



The Organization Ontology

W3C Recommendation 16 January 2014

This version:

<http://www.w3.org/TR/2014/REC-vocab-org-20140116/>

Latest published version:

<http://www.w3.org/TR/vocab-org/>

Implementation report:

http://www.w3.org/2011/gld/wiki/ORG_Implementations

Previous version:

<http://www.w3.org/TR/2013/PR-vocab-org-20131217/>

Editor:

Dave Reynolds, [Epimorphics Ltd.](#)

Please refer to the [errata](#), a list of issues with this document discovered after publication.

This document is also available in this non-normative format: [diff to previous version](#)

The English version of this specification is the only normative version. Non-normative [translations](#) may also be available.

Copyright © 2012-2014 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

Abstract

This document describes a core ontology for organizational structures, aimed at supporting linked data publishing of organizational information across a number of domains. It is designed to allow domain-specific extensions to add classification of organizations and roles, as well as extensions to support neighbouring information such as organizational activities.

The namespace for all terms in this ontology is: <http://www.w3.org/ns/org#>

The vocabulary defined in this document is also available in these non-normative formats: [RDF/XML](#) and [Turtle](#).

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](#) at <http://www.w3.org/TR/>.

This document has been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and is endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

This ontology was originally developed [and published](#) outside of W3C, but has been extended and further developed within the Government Linked Data Working Group.

This document was published by the [Government Linked Data Working Group](#) as a Recommendation. If you wish to make comments regarding this document, please send them to public-gld-comments@w3.org ([subscribe](#), [archives](#)). All comments are welcome.

Please see the Working Group's [implementation report](#).

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](#). W3C maintains a [public list of any patent disclosures](#) made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains [Essential Claim\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

Table of Contents

1. Overview of ontology
 - 1.1 Example
2. Description and commentary
 - 2.1 Organizational structure
 - 2.2 Membership and Reporting structure
 - 2.3 Location information
 - 2.4 Organizational history
 - 2.5 Notes on modelling style
3. Conformance
4. Namespaces
5. Ontology Reference
 - 5.1 Index of classes and properties
 - 5.2 Organizational structure
 - 5.2.1 Class: Organization
 - 5.2.2 Class: FormalOrganization
 - 5.2.3 Class: OrganizationalUnit
 - 5.3 Membership, roles, posts and reporting
 - 5.3.1 Property: memberOf
 - 5.3.2 Property: hasMember

- 5.3.3 [Property: headOf](#)
- 5.3.4 [Class: Membership](#)
- 5.3.5 [Class: Role](#)
- 5.3.6 [Class: Post](#)
- 5.3.7 [Property: reportsTo](#)
- 5.4 [Location](#)
 - 5.4.1 [Class: Site](#)
 - 5.4.2 [Property: location](#)
- 5.5 [Projects and other activities](#)
 - 5.5.1 [Class: OrganizationalCollaboration](#)
- 5.6 [Historical information](#)
 - 5.6.1 [Class: ChangeEvent](#)
- A. [Change history](#)
- B. [Acknowledgements](#)
- C. [References](#)
 - C.1 [Normative references](#)
 - C.2 [Informative references](#)

Introduction

This document describes a core ontology (ORG) for organizational structures, aimed at supporting linked data publishing of organizational information across a number of domains. It is designed to allow domain-specific extensions to add classification of organizations and roles, as well as extensions to support neighbouring information such as organizational activities.

This document does not prescribe any particular method of deploying data expressed in ORG. ORG is applicable in many contexts including RDF accessible via SPARQL endpoints, embedded in HTML pages, or serialized as an RDF/XML or Turtle. The examples in this document use Turtle [[turtle](#)] in the interests of readability.

1. Overview of ontology

This section is non-normative.

This ontology is designed to enable publication of information on organizations and organizational structures including governmental organizations. It is intended to provide a generic, reusable core ontology that can be extended or specialized for use in particular situations.

The ontology gives terms to support the representation of:

- [organizational structure](#)
 - notion of an organization
 - decomposition into sub-organizations and units
 - purpose and classification of organizations
- [reporting structure](#)
 - membership and reporting structure within an organization
 - roles, posts, and the relationship between people and organizations
- [location information](#)
 - sites or buildings, locations within sites
- [organizational history \(merger, renaming\)](#)

This coverage corresponds to the type of information typically found in organizational charts. As such it does not offer a complete representation for all the nuances of organizational control structures and flows of accountability and empowerment. Developers are encouraged to create extension vocabularies for such purposes, building upon this generic foundation.

The ontology does not provide category structures for organization type, organization purpose or roles. Different domains will have different requirements for classification of such concepts. Instead the ontology provides just the core base concepts needed to allow extensions to add specific sub-class structures or classification schemes as required. Users of the ontology are encouraged to define **profiles** which strengthen interoperability by specifying particular controlled vocabularies to use for these concepts.

A pictorial illustration of the main classes and relationships in ORG is shown below. All terms are within the ORG namespace (<http://www.w3.org/ns/org#>, preferred prefix *org*;) unless an explicit prefix is given. The namespaces for all referenced vocabularies are given in the section on [Namespaces](#).

In many organizations there is a hierarchy of unit structures. For example we might see a containment hierarchy like:

```
Corporation
  BusinessUnit
    Division
      Function
```

Such hierarchies are specific to the particular organization, or class of organization being modelled. Profiles of ORG may include sub-classes of [org:Organization](#) and [org:OrganizationalUnit](#) to represent such structures and specialize or restrict the use of [org:hasSubOrganization](#) to match the desired hierarchy.

