

Tagging Legal Text

Editor: [John McClure](#)
Hypergrove Engineering
February 28, 2006

1. Introduction

Most legal jurisdictions have now published their statutes and other official documents on the Web, usually serving them as HyperText Markup Language (HTML4) web-pages or as Adobe PDF image files. Though immediately useful as a textual reference, these formats do not represent the legal content of the documents in a manner relevant to software agents operating on the Internet.

This document provides some guidelines for converting HTML/PDF documents into "semantic web resources" that may be harvested by Internet software agents delivering an entirely new class of Internet application to a user's browser. The World Wide Web Consortium (W3C) has published several technical recommendations that form the foundation for construction of a "Semantic Web".

The "Semantic Web" can be conceptualized as a boundless networked database on the Internet that exists either beside or atop conventional HTML webpages. Creating this database requires either that [Resource Description Framework](#) (RDF) *files* be retrievable from one's web-site or, equivalently, that RDF *keywords* be [embedded](#) within one's HTML files [\[1\]](#). The equivalence between these methods arises from the fact that the 'keywords' embedded within an HTML file *are the same* as those used within an RDF file.

The W3C has established standards for definitions of keywords, but has wisely refrained from defining keyword vocabularies, believing that these vocabularies (called 'ontologies') should be produced by individual domains and their subject-matter experts. In this regard, the ontologies published by [Legal-RDF](#) are applied to the requirements of tagging the content of legal documents.

2. Functional Requirements

2.1 Information Model

Legal documents -- indeed, all documents -- may through its text content contain the following types of statements.

2.1.1 Existential Statements

A document can 'declare' things that are said to exist by virtue of its declaration that they exist. Their existence have their foundation in reality or not, that is, with regard to whether the document is considered "non-fiction", "fiction", "historical fiction", or "science fiction". Further, a document can declare specific things that are said to have existed in the past or will exist in the future. A document can similarly define specific events that are occurring, did occur, or will occur.

Examples: a particular 'country' (per a constitution or other chartering document); a certain 'committee' (per a statute, minutes, announcement, etc.); a certain 'person' (per a birth certificate, novel, articles of incorporation, etc.); a 'contract' (per a statute, contractual agreement, legal opinion, etc.); a 'position' such as President (per a constitution, articles of incorporation, etc.); a 'hearing' (per a transcript); an 'examination' (per a test); and a 'topic' (per a legal code, an encyclopedia, etc).

These examples correspond to the top-level divisions of the Legal-RDF ontologies (Actor, Role, Scene, Prop, Theme, Drama, and Script), but not Quality, or Quantity.

2.1.2 Definitional Statements

A document can define types of things said to exist, that did exist, or that will exist, by virtue of its declaration that they do, will, or did exist. Similarly a document can define classes of events that are said to have occurred, are occurring, or will occur.

Examples: a 'Washington State citizen' (per statute); a 'party' such as a 'seller' (per a contract-agreement); a 'relief request' having minimum incidence requirements (per a regulation); an 'industrial building' identifying required uses or capabilities (per a rule); a 'tax-exempt event'

In addition to an optional effective period, definitions of classes of things or of events always identify the minimum

requirements that a thing or event must (not) have or be before it may be said to be representative of the class. These minimum requirements may either be about the non/existence of associated attributes, about the value(s) of its associated attributes, or about the truth (or falseness) of its ability to be representative of one or more other classes of thing.

2.1.3 Factual Statements

A document can declare information said to be true (or that its falseness is true) about a specific thing or class of thing. These statements associate attributive information with a thing or class of thing, or event or class of event. A factual statement also may relate actors, roles, scenes, props, dramas, themes, and scripts to one-another. The attributing information can be properties that the actor, role, scene, prop, drama, theme, or script currently has, once had in the past, or will have in the future. The attribute may also be related in the negative sense, that is, as an attribute that the thing (or class of things) does not, did not, or will not, have as a property.

Examples: any name, description, identifier, date, time, interval of time, numeric quantity, image, or other text information that is (not) associated with an actor, role, scene, prop, theme, or drama.

