the system to serve customers more proactively.

6. CONCLUSIONS

The paper has demonstrated the formalization of domain ontology from natural language for Land Administration. In order to build the domain ontology to emphasize user roles, additional classes and relationships have been added. The ontology attempts to support land administration systems that aim to serve customers more proactively for land administration routine processes such as registrations of land titles and submissions of survey plans.

The development illustrated in the paper, however, is just an initial step to support automation in land administration. One immediate step to further the research is to develop rules to capture and represent more complex conditions for handling different user situations. To support automation in cadastral job processing, countries like Australia, New Zealand and Singapore have considered LandXML as a national cadastral standard to capture land surveying information, such as traverse, parcels, surveyor details, etc. With the integration from OWL and LandXML (Soon, 2012), the use of rule language, such as SWRL and RIF is expected to raise the level of automation in land administration processes.

Another area to look at is the temporal constraints and relationships in the ontology. The temporal aspect was not considered in the paper although all classes in the ontology are versioned objects. The DataProperties related to time should be fully defined. The temporal rules can then be constructed to reason about the classes in the ontology through the defined DataProperties.

ACKNOWLEDGEMENTS

Invaluable comments and suggestions from the seven anonymous reviewers are greatly appreciated. I am also very grateful to Prof Peter Van Oosterom for his feedback and supports in creating the final paper. I would also like to thank the Singapore Land Authority (SLA) for the financial support during the LADM Workshop.

REFERENCES

Agarwal, P. (2005). Ontological Considerations in GIScience. In: *International Journal of Geographical Information Science*, 19(5): 501-536. Taylor & Francis.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P.F. (2010). *The Description Logic Handbook: Theory, Implementation and Applications, 2nd Edition*. Cambridge University Press.

Boskovic, D., Ristić, A., Govedarica, M., & Pržulj, D. (2010). Ontology Development for Land Administration. In *Proceedings of 8th International Symposium on Intelligent Systems and Informatics (SISY)*, 437-442. IEEE.

De Bruijn, J. and Welty, C. (2013). RIF RDF and OWL Compatibility (Second Edition). Retrieved July 10, 2013, from http://www.w3.org/TR/2013/REC-rif-rdf-owl-20130205/.

Fonseca, F., Egenhofer, M., Agouris, P. and Camara, G. (2002). Using Ontologies for Integrated Geographic Information Systems. In: *Transactions in GIS*, 6(3): 231-257. Wiley.

Guarino, N. (1992). Concepts, Attributes and Arbitrary Relations: Some Linguistic and Ontological Criteria for Structuring Knowledge Bases. In *Data & Knowledge Engineering*, *8*(3), 249-261. Elsevier.

Guarino, N. (1998). Formal Ontology and Information Systems. In *Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998*, 3-15. IOS Press.

Hoekstra, R. (2010). Representing Social Reality in OWL 2. In *Proceedings of the 7th International Workshop on OWL: Experiences and Directions (OWLED 2010),* San Francisco, California, USA, June 21-22, 2010.

Horrocks, I., Patel-Schneider P. F., Boley, H., Tabet, S., Grosof, B. and Dean, M. (2004). SWRL: A Semantic Web Rule Language: Combining OWL and RuleML. Retrieved July 10, 2013, from http://www.w3.org/Submission/SWRL/.

ISO (2012). *Geographic Information – Land Administration Domain Model (LADM)*. ISO 19152:2012(E). International Organization for Standardization (ISO). Geneva, Switzerland.

Kolas, D., Hebeler, J. and Dean, M. (2005). Geospatial Semantic Web: Architecture of Ontologies. In: Rodriguez, M. A., Cruz, I., Levashkin, S. and Egenhofer, M. (Eds): *Geospatial Semantics 2005*, Lecture Notes in Computer Science, 3799: 183-194. Springer.

Kozaki, K., Sunagawa, E., Kitamura, Y., & Mizoguchi, R. (2007). Role Representation Model Using OWL and SWRL. In *Proc. of 2nd Workshop on Roles and Relationships in Object Oriented Programming, Multiagent Systems, and Ontologies*, 39-46.

Loebe, F. (2003). *An Analysis of Roles: Towards Ontology-Based Modeling*. Master Thesis. University of Leipzig, Germany.

Lutz, M. and Klien, E. (2006). Ontology-Based Retrieval of Geographic Information. In *International Journal of Geographical Information Science*, 20(3): 233–260. Taylor & Francis.

Mark, D., Smith, B., Egenhofer, M., and Hirtle, S. (2004). Ontological Foundations for Geographic Information Science. In McMaster, R. B., and Usery, E. L. (Eds), *A Research Agenda for Geographic Information Science,* 335-351. CRC Press.

Mizoguchi, R., Kozaki, K., and Kitamura, Y. (2012). Ontological Analyses of Roles. In *Proceedings of 2012 Federated Conference on Computer Science and Information Systems (FedCSIS),* 489-496. IEEE.

Sladić, D., Govedarica, M., Pržulj, D., Radulović, A., & Jovanović, D. (2013). Ontology for Real Estate Cadastre. In *Survey Review*. Maney Publishing. (In publication)

Soon, K. and Kuhn, W. (2004). Formalizing User Actions for Ontologies. In: Egenhofer, M., Freksa, C. and Miller (Eds): *GIScience 2004*, Lecture Notes in Computer Science, 3234: 299-312. Maryland, US. Springer-Verlag.

Soon, K. H. (2010). Ontological Foundations of Geographical Data. In: Warf, B. (Ed.). *Encyclopedia of Geography*: 4. SAGE Publications.

Soon, K. H. (2012). A Conceptual Framework of Representing Semantics for 3D Cadastre in Singapore. In: *Proceedings of the 3rd International Workshop on 3D Cadastres*. Shenzhen, China. FIG.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations* (Vol. 13). Pacific Grove: Brooks/Cole.

Steimann, F. (2000). On the Representation of Roles in Object-Oriented and Conceptual Modelling. In *Data and Knowledge Engineering*, 35(1):83–106. Elsevier.

Van Oosterom, P. and Zlatanova, S. (2008). *Creating Spatial Information Infrastructures: Toward the Spatial Semantic Web*. CRC Press.

Visser, U., Stuckenschmidt, H., Schuster, G., and Vogele, T. (2002). Ontologies for Geographic Information Processing. In *Computers & Geosciences*, 28(1):103–117. Elsevier.

W3C OWL Working Group (2012). OWL 2 Web Ontology Language: Document Overview (Second Edition). Retrieved July 10, 2013, from http://www.w3.org/TR/owl2-overview/.

Zedlitz, J., Jörke, J. and Luttenberger, N. (2012). From UML to OWL 2. *Knowledge Technology*, 154-163. Springer Berlin Heidelberg.

Zhao, T., Zhang, C., Wei, M., and Peng, Z. (2008). Ontology-Based Geospatial Data Query and Integration. In *Geographic Information Science*, 370-392. Springer.

BIOGRAPHICAL NOTES

Kean Huat Soon is a Senior Surveyor at the Land Survey Division of Singapore Land Authority. He earned a Msc in Geography from the Pennsylvania State University, with focus on geospatial ontology development. His research interests include semantic interoperability, data modeling, cadastral information system and ontology.

CONTACTS

Kean Huat SOON
Singapore Land Authority
55 Newton Road, #12-01
Revenue House
Singapore 307987
SINGAPORE
Phone: +65 6478 3537
Fax: +65 6323 9937
E-mail: soon_kean_huat@sla.gov.sg
Website: http://www.sla.gov.sg/