Organizational classification

In a number of circumstances we wish to classify organizations. There are many approaches that could be taken for this. It can be based on the legal structure under which the organization operates. For example in UK legislation there are defined notions of Partnership, Limited Company etc that can be used as a basis for classification. Alternatively organizations can be classified by the service they provide (e.g. educational, manufacturing, legal service etc).

ORG is neutral with respect to such choices. It is anticipated that profiles will either introduce sub-classes of [org:Organization](#) or define a classification scheme for organizations. To support the latter the ontology supplies a property [org:classification](#) which can be used to classify an organization using a SKOS [[SKOS-REFERENCE](#)] concept scheme.

Which of these mechanisms to use depends on the situation. If the classification is not intrinsic to the organization but simply some way to group organizations, for example as part of a directory, then [org:classification](#) should be used. If the classification is a reflection of the intrinsic nature of the organization and affects other properties then the sub-class approach should be used. For example, only charities have charity numbers so it would be better to represent a charity as a sub-class of [org:FormalOrganization](#) rather than via a taxonomic labelling.

2.2 Membership and Reporting structure

This section is non-normative.

ORG provides a number of ways to represent the relationship between people and organizations, together with the internal reporting structure of an organization. Experience with early versions of the ontology demonstrated that there is no "one size that fits all". In some cases a very simple direct representation is preferred for ease of consumption. In other cases a more complex representation is needed to capture the nuances of the situation. An ORG profile may specify that a particular subset of these mechanisms be used.

Direct membership relation

This simplest representation provided by ORG is to directly state that some individual (represented as a [foaf:Agent](#)) is [org:memberOf](#) an organization. To represent specific roles that the person plays, ORG profiles may define sub-properties of [org:memberOf](#). In particular, the notion of a leader or head of an organization is so common that ORG provides a built in property specialization of [org:memberOf](#), namely [org:headOf](#) for this purpose.

For example:

```
<http://example.com/people#ceo>
  org:headOf <http://example.com/org#id>.
```

Membership n-ary relationship

However, in general it is advantageous to have an explicit representation of the organizational role that the person fulfils (e.g. for publication of responsibilities associated with the role). This is supported by the [org:Role](#) class. The situation of an Agent fulfilling that role within an organization is then expressed through instances of the [org:Membership](#) n-ary relationship. This also makes it possible to annotate the relationship with qualifying information such as duration, salary, reference to the employment contract and so forth.

For example:

```
<http://example.com/org#id> a org:FormalOrganization;
  skos:prefLabel "Example Ltd".

eg:ctoRole a org:Role;
  skos:prefLabel "CTO".

[] a org:Membership;
  org:member <http://example.com/people#jo>;
  org:organization <http://example.com/org#id>;
  org:role eg:ctoRole;
  org:memberDuring [a time:Interval; time:hasBeginning [
    time:inXSDDateTime "2009-11-01T09:00:00Z"^^xsd:dateTime]].
```

The relationship between this full n-ary relationship and the direct [org:memberOf](#) property can be expressed as an entailment rule, using SPARQL Construct [[RDF-SPARQL-QUERY](#)]:

```
CONSTRUCT {
  ?agent org:memberOf ?org.
} WHERE {
  [] a org:Membership;
  org:member ?agent;
  org:organization ?org.
}
```

Since this representation can be a little less convenient to query and explore via linked data browsing tools the core allows both explicit roles and simple direct relations to be used simultaneously. The relationship between the Role resource and the corresponding property can be indicated through the [org:roleProperty](#) annotation. Thus we might extend the above example with:

```
eg:ctoRole a org:Role;
  org:roleProperty eg:ctoOf.

eg:ctoOf a owl:ObjectProperty, rdf:Property;
```

```

skos:prefLabel "CTO";
rdfs:subPropertyOf org:memberOf.

<http://example.com/people#jo>
  eg:ctoOf <http://example.com/org#id>.

```

The semantics of [org:roleProperty](#) can be expressed using a second closure rule:

```

CONSTRUCT {
  ?agent ?roleprop ?org.
} WHERE {
  [] a org:Membership;
  org:member ?agent;
  org:organization ?org;
  org:role [ org:roleProperty ?roleprop ].
}

```

Tool chains may generate [org:Membership](#) instances and then apply this closure rule to add any corresponding short-cut specializations of [org:memberOf](#).

Posts

This section is non-normative.

The third representation that is provided by ORG is that of a [org:Post](#) which represents some position in the organization that may or may not be currently filled. Posts enable reporting structures and organization charts to be represented independently of the individuals holding those posts. Posts can report to other Posts.

So a [org:Post](#) can exist without someone holding that post. In contrast, a [org:Membership](#) represents the relationship between a particular individual (Agent) and the organization and does not exist unless there is an Agent to partake of the relationship.

While commonly a Post would be held by a single person there are situations in government organizations where a Post may itself be, or be treated as, an Organization. There are no disjointness constraints precluding an application of ORG from treating an entity as both a [org:Post](#) and an [org:Organization](#) simultaneously, if that is an appropriate modelling of the situation.

A post can have an associated [org:Role](#).