2.1.4 Conditional Statements [2]

A document can declare conditionally-true (or false) statements. These statements may concern the non/existence of specific things; the non/existence of instances of classes of things; the non/truth of factual statements; or the non/truth of other conditional statements.

A conditional statement is modelled as

If <Condition-A> holds then <Party-A> is {obliged, permitted, prohibited} to/by <Party-B> to perform <Action-X> {before|until} <Condition-B>.

As can be seen, three types of conditional statements need to be accommodated by markup within a document.

- **Obligations** -- requires Party-A to perform Action-X.
- **Permissions** -- allows Party-A to perform Action-X.
- **Prohibitions** -- denies Party-A the permission to perform Action-X.

The above use of 'Party-A' and 'Party-B' can be generalized to any type of actor, role, scene, prop, theme, drama, or script resource. The notion above to 'perform Action-X' can be generalized to include negative assertions; assertions about the existence of a resource; assertions about the classes of which a resource is representative; and assertions about attributing properties associated with a resource.

The above use of 'Condition-A' and 'Condition-B' can be generalized to a date-relevant dependency, that is, a starting-date whose value depends on the non/occurrence of Condition-A, and an ending-date whose value depends on the non/occurrence of Condition-B. These two dates may be qualified relative to whether the occurrence of the Condition needs to be initiated or partially or fully completed in order to establish the starting- or ending-date after or before which the obligation, permission, or prohibition is effective.

2.1.5 Other Statement Types

Comparative statements need to be discussed. Choice/menu statements may be a special case of conditional.

2.2 Operational Requirements

2.2.1 Scoped Definitions

Definitions of legal concepts (definitional statements) are established by legal documents and legal instruments, not by an ontology. For instance, an ontology can define a "Citizen" as a class, however no ontology can validly define for example, a "Washington State Citizen" -- these concepts are defined by their relevant statutes; all statutes are rooted in documents published on the Semantic Web.

Construction of a rationalized Semantic Web strongly suggests that such definitions best share a common class when possible, e.g., <http://www.legalrdf.org/Actor/Citizen>, which at least establishes the minimum information requirements to be met by jurisdictional or otherwise scoped definitions.

2.2.2 Superseded Definitions

A mechanism is needed to overcome superseded definitions. Since legal documents, or content within legal documents, may be superseded by subsequent documents, external references to superseded legal content (its classes and resource definitions) may not be valid.

2.2.3 XHTML Representations

In addition to markup within an RDF files, tagging guidelines must support markup within XHTML files in a manner

conforming to (currently, prospective) standards issued by the W3C.

3. Tagging Guidelines

3.1 Introduction

The Legal-RDF ontologies refine the typical architecture of Resource Description Framework ontologies. RDF statements are modelled as subjects and predicates; typical ontologies define predicates that combine a verb and direct-object into a single term. Legal-RDF however constrains its predicates to be possessive and existential verbs, thereby yielding a tagging system that is more semantically powerful, more intuitively natural, and more consistent with conventional data modelling practice.

Conventional RDF would, for instance, associate a name with a location as:

```
<Location rdf:ID='&the;EiffelTower'>
  <hasName xml:lang='EN'>Eiffel Tower</hasName>
</Location>
```

Legal-RDF separates the conventional predicate's possessive verb from its noun part as:

```
<Location rdf:ID='&the;EiffelTower'>
  <has>
    <LocationName eng='Eiffel Tower' />
  </has>
</Location>
```

There are many benefits and few costs associated with this refinement of typical RDF ontology architecture. Perhaps the greatest cost occurs in the extension of the relational concept of a "triple" as embedded within RDF databases, to include dimensions of time and type. The 'type' dimension refers to whether the predicate verb is possessive or existential; the 'time' dimension relates whether the relationship exists in the past, present, or future. These dimensions may be further qualified, for instance deontologically, that is, as a prohibition, permission, or obligation.

Table 1. Legal-RDF Predicates

	Factual Predicates	Conditional Predicates		
		Obligation	Permission	Prohibition
Past Existential	wasA wasNotA	mustHaveBeenA	mayHaveBeenA	mustNotHaveBeenA
Present Existential	isA isNotA	mustBeA	mayBeA	mustNotBeA
Future Existential	willBeA willNotBeA			
Past Possessive	had hadNot	mustHaveHad	mayHaveHad	mustNotHaveHad
Present Possessive	has hasNot	mustHave	mayHave	mustNotHave
Future Possessive	willHave willNotHave			

For instance, to identify that a person, a former citizen of the State of New York, is now a citizen of the State of Washington who is to become a citizen of the State of California, would be:

```
<!Entity cacd "http://www.legalrdf.org/US/CA/Government/Code/">
<!Entity nygd "http://www.legalrdf.org/US/NY/Government/Code/">
<!Entity wacd "http://www.legalrdf.org/US/WA/Government/Code/">

<Person rdf:ID='&the;Person' asOf='2006-02-28'>
  <isA rdf:resource='&wacd;Citizen' />
  <wasA rdf:resource='&nygd;Citizen' />
  <willBeA rdf:resource='&cacd;Citizen' />
</Person>
```

In certain situations, it may be more convenient or appropriate to state an existential predicate as a possessive attribute.^[3]

```
<Person rdf:ID='&the;Person'>
  <has>
    <CitizenType rdf:about='&wacd;Citizen' />
  </has>
</Person>
```

3.1.1 Basic XHTML Tagging

XHTML files are tagged in-line to the XHTML elements, inserting ``, `<meta>`, and `<link>` elements as needed. The W3C has two specifications for embedding RDF keywords within XHTML documents, [\[XHTML2\]](#) and [\[RDFA\]](#). These specifications are the foundation for the "dotted name" technique shown here; mapping between these approaches is straight-forward and discussed briefly below.

The 'dotted name' property identifiers simply concatenate keywords together.

```
<dl about='&the;Person'>
  <dt>Current Address:</dt>
  <dd property='legalRDF:PersonalAddress'>123 Main Street</dd>
  <dt>Previous Address:</dt>
  <dd property='legalRDF:had.PersonalAddress.eng'>345 Water Street</dd>

  <dt>Citizenship:</dt>
  <dd>State of Washington <link rel="legalRDF:isA" href="&wacd;Citizen" /></dd>
</dl>
```

Note that the "Current Address" tagging presumed both the applicable language and (for current possessives only) helper verb. If these were not presumed, then its proper tagging would be:

```
<dl about='&the;Person'>
  <dt>Current Address:</dt>
  <dd property='legalRDF:has.PersonalAddress.eng'>123 Main Street</dd>
</dl>
```

Technically, the "dotted name" notation can be easily translated into the format expected by conventional RDF/A processors. This translation is fully transparent to an end-user who performs the tagging.

```
<dl about='&the;Person'>
  <dt>Current Address:</dt>
  <dd>
    <meta id='idAddress' />
    <link about='#idAddress' rel='legalRDF:isA' href='&legalRDF;PersonalAddress' />
    <span about='#idAddress' property="eng">123 Main Street</span>
    <link about='&the;Person' rel="legalRDF:has" href="#idAddress" />
  </dd>
</dl>
```