Relationship between Posts and Memberships

In many situations only one of Post or Membership is needed, and ORG profiles may specify that use of one of the two is preferred. In cases where the structure of the organization is to be given, independently of the people within that structure, then [org:Post](#) is the appropriate representation to choose. In cases where the aim is to record the people who make up the organization and those memberships are likely to be annotated (e.g. with duration of the membership) then [org:Membership](#) is appropriate.

We can state a formal relationship between these representations in the form of two entailment rules:

```

CONSTRUCT {
  ?agent org:memberOf ?org.
} WHERE {
  ?agent org:holds ?post.
  ?post org:postIn ?org.
}

```

```

CONSTRUCT {
  [] a org:Membership;
  org:member ?agent;
  org:organization ?org;
  org:role ?role.
} WHERE {
  ?agent org:holds ?post.
  ?post org:postIn ?org.
  ?post org:role ?role.
}

```

2.3 Location information

This section is non-normative.

ORG provides [org:Site](#) to represent locations at which organizations exist. The relations [org:siteOf](#) and [org:hasSite](#) establish links between a [org:Site](#) and an organization. We distinguish a primary site ([org:hasPrimarySite](#)) to indicate the default means by which an organization can be contacted, and a registered site ([org:hasRegisteredSite](#)) to indicate a legally registered site for the organization.

The ontology provides [org:siteAddress](#) to define the address of a site using a vocabulary such as the vCard [[vcard-rdf](#)] vocabulary.

2.4 Organizational history

This section is non-normative.

Any aspect of organizational structure is subject to change over time. For the most part this should be handled by an external mechanism such as named graphs. When Organizations change substantially (not simply a change of personnel or internal structure), for example a merger to create a new organization, then the new Organization will typically be denoted by a new URI. In that case we need some vocabulary to describe that change over time and the relationship between the original and resulting resources. ORG provides [org:ChangeEvent](#) and associated properties as a foundation for this, building upon the PROV-O Provenance Vocabulary [[prov-o](#)].

For example to indicate that an organization now called "Department for Education" was formed as a result of rebranding and restructuring and organization called "Department for Children Schools and Family" we might state:

```

<http://example.com/DfE> a org:Organization;
  skos:prefLabel "Department for Education"@en .

```

```
<http://example.com/DCSF> a org:Organization;
  skos:prefLabel "Department for Children Schools and Family"@en .

<http://example.com/regorgMay2010> a org:ChangeEvent;
  rdfs:comment "Post-election re-organization and rebranding"@en ;
  org:originalOrganization <http://example.com/DfE> ;
  org:resultingOrganization <http://example.com/DCSF> .
```

An application can use terms from the PROV-O vocabulary to further describe the change event, for example the period of time over which it occurred. Such usage of PROV-O terms should take into account the semantic constraints [prov-constraints] of the PROV model.

It is sometimes convenient to be able to directly link from an organization to a previous organization from which it descended. This is supported by using the [prov:wasDerivedFrom](#) relationship. ORG declares the property chain axiom:

```
SubObjectPropertyOf( ObjectPropertyChain( org:resultedFrom org:originalOrganization ) prov:wasDerivedFrom )
```

Which can also be expressed using a SPARQL CONSTRUCT

```
CONSTRUCT {
  ?orgR prov:wasDerivedFrom ?orgO .
} WHERE {
  ?orgR org:resultedFrom / org:originalOrganization ?orgO .
}
```

Thus in our previous example, given that [org:resultedFrom](#) and [org:resultingOrganization](#) are inverse of each other, we can deduce:

```
<http://example.com/DfE> prov:wasDerivedFrom <http://example.com/DCSF> .
```

Note

In earlier versions of this ontology the [OPMV Provenance Vocabulary](#) was used. We believe that the PROV-O terms used here are equivalent to the corresponding OPMV terms previously used and that this change does not affect the semantics of the ontology.

2.5 Notes on modelling style

This section is non-normative.

Use of inverses: designers differ on whether providing pairs of inverse relationships between concepts is good practice compared to declaring each relationship in just one direction. In this design we provide inverses for most relations (omitting attribute-like relations). This makes it easier to query the data in linked data settings where a (non-symmetric) closed bounded description is often the default description of each resource. This does incur a cost in terms of maintenance of those relationships. Particular applications of the ontology may adopt a profile in which only certain directions are asserted in the data and leave it up to clients to apply any inverseOf reasoning they require.

Naming: some designers prefer to name properties by nouns which describe the object of the property, others prefer to treat property names as names of the link and use a pattern to indicate the direction of the link. Here we adopt the latter approach for those properties which are relational and especially when the direction is ambiguous. We use the URI pattern `org:hasFoo/org:fooOf` for this but simplify the labels to "foo" and "foo of" to improve readability in linked data viewers.

3. Conformance

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **MAY**, and **OPTIONAL** in this specification are to be interpreted as described in [RFC2119].

A data interchange, however that interchange occurs, is conformant with ORG if:

- it uses terms (classes and properties) from ORG in a way consistent with their semantics as declared in this specification;
- it does **not** use terms from other vocabularies **instead** of ones defined in this vocabulary that could reasonably be used (use of such terms **in addition** to ORG terms is permissible).

A conforming data interchange:

- **MAY** include terms from other vocabularies;
- **MAY** use only a subset of ORG terms.