3.2 Class Retrievals

The definition of a `&wacd;Citizen` when retrieved by a Semantic Web agent could indicate that it is a subclass of the U.S.C. "Citizen" class (itself a subclass of the Legal-RDF "Citizen" class), and that it is defined by (say) the 10th statement within the 14th chapter of the 2nd title within the legal code of the State of Washington effective July, 2005. Finally, it may also indicate that it had previous definitions.

```
<!Entity wacd "http://www.legalrdf.org/US/WA/Government/Code/">
<!Entity usc "http://www.legalrdf.org/US/Code/">
<!Entity legalRDF "http://www.legalrdf.org/">

<Class rdf:ID='&wacd;Citizen'>
  <has>
    <SuperClass rdf:about='&usc;Citizen' />
    <StatutoryCitation rdf:about='&wacd;2005/2.14/S10' asOf='2005-07-01' />
  </has>
  <had>
    <StatutoryCitation rdf:about='&wacd;2004/2.14/S07' asOf='2004-07-01' />
    <StatutoryCitation rdf:about='&wacd;2003/2.14/S06' asOf='2003-07-01' />
    <StatutoryCitation rdf:about='&wacd;2002/2.14/S06' asOf='2002-07-01' />
    <StatutoryCitation rdf:about='&wacd;2001/2.14/S06' asOf='2001-07-01' />
  </had>
</Class>
```

Rather than requiring the agent to access the resource &wacd;2005/2.14/S10 separately, the class definition may include salient characteristics of the class' definition. In this example, a citizen of the State of Washington is a Person who is a PermanentResident of the State of Washington.

```
<Class rdf:ID='&wacd;Citizen'>
  <has>
    <SuperClass rdf:about='&usc;Citizen' />
    <StatutoryCitation rdf:about='&wacd;2005/2.14/S10' asOf='2005-07-01'>
      <has>
        <StatementSubject rdf:about='&legalRDF;Person'>
          <isA rdf:resource='&wacd;PermanentResident' />
        </StatementSubject>
      </has>
    </StatutoryCitation>
  </has>
</Class>
```

Suppose that a PermanentResident is defined in the legal code as one who has a local address; who is an owner of a local residence; or who is a renter of a local residence.

```
<Class rdf:ID='&wacd;PermanentResident'>
  <has>
    <SuperClass rdf:about='&usc;Resident' />
    <StatutoryCitation rdf:about='&wacd;2005/2.11/S005' asOf='2005-07-01'>
      <has>
        <rdf:Alt>
          <rdf:li>
            <StatementSubject rdf:about='&legalRDF;Person'>
              <has>
                <wacd:LocalAddress />
              </has>
            </StatementSubject>
          </rdf:li>
          <rdf:li>
            <StatementSubject rdf:about='&legalRDF;Person'>
              <isA>
                <wacd:ResidentialRenter />
              </isA>
            </StatementSubject>
          </rdf:li>
          <rdf:li>
            <StatementSubject rdf:about='&legalRDF;Person'>
              <isA>
                <wacd:ResidentialOwner />
              </isA>
            </StatementSubject>
          </rdf:li>
        </rdf:Alt>
      </has>
    </StatutoryCitation>
  </has>
</Class>
```

Now the definition of a "local address" is given in which the address' locale is constrained to be the State of Washington.

```
<Class rdf:ID='&wacd;LocalAddress'>
  <has>
    <SuperClass rdf:about='&legalRDF;Address' />
    <StatutoryCitation rdf:about='&wacd;2005/2.11/S006' asOf='2005-07-01'>
      <has>
        <StatementSubject rdf:about=''>
          <has>
            <AddressLocale rdf:about='&legalRDF;StateOfWashington' />
          </has>
        </StatementSubject>
      </has>
    </StatutoryCitation>
  </has>
</Class>
```

Now the types of residents are defined with respect to whether the address that they own or are renting, is a local State of Washington address.

```

<Class rdf:ID='&wacd;ResidentialRenter'>
  <has>
    <SuperClass rdf:about='&legalRDF;Renter' />
    <StatutoryCitation rdf:about='&wacd;2005/2.11/S006' asOf='2005-07-01'>
      <has>
        <StatementSubject rdf:about='Person'>
          <has>
            <RenterAddress>
              <isA rdf:resource='&wacd;LocalAddress' />
            </RenterAddress>
          </has>
        </StatementSubject>
      </has>
    </StatutoryCitation>
  </has>
</Class>
<Class rdf:ID='&wacd;ResidentialOwner'>
  <has>
    <SuperClass rdf:about='&legalRDF;Owner' />
    <StatutoryCitation rdf:about='&wacd;2005/2.11/S006' asOf='2005-07-01'>
      <has>
        <StatementSubject rdf:about='Person'>
          <has>
            <OwnerAddress>
              <isA rdf:resource='&wacd;LocalAddress' />
            </OwnerAddress>
          </has>
        </StatementSubject>
      </has>
    </StatutoryCitation>
  </has>
</Class>

```

3.3 Class Applications

A key capability of Legal-RDF is the ease by which one can determine what information is missing or inappropriate in a resource's record that would validate or invalidate a claim to its membership in a class.

For instance, the following code would raise a signal because though an individual is said to be a Citizen of the State Of Washington, there is no evidence in the record that this is a true assertion; there needs to be either a local address directly associated with the individual; or a statement that the individual is a renter at, or an owner of, a local address.

```

<Person rdf:ID='&the;Person'>
  <isA rdf:resource='&wacd;Citizen' />
</Person>

```

The Legal-RDF ontologies are architected such that the credibility of assertions can be tested. For instance in the below example, because &the;Person is a &wacd;PermanentResident then, given the definition of a &wacd;Citizen, one can conclude that &the;Person is a resident of the State of Washington. Similarly, one can validate &the;Person's citizenship from the statement that &the;Person is a &wacd;ResidentialOwner or, similarly, that &the;Person is a &wacd;ResidentialRenter.

One of the powerful features of Legal-RDF is the ability to conclude that if &the;Person has an OwnerAddress attribute -- which implies that the Person is an Owner -- and that that address is a &wacd;LocalAddress, then it can be concluded that the individual is a &wacd;ResidentialOwner and, therefore, is a &wacd;Citizen. [Note: the sample below shows an alternative and equally valid expression of the types associated with a resource.]

```

<Person rdf:ID='&the;Person'>
  <isA>
    <wacd;Citizen />
    <wacd;PermanentResident />
    <wacd;ResidentialOwner />
    <wacd;ResidentialRenter />
  </isA>
  <has>
    <wacd;LocalAddress rdf:resource='someAddress' />
    <RenterAddress rdf:resource='someAddress' />
    <OwnerAddress rdf:resource='someAddress' />
  </has>
</Person>

```

3.4 Tagging Existential Statements

Existential statements use the six existential predicates (isA, isNotA, etc.) to declare the type of a resource. If the rdf:type predicate is used, it is to be interpreted as a present-tense existential.

```
<!Entity cacd "http://www.legalrdf.org/US/CA/Government/Code/">
<!Entity nycd "http://www.legalrdf.org/US/NY/Government/Code/">
<!Entity wacd "http://www.legalrdf.org/US/WA/Government/Code/">

<Person rdf:ID='&the;Woman' asOf='2006-02-28'>
  <rdf:type rdf:resource='&wacd;Citizen' />
  <isA rdf:resource='&wacd;Citizen' />
  <isNotA rdf:resource='&nycd;Citizen' />
  <wasA rdf:resource='&nycd;Citizen' />
  <wasNotA rdf:resource='&caycd;Citizen' />
  <willBeA rdf:resource='&cacd;Citizen' />
  <willNotBeA rdf:resource='&nycd;Citizen' />
</Person>
```

In XHTML, existential statements are tagged using a XHTML <link> element that references the relevant Legal-RDF existential predicate.

```
<p about='&the;Woman'>
  <meta property='legalRDF:asOf' content='2006-02-28' />
  <link rel='legalRDF:isA' href='&legalRDF;Person' />

  She is now a citizen of the State of Washington,
  <link rel='legalRDF:isA' href='&wacd;Citizen' />
  not a citizen of New York as commonly believed.
  <link rel='legalRDF:isNotA' href='&nycd;Citizen' />
  While it's true she <em>was</em> a citizen of New York,
  <link rel='legalRDF:wasA' href='&nycd;Citizen' />
  it's inaccurate that she's ever lived in California.
  <link rel='legalRDF:wasNotA' href='&cacd;Citizen' />
  She tells me that she is moving to California though,
  <link rel='legalRDF:willBeA' href='&cacd;Citizen' />
  <link rel='legalRDF:willNotBeA' href='&wacd;Citizen' />
  and that she's unlikely ever to return to New York.
  <link rel='legalRDF:willNotBeA' href='&nycd;Citizen' />
</p>
```

3.4.1 Existential Collisions

The statement above that &the;Woman is a Person is technically redundant whenever a Semantic Web agent determines that one of its Citizen classes is ultimately a subclass of the &legalRDF;Person class.