An **ORG profile** is a specification for data interchange that adds additional constraints to ORG. Additional constraints in a profile **MAY** include (but are not limited to):

- a minimum set of required terms;
- classes and properties for additional terms not covered in ORG;
- controlled vocabularies or controlled sets of URIs to use as acceptable values for properties;
- guidance on use of pairs of inverse properties (such as selecting only one member of the pair to be included, or requiring that both members be explicitly included);
- guidance on choice of modelling approach for roles (see [Membership and Reporting structure](#)).

4. Namespaces

The namespace for ORG is <http://www.w3.org/ns/org#>. However, it should be noted that ORG extends and uses terms from other vocabularies. A full set of alphabetically ordered namespaces and prefixes used in this document is shown in the table below.

Prefix	Namespace	Reference
foaf	http://xmlns.com/foaf/0.1/	[FOAF]
gr	http://purl.org/goodrelations/v1#	[good-relations]
prov	http://www.w3.org/ns/prov#	[prov-o]
org	http://www.w3.org/ns/org#	
owl	http://www.w3.org/2002/07/owl#	[OWL2-PRIMER]
time	http://www.w3.org/2006/time#	[OWL-TIME]
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	[RDF-CONCEPTS]
rdfs	http://www.w3.org/2000/01/rdf-schema#	[RDF-SCHEMA]
skos	http://www.w3.org/2004/02/skos/core#	[SKOS-REFERENCE]
vcard	http://www.w3.org/2006/vcard/ns#	[vcard-rdf]
dct	http://purl.org/dc/terms/	[DC11]

5. Ontology Reference

5.1 Index of classes and properties

Classes: | [ChangeEvent](#) | [FormalOrganization](#) | [Membership](#) | [OrganizationalCollaboration](#) | [OrganizationalUnit](#) | [Organization](#) | [Post](#) | [Role](#) | [Site](#) |

Properties: | [basedAt](#) | [changedBy](#) | [classification](#) | [hasMember](#) | [hasMembership](#) | [hasPost](#) | [hasPrimarySite](#) | [hasRegisteredSite](#) | [hasSite](#) | [hasSubOrganization](#) | [hasUnit](#) | [headOf](#) | [heldBy](#) | [holds](#) | [identifier](#) | [linkedTo](#) | [location](#) | [memberDuring](#) | [memberOf](#) | [member](#) | [organization](#) | [originalOrganization](#) | [postIn](#) | [purpose](#) | [remuneration](#) | [reportsTo](#) | [resultedFrom](#) | [resultingOrganization](#) | [role](#) | [roleProperty](#) | [siteAddress](#) | [siteOf](#) | [subOrganizationOf](#) | [transitiveSubOrganizationOf](#) | [unitOf](#) |

5.2 Organizational structure

5.2.1 Class: Organization

Represents a collection of people organized together into a community or other social, commercial or political structure. The group has some common purpose or reason for existence which goes beyond the set of people belonging to it and can act as an Agent. Organizations are often decomposable into hierarchical structures.

RDFS Class:	org:Organization
subClassOf:	foaf:Agent
equivalentClass:	foaf:Organization
Usage note:	It is recommended that SKOS lexical labels should be used to label the Organization. In particular skos:prefLabel for the primary (e.g. legally recognized name), skos:altLabel for alternative names (trading names, colloquial names) and skos:notation to denote codes from a code list. Alternative names: <i>Collective</i> , <i>Body</i> , <i>Group</i> .

Property: subOrganizationOf

Represents hierarchical containment of Organizations or OrganizationalUnits; indicates an Organization which contains this Organization.

RDF Property:	org:subOrganizationOf
Domain and Range:	org:Organization
Usage note:	Inverse of org:hasSubOrganization .

Property: transitiveSubOrganizationOf

Represents hierarchical containment of Organizations or OrganizationalUnits; indicates an Organization which contains this Organization, directly or indirectly.

RDF Property:	org:transitiveSubOrganizationOf
----------------------	---

RDF Property:	org:transitiveSubOrganizationOf
Domain and Range:	org:Organization
Transitive super property of:	org:subOrganizationOf
Usage note:	The transitive closure of subOrganizationOf, giving a representation of all organizations that contain this one. Note that technically this is a super property of the transitive closure so it could contain additional assertions but such usage is discouraged.

Property: hasSubOrganization

Represents hierarchical containment of Organizations or OrganizationalUnits; indicates an organization which is a sub-part or child of this organization.

RDF Property:	org:hasSubOrganization
Domain and Range:	org:Organization
Usage note:	Inverse of org:subOrganizationOf .

Property: purpose

Indicates the purpose of this Organization. There can be many purposes at different levels of abstraction but the nature of an organization is to have a reason for existence and this property is a means to document that reason. An Organization may have multiple purposes.

RDF Property:	org:purpose
Domain:	org:Organization
Usage note:	It is recommended that the purpose be denoted by a controlled term or code list, ideally a skos:Concept . However, the range is left open to allow for other types of descriptive schemes. It is expected that profiles of this vocabulary will constrain the range of org:purpose . Alternative names: <i>remit</i> , <i>responsibility</i> (esp. if applied to OrganizationalUnits such as Government Departments).

Property: classification

Indicates a classification for this Organization within some classification scheme.

Note that it also permissible for applications to define sub-classes of [org:Organization](#) as a means to represent organizational categories.