However, if there is only one negative assertion and no positive assertion, then the class hierarchy applicable to the pertinent class cannot drive subsequent algorithmic conclusions. For example, if &the;Woman is not a New York citizen, without also indicating where she is a citizen, then it **may not** be concluded that not only is she not a New York citizen, but also that she is not a &usc;Citizen, nor a &legalRDF;Citizen, nor a &legalRDF;Person. Rather, queries in this situation about whether &the;Woman isA &legalRDF;Person (or &legalRDF;Citizen or &usc;Citizen) can only resolve to "no information".

Whenever a positive existential assertion is made, then its superclass hierarchy always applies existentially. For instance, if &the;Woman is a Washington State citizen, then due to the inheritance applicable to &wacd;Citizen, &the;Woman is equally a &usc;Citizen, &legalRDF;Citizen, and &legalRDF;Person.

Collisions can occur whenever a negative existential assertion is made in the presence of a countervailing positive existential assertion with which it shares a common class inheritance. These collisions are always resolved in favor of the hierarchy applicable to the positive existential assertion. For instance, though a collision occurs for &the;Woman who is a Washington State citizen but not a New York citizen (at the point of the &usc;Citizen class), &the;Woman nevertheless remains an instance of the &usc;Citizen, &legalRDF;Citizen, and &legalRDF;Person classes.

3.4.2 Existential Identifiers

Resources are declared in the Resource Description Framework by the rdf:ID attribute; the value of this attribute is a URI that may be queried using the mime-type "application/rdf+xml". The RDF provides the rdf:about attribute -- whose value is a URI for a resource -- for statements that are 'about' that resource. The RDF itself does not use the xml:id attribute in any way.

In an XHTML context, the xml:id attribute is used to identify a block of text or some other element. The emerging

standard for embedding RDF within XHTML [RDFA] currently allows an xml:id to be a URI [4]. At the moment the W3C is uncertain about whether an xml:id can be interpreted as an rdf:ID, that is, whether the value of an xml:id can be the value of an 'about' attribute; the apparent conflict in W3C standards stems from distinguishing between "XHTML resources" and "information resources" and involves the mime-type used to access that resource.

The Legal-RDF approach in the XHTML context is to always use the 'about' attribute, with the exception of statements generated during the conversion of dotted-names to conventional RDF/A expressions. This restriction may be relaxed as the W3C resolves the issues.

3.5 Tagging Factual Statements

Factual statements use the six possessive predicates (has, hasNot, etc.) to declare properties possessed by a resource. If no predicate is used to link an attributing property to a resource (said to be an "anonymous" relationship) then the property is to be interpreted as one possessed by the resource in the present-tense, that is, as if the <has> predicate had been used.

Text strings can be located in either of two places: formally, as the value of one of the Legal-RDF text attributes (e.g., "eng") or informally, as the value of a Legal-RDF direct-object XML element (e.g., "PersonalAddress" and "AddressLocale"). In the RDF, a text attribute can be represented either as an XML attribute, or as a child XML element. For those instances when text is provided as content of a Legal-RDF direct-object XML element, the text is interpreted as the *title* of the resource; Legal-RDF ontologies have been structured to identify those resource properties that are to contain the title of the resource.

```
<Person rdf:ID='&the;Person' asOf='2006-02-28'>

  <!-- following is assumed to be within a <has> element -->
  <PersonalAddress eng='123 Main Street' />

  <had>
    <PersonalAddress><eng>345 Water Street</eng></PersonalAddress>
  </had>
  <willHave>
    <PersonalAddress>789 Water Street</PersonalAddress>
  </willHave>
  <hasNot>
    <PersonalAddress>
      <has>
        <AddressLocale rdf:about='&legalRDF;StateOfNewYork' />
      </has>
    </PersonalAddress>
  </hasNot>
</Person>
```