RDF Property:	org:classification
Domain:	org:Organization
Range:	skos:Concept
Usage note:	Extension vocabularies may wish to specialize this property to have a range corresponding to a specific skos:ConceptScheme

Property: identifier

Gives an identifier, such as a company registration number, that can be used to uniquely identify the organization.

RDF Property:	org:identifier
Domain:	org:Organization
subPropertyOf:	skos:notation
Usage note:	Many different national and international identifier schemes are available from other vocabularies. The ORG ontology is neutral to which schemes are used. The particular identifier scheme should be indicated by the datatype of the identifier value. Using datatypes to distinguish the notation scheme used is consistent with recommended best practice for skos:notation of which this property is a specialization.

Property: *linkedTo*

Indicates an arbitrary relationship between two organizations.

RDF Property:	org:linkedTo
Domain and Range:	org:Organization
Usage note:	Specializations of this can be used to, for example, denote funding or supply chain relationships.

5.2.2 Class: FormalOrganization

An Organization which is recognized in the world at large, in particular in legal jurisdictions, with associated rights and responsibilities. Examples include a corporation, charity, government or church.

RDFS Class:	org:FormalOrganization
subClassOf:	org:Organization
Usage note:	Note that this is a super class of gr:BusinessEntity and it is recommended to use the GoodRelations vocabulary to denote Business classifications such as DUNS or NAICS.

5.2.3 Class: OrganizationalUnit

An Organization such as a department or support unit which is part of some larger Organization and only has full recognition within the context of that Organization. In particular the unit would not be regarded as a legal entity in its own right.

RDFS Class:	org:OrganizationalUnit
subClassOf:	org:Organization
Usage note:	Units can be large and complex containing other Units. Alternative names: <i>Department</i>

Property: *hasUnit*

Indicates a unit which is part of this Organization, e.g. a Department within a larger Organization.

RDF Property:	org:hasUnit
Domain:	org:Organization
Range:	org:OrganizationalUnit
subPropertyOf:	org:hasSubOrganization
Usage note:	Inverse of org:unitOf .

Property: *unitOf*

Indicates an Organization of which this Unit is a part, e.g. a Department within a larger Organization.

RDF Property:	org:unitOf
Domain:	org:OrganizationalUnit
Range:	org:Organization
subPropertyOf:	org:subOrganizationOf
Usage note:	This is the inverse of org:hasUnit .

5.3 Membership, roles, posts and reporting

5.3.1 Property: memberOf

Indicates that an agent (person or other organization) is a member of the Organization with no indication of the nature of that membership or the role played. Note that the choice of property name is not meant to limit the property to only formal membership arrangements, it is also intended to cover related concepts such as affiliation or other involvement in the organization. Extensions can specialize this relationship to indicate particular roles within the organization or more nuanced relationships to the organization.

RDF Property:	org:memberOf
Domain:	foaf:Agent
Range:	org:Organization
inverseOf:	org:hasMember

5.3.2 Property: hasMember

Indicates an agent (person or other organization) who is a member of the subject Organization. Inverse of [org:memberOf](#), see that property for further clarification.

RDF Property:	org:hasMember
Domain:	org:Organization
Range:	foaf:Agent
Equivalent property:	foaf:member
inverseOf:	org:memberOf
Usage note:	Provided for compatibility with foaf:member .

5.3.3 Property: headOf

Indicates that a person (or other agent) is the leader or formal head of the Organization.

RDF Property:	org:headOf
Domain:	foaf:Agent
Range:	org:Organization
subPropertyOf:	org:memberOf

5.3.4 Class: Membership

Indicates the nature of an Agent's membership of an organization.

RDFS Class:	org:Membership
Usage note:	Represents an n-ary relationship between an Agent, an Organization and a Role. It is possible to directly indicate membership, independent of the specific Role, through use of the org:memberOf property.

Property: member

Indicates the Person (or other Agent including Organization) involved in the Membership relationship.

RDF Property:	org:member
Domain:	org:Membership
Range:	foaf:Agent
Type:	owl:FunctionalProperty
Usage note:	Inverse of org:hasMembership

Property: organization

Indicates the organization in which the Agent is a member.

RDF Property:	org:organization
Domain:	org:Membership
Range:	org:Organization
Type:	owl:FunctionalProperty

Property: role

Indicates the Role that the Agent plays in a Membership relationship with an Organization. Can also be used on a [org:Post](#) to indicate the role that any holder of the Post plays.

RDF Property:	org:role
Domain:	<code>owl:unionOf(org:Membership org:Post)</code>
Range:	org:Role

Property: hasMembership

Indicates a membership relationship that the Agent plays.

RDF Property:	org:hasMembership
Domain:	foaf:Agent
Range:	org:Membership
Usage note:	Inverse of org:member .

Property: memberDuring

Optional property to indicate the interval for which the membership is/was valid.

RDF Property:	org:memberDuring
Domain:	org:Membership
Range:	Formally the interval representation is left open, however as an informative note the use of time:Interval from [OWL-TIME] is suggested.

5.3.5 Class: Role

Denotes a role that a Person or other Agent can take in an organization. Instances of this class describe the abstract role; to denote a specific instance of a person playing that role in a specific organization use an instance of [org:Membership](#).

RDFS Class:	org:Role
subClassOf:	skos:Concept
Usage note:	It is common for roles to be arranged in some taxonomic structure and we use SKOS to represent that. The normal SKOS lexical properties should be used when labelling the Role. Additional descriptive properties for the Role, such as a Salary band, may be added by extension vocabularies.

Property: roleProperty

This is a metalevel property which is used to annotate a [org:Role](#) instance with a sub-property of [org:memberOf](#) that can be used to directly indicate the role for ease of query.

RDF Property:	org:roleProperty
Domain:	org:Role
Range:	rdf:Property
Usage note:	The intended semantics is that a Membership relation involving the Role implies the existence of a direct property relationship through an inference rule of the form: { [] org:member ?a; org:organization ?o; org:role [org:roleProperty ?r] } -> {?a ?r ?o}

Property: remuneration

Indicates a salary or other reward associated with the role.

RDF Property:	org:remuneration
Domain:	org:Role
Usage note:	Typically this will be denoted using an existing representation scheme such as gr:PriceSpecification but the range is left open to allow applications to specialize it (e.g. to remunerationInGBP).

5.3.6 Class: Post

A Post represents some position within an organization that exists independently of the person or persons filling it. Posts may be used to represent situations where a person is a member of an organization ex officio (for example the Secretary of State for Scotland is part of UK Cabinet by virtue of being Secretary of State for Scotland, not as an individual person). A post can be held by multiple people and hence can be treated as a organization in its own right.

RDFS Class:	org:Post
--------------------	--------------------------

Property: holds

Indicates a Post held by some Agent.

RDF Property:	org:holds
Domain:	foaf:Agent
Range:	org:Post
subPropertyOf:	org:memberOf
Usage note:	Inverse of org:heldBy .

Property: heldBy

Indicates an Agent which holds a Post.

RDF Property:	org:heldBy
Domain:	org:Post
Range:	foaf:Agent
subPropertyOf:	org:hasMember
Usage note:	Inverse of org:holds .

Property: postIn

Indicates the Organization in which the Post exists.

RDF Property:	org:postIn
----------------------	----------------------------

RDF Property:	org:postIn
Domain:	org:Post
Range:	org:Organization
Usage note:	Inverse of org:hasPost .

Property: hasPost

Indicates a Post which exists within the Organization.

RDF Property:	org:hasPost
Domain:	org:Organization
Range:	org:Post
Usage note:	Inverse of org:postIn .

5.3.7 Property: reportsTo

Indicates a reporting relationship as might be depicted on an organizational chart. It can be used to indicate a reporting relationship directly between Agents or between Posts that Agents could hold.

RDF Property:	org:reportsTo
Domain:	owl:unionOf(foaf:Agent org:Post)
Range:	owl:unionOf(foaf:Agent org:Post)
Usage note:	The precise semantics of the reporting relationship will vary by organization but is intended to encompass both direct supervisory relationships (e.g. carrying objective and salary setting authority) and more general reporting or accountability relationships (e.g. so called <i>dotted line</i> reporting).

5.4 Location

5.4.1 Class: Site

An office or other premise at which the organization is located. Many organizations are spread across multiple sites and many sites will host multiple locations.

RDFS Class:	org:Site
Usage note:	In most cases a Site will be a physical location. However, we don't exclude the possibility of non-physical sites such as a virtual office with an associated post box and phone reception service. Extensions may provide sub-classes to denote particular types of site.

Property: siteAddress

Indicates an address for the site in a suitable encoding. Use of a well known address encoding such as the vCard [[vcard-rdf](#)] vocabulary is encouraged but the range is left open to allow other encodings to be used. The address may include email, telephone, and geo-location information and is not restricted to a physical address.

RDF Property:	org:siteAddress
Domain:	org:Site

Property: hasSite

Indicates a site at which the Organization has some presence even if only indirect (e.g. virtual office or a professional service which is acting as the registered address for a company).

RDF Property:	org:hasSite
----------------------	-----------------------------

RDF Property:	org:hasSite
Domain:	org:Organization
Range:	org:Site
inverseOf:	org:siteOf

Property: siteOf

Indicates an Organization which has some presence at the given site.

RDF Property:	org:siteOf
Domain:	org:Site
Range:	org:Organization
inverseOf:	org:hasSite

Property: hasPrimarySite

Indicates a primary site for the Organization, this is the default means by which an Organization can be contacted and is not necessarily the formal headquarters.

RDF Property:	org:hasPrimarySite
Domain:	org:Organization
Range:	org:Site
subPropertyOf:	org:hasSite

Property: hasRegisteredSite

Indicates the legally registered site for the organization, in many legal jurisdictions there is a requirement that FormalOrganizations such as Companies or Charities have such a primary designed site.

RDF Property:	org:hasRegisteredSite
Domain:	org:Organization
Range:	org:Site
subPropertyOf:	org:hasPrimarySite

Property: basedAt

Indicates the site at which a person is based. We do not restrict the possibility that a person is based at multiple sites.

RDF Property:	org:basedAt
Domain:	foaf:Person
Range:	org:Site

5.4.2 Property: location

Gives a location description for a person within the organization, for example a *Mail Stop* for internal posting purposes.

RDF Property:	org:location
----------------------	------------------------------

RDF Property:	org:location
Domain:	foaf:Person
Range:	xsd:string

5.5 Projects and other activities

5.5.1 Class: OrganizationalCollaboration

A collaboration between two or more Organizations such as a project. It meets the criteria for being an Organization in that it has an identity and defining purpose independent of its particular members but is neither a formally recognized legal entity nor a sub-unit within some larger organization. Might typically have a shorter lifetime than the Organizations within it, but not necessarily.