In XHTML, factual statements are tagged using the XHTML "property" attribute. (In the example below, it is unnecessary to identify the &the;Person is-A Person, because the PersonalAddress property only may be associated with instances of the Person class.)

```
<dl about='&the;Person'>
  <dt>Current Address:</dt>
  <dd property='legalRDF:PersonalAddress'>123 Main Street</dd>
  <dt>Previous Address:</dt>
  <dd property='legalRDF:had.PersonalAddress.eng'>345 Water Street</dd>
</dl>
```

Legal-RDF structures its text attributes into a hierarchy consistent with text parsing algorithms and international standards for languages, currencies, units of measure, and time. The root of the hierarchy, the "text" property, is preceded by whether the string is language-specific, a numeric quantity, or a date or time expression. Numeric quantity sub-types correspond to the SI Units of Measure and to national currencies, as one would expect. All text attributes by convention are in lower-case.

```
<Person rdf:ID='&the;Person' asOf='2006-02-28'>
  <has>
    <GivenName text='Jimmy' />
    <FamilyName eng='Jones' />
    <PersonalHeight in='68' cm='172.7' />
    <AnnualIncome yr='2006' text='$ 50,000' />
  </has>
  <had>
    <AnnualIncome yr='2005' usd='47000' />
  </had>
</Person>
```

A direct-object element can have more than one text attribute. Elements therefore have the capacity to simultaneously carry multiple translations for a single value, e.g., "eng" and "spa" for English and Spanish strings. For numeric values, multiple

units of measurement can be simultaneously carried by a single element, e.g., "in" and "cm" for Inch and Centimeter values. For financial values, multiple currencies can be accommodated, e.g., "usd" and "eur" for United States Dollar and Euro-dollar values.

In the XHTML context, representing multiple values with a dotted-name requires particular consideration. For instance above, the AnnualIncome element has both "yr" and "usd" attributes. The special keyword "Context" on a sibling <meta> element is adequate to communicate additional properties of the trailing 'direct-object' in the sibling property's dotted-name. In the example, the "Context" references the AnnualIncome element created when the dotted-name is expanded (as described above).

```
<dl about='&the;Person'>
  <dt>First Name:</dt><dd property='legalRDF:has.GivenName.text'>Jimmy</dd>
  <dt>Last Name:</dt> <dd property='legalRDF:has.FamilyName.eng'>Jones</dd>
  <dt>Height (inches):</dt>
  <dd>
    <span property='legalRDF:has.PersonalHeight.in'>68</span>
    <meta property='legalRDF:Context.cm' content='172.7' />
  </dd>
  <dt>Annual Income:</dt>
  <dd>
    <span property='legalRDF:has.AnnualIncome.text'>$ 50,000</span>
    <meta property='legalRDF:Context.yr' content='2006' />
  </dd>
  <dt>Last Year's Income:</dt>
  <dd>
    <span property='legalRDF:had.AnnualIncome.usd'>47000</span>
    <meta property='legalRDF:Context.yr' content='2005' />
  </dd>
</dl>
```

3.5.1 Flowing Text Properties

In the XHTML context, a single text property can be split across multiple XHTML elements. Should an application want to give the impression of pagination for instance, a text paragraph could flow across a page boundary onto the next page. In the Legal-RDF document model, a document and a page have paragraphs while a paragraph exists on one or more pages.