RDFS Class:	org:OrganizationalCollaboration
subClassOf:	org:Organization
Usage note:	All members are org:Organization s rather than individuals and those Organizations can play particular roles within the venture. Alternative names: <i>Project, Venture, Endeavour, Consortium</i>

5.6 Historical information

5.6.1 Class: ChangeEvent

Represents an event which resulted in a major change to an organization such as a merger or complete restructuring. It is intended for situations where the resulting organization is sufficiently distinct from the original organizations that it has a distinct identity and distinct URI.

RDFS Class:	org:ChangeEvent
subClassOf:	prov:Activity
Usage note:	Extension vocabularies should define sub-classes of this to denote particular categories of event. The time period over which the event occurred should be expressed using prov:startedAtTime and prov:endedAtTime . A textual description of the event may be given by dct:description .

Property: originalOrganization

Indicates one or more organizations that existed before the change event. Depending on the event they may or may not have continued to exist after the event.

RDF Property:	org:originalOrganization
Domain:	org:ChangeEvent
Range:	org:Organization
inverseOf:	org:changedBy
subpropertyOf:	prov:used

Property: changedBy

Indicates a change event which resulted in a change to this organization.

RDF Property:	org:changedBy
Domain:	org:Organization
Range:	org:ChangeEvent
inverseOf:	org:originalOrganization
Usage note:	Depending on the event the organization may or may not have continued to exist after the event.

Property: *resultedFrom*

Indicates an event which resulted in (led to, generated) this organization.

RDF Property:	org:resultedFrom
Domain:	org:Organization
Range:	org:ChangeEvent
subpropertyOf:	prov:wasGeneratedBy
inverseOf:	org:resultingOrganization

Property: *resultingOrganization*

Indicates an organization which was created or changed as a result of the event.

RDF Property:	org:resultingOrganization
Domain:	org:ChangeEvent
Range:	org:Organization
inverseOf:	org:resultedFrom

Property chain axiom

In addition the ontology defines the following relationship between [org:resultedFrom](#), [org:originalOrganization](#) and [prov:wasDerivedFrom](#) :

```
SubObjectPropertyOf( ObjectPropertyChain( org:resultedFrom org:originalOrganization ) prov:wasDerivedFrom )
```

A. Change history

Changes since [the W3C Proposed Recommendation 17 December 2013](#): None.

Changes since [W3C Candidate Recommendation 25 June 2013](#):

- The range of [org:memberDuring](#) has been made unconstrained. Use of [time:Interval](#) is still suggested but is not a normative requirement and so the reference to [OWL-TIME] is now informative.
- The reference to foaf has been made normative to reflect its usage in the definitions of several ORG terms.
- Clarified textual description of [org:memberOf](#) and [org:hasMember](#)
- Changed reference to Turtle specification to be non-normative, it is only used in non-normative examples.
- Updated the diagram in section 1 to a clearer, more complete version, thanks to João Paulo Almeida.
- Added namespace and links to vocabulary file to the Abstract for ease of reference.
- Duplicated the index of vocabulary terms earlier in the document for ease of reference.
- Removed CR specific text on implementation feedback and At Risk features.

Changes since [W3C Working Draft 23 October 2012 \(Last Call\)](#):

- Added explicit declarations that [org:member](#) and [org:organization](#) are functional properties. This is a clarification rather than an intended change of semantics.
- Removed assertion that [org:Post](#) is a sub class of [org:Organization](#), adding an informative note that ORG applications are still free to declare entities as being instances of both classes.
- Added [property chain axiom](#) for [prov:wasDerivedFrom](#).
- Removed the range constraint on [org:siteAddress](#) to allow other encodings than VCard to be used.
- Added a statement that [org:Organization](#) is equivalent to the foaf:Organization class. This statement was present in the ontology itself at the time of last call but not sufficiently clear in this document.
- Removed informative comment that the org:reportsTo graph is acyclic, this is not necessarily the case.
- Changed document title to use title case and definite article.
- Updated the references to PROV-O and Turtle.
- Marked the terms in Section 5.6 Historical Information as At Risk.
- Marked the normative use of OWL Time as At Risk.

Changes since [W3C Working Draft 05 April 2012](#):

- Modified [org:ChangeEvent](#), [org:originalOrganization](#) and [org:resultedFrom](#) to extend the PROV-O vocabulary instead of OPMV.
- Added [org:Post](#) and associated properties.
- Corrected description of [org:hasSubOrganization](#) and [org:subOrganizationOf](#).
- Corrected domain of [org:remuneration](#) to match ontology.
- Corrected domain/range of [org:hasMember](#) to match ontology.
- Added missing entry for [org:memberOf](#) to match ontology.
- Rephrasing and restructuring throughout the document to improve clarity.

Changes since previous version 0.3 [2010-06-09](#):

- Added [org:identifier](#) to support generic use of organization identifier schemes, including local and national schemes.

Changes since previous version 0.2 [2010-06-07](#):

- Added [org:changedBy](#) and [org:resultingOrganization](#) as inverses to the OPMV compatible properties and added a note on use of inverses.

Changes since previous version 0.1 [2010-05-28](#):

- Rephrased the introduction to clarify scoping of the ontology.
- Made [org:Organization](#) equivalent to [foaf:Organization](#) following clarifying discussions with Dan Brickley.
- Added [org:hasMember](#) as an inverse of [org:memberOf](#) in order to be able to declare the equivalent to [foaf:member](#), thanks to Dan Brickley for the suggestion.
- Made [org:unitOf/org:hasUnit](#) sub properties of [org:subOrganizationOf/org:hasSubOrganization](#), thanks to Dave Challis for prompting this clarification.
- Added [org:transitiveSubOrganizationOf](#), thanks to Damian Steer for the suggestion.
- Fixed range statements for [org:role](#) and [org:organization](#) to correspond to the documentation and intent, thanks to Bernard Vatan for spotting that problem.
- Modified description of [org:memberOf](#) to clarify that the notion of membership is very broad and not meant to be limited to formal notions of membership.
- Clarified the description of [org:FormalOrganization](#).
- Fixed bug in restriction definition for [org:OrganizationalCollaboration](#), thanks to Start Williams for point out the problem.
- Reversed [org:resultingOrganization](#) to [org:resultedFrom](#) for compatibility with OPMV, thanks to Jeni Tension for pointing out the problem.
- Changed labels for [has*](#) properties to read as nouns, leaving URIs unchanged.
- Added [rdfs:isDefinedBy](#) declarations, thanks to Kingsley Idehen for pointing out the lack of those.
- Clarified license terms, thanks to Dan Brickley for pointing out the lack.

B. Acknowledgements

This ontology was originally developed for use by data.gov.uk as part of the UK Linked Data Kernel project under the leadership and guidance of John Sheridan (The National Archives). Jeni Tension provided immensely useful feedback and suggestions on the first draft, which greatly improved its design. The work also took inspiration from a number of other ontologies, particularly the thoughtfully designed [Gazette Organization ontology](#) and [Proton-top](#).

The editor gratefully acknowledges comments and support from the Government Linked Data Working Group - particularly to Daniel Vila for transcribing the original documentation into [W3C](#) format and to Bart van Leeuwen for feedback and discussions that resulted in the addition of [org:Post](#).

The modelling of [org:Post](#) is closely based on the work of Jeni Tension on modelling UK Government.

C. References

C.1 Normative references

[FOAF]

Dan Brickley, Libby Miller. [FOAF Vocabulary Specification 0.98](#). 9 August 2010. URL: <http://xmlns.com/foaf/spec/>

[OWL2-PRIMER]

Pascal Hitzler; Markus Krötzsch; Bijan Parsia; Peter Patel-Schneider; Sebastian Rudolph. [OWL 2 Web Ontology Language Primer \(Second Edition\)](#). 11 December 2012. W3C Recommendation. URL: <http://www.w3.org/TR/owl2-primer/>

[RDF-CONCEPTS]

Graham Klyne; Jeremy Carroll. [Resource Description Framework \(RDF\): Concepts and Abstract Syntax](#). 10 February 2004. W3C Recommendation. URL: <http://www.w3.org/TR/rdf-concepts/>

[RDF-SCHEMA]

Dan Brickley; Ramanathan Guha. [RDF Schema 1.1](#). 9 January 2014. W3C Proposed Edited Recommendation. URL: <http://www.w3.org/TR/rdf-schema/>

[RDF-SPARQL-QUERY]

Eric Prud'hommeaux; Andy Seaborne. [SPARQL Query Language for RDF](#). 15 January 2008. W3C Recommendation. URL: <http://www.w3.org/TR/rdf-sparql-query/>

[RFC2119]

S. Bradner. [Key words for use in RFCs to Indicate Requirement Levels](#). March 1997. Internet RFC 2119. URL: <http://www.ietf.org/rfc/rfc2119.txt>

[SKOS-REFERENCE]

Alistair Miles; Sean Bechhofer. [SKOS Simple Knowledge Organization System Reference](#). 18 August 2009. W3C Recommendation. URL: <http://www.w3.org/TR/skos-reference>

[prov-o]

Timothy Lebo; Satya Sahoo; Deborah McGuinness. [PROV-O: The PROV Ontology](#). 30 April 2013. W3C Recommendation. URL: <http://www.w3.org/TR/prov-o/>

C.2 Informative references

[DC11]

Dublin Core metadata initiative. [Dublin Core metadata element set, version 1.1](#). July 1999. Dublin Core recommendation. URL: <http://dublincore.org/documents/dcmi-terms/>

[OWL-TIME]

Jerry Hobbs; Feng Pan. [Time Ontology in OWL](#). 27 September 2006. W3C Working Draft. URL: <http://www.w3.org/TR/owl-time>

[good-relations]

Martin Hepp, [GoodRelations Language Reference](#), Version 1.0, 1 October 2011, URL: <http://www.heppnetz.de/ontologies/goodrelations/v1.html>

[prov-constraints]

James Cheney; Paolo Missier; Luc Moreau. [Constraints of the PROV Data Model](#). 30 April 2013. W3C Recommendation. URL: <http://www.w3.org/TR/prov-constraints/>

[turtle]

Eric Prud'hommeaux; Gavin Carothers. [RDF 1.1 Turtle](#). 9 January 2014. W3C Proposed Recommendation. URL: <http://www.w3.org/TR/turtle/>

[vcard-rdf]

Renato Iannella; James McKinney. [vCard Ontology](#). 24 September 2013. W3C Working Draft. URL: <http://www.w3.org/TR/vcard-rdf/>