In the example below, the current document is referenced as an object by using the RDF/A convention of a null 'about' attribute. Two other mechanisms are also demonstrated. First, text fragments are indicated by a numeric suffix on the name of the text attribute ("eng"). Second, the "Context" keyword is used to relate direct-object properties to the last direct-object in the contextual dotted-name property.

```
<div class='classPage' about='&the;FirstPage'>
  <link about='' rel='legalRDF:has.DocumentPage' href='&the;FirstPage' />
  <p about='' property='legalRDF:has.DocumentParagraph.eng.1'>
    <link property='legalRDF:Context.has.ContentPage' href='&the;FirstPage' />
    Here is text that does not finish until after a page boundary.</p>
</div>
<div class='classPage' about='&the;SecondPage'>
  <link about='' rel='legalRDF:has.DocumentPage' href='&the;SecondPage' />
  <p about='' property='legalRDF:has.DocumentParagraph.eng.2'>
    <link property='legalRDF:Context.has.ContentPage' href='&the;SecondPage' />
    Here is the remainder of the text for the paragraph.</p>
</div>
```

This annotation can be transformed to the following RDF/A representation.

```
<div class='classPage' about='&the;FirstPage'>
  <link about='&the;FirstPage' rel='legalRDF:isA' href='&legalRDF;DocumentPage' />
  <link about='' property='legalRDF:has' href='&the;FirstPage' />
  <p>
    <meta id='idPara' />
    <link about='#idPara' rel='legalRDF:isA' href='&legalRDF;DocumentParagraph' />
    <meta id='idPage1' />
    <link about='#idPage1' rel='legalRDF:isA' href='&legalRDF;ContentPage' />
    <link about='#idPara' property='legalRDF:has' href='#idPage1' />
    <link about='' rel="legalRDF:has" href="#idPara"/>
    <meta id='idPart1' />
    <span about='#idPart1' property="eng">Here is text that does not finish ...</span>
    <link about='#idPart1' rel='legalRDF:isA' href='&legalRDF;ParagraphContent' />
    <link about='#idPara' property='legalRDF:has' href='#idPart1' />
  </p>
</div>
```

```

<div class='classPage' about='&the;SecondPage'>
  <link about='&the;SecondPage' rel='legalRDF:isA' href='&legalRDF;DocumentPage' />
  <link about='' property='legalRDF:has' href='&the;SecondPage' />
  <p about='' property='legalRDF:has.DocumentParagraph.eng.2'>
    <meta id='idPage2' />
    <link about='#idPage2' rel='legalRDF:isA' href='&legalRDF;ContentPage' />
    <link about='#idPara' property='legalRDF:has' href='&the;SecondPage' />
    <meta id='idPart2' />
    <span about='#idPart2' property="eng">Here is the remainder of the paragraph.</span>
    <link about='#idPart2' rel='legalRDF:isA' href='&legalRDF;ParagraphContent' />
    <link about='#idPara' property='legalRDF:has' href='#idPart2' />
  </p>
</div>

```

3.6 Tagging Conditional Statements

A conditional statement is composed of two sub-statements: an Hypothesis (the "if" part) and a Conclusion (the "then" part). Hypotheses identify one or more attributing properties or states of existence for a resource that must be true for the statement's Conclusion(s) to be true. Conclusions can be characterized as statements of fact, or as statements of permission, prohibition, or obligation on the part of a resource to possess a particular set of attributing properties or to be representative of a set of particular classes.

In the first example below, if the-person has a given characteristic or existential state, then the-person may/must/mustNot have some other characteristic or state. The second and third statements show examples that AND and OR hypotheses, respectively. The fourth statement shows a complex set of hypotheses and a complex set of conclusions.

```

<ConditionalStatement rdf:ID='&the;FirstStatement' asOf='2006-02-28'>
  <has>
    <StatementHypothesis rdf:about='&the;Person' />
    <StatementConclusion rdf:about='&the;Person' />
  </has>
</ConditionalStatement>
<ConditionalStatement rdf:ID='&the;SecondStatement' asOf='2006-02-28'>
  <has>
    <rdf:Seq>
      <rdf:li><StatementHypothesis rdf:about='&the;Person' /></rdf:li>
      <rdf:li><StatementHypothesis rdf:about='&the;Person' /></rdf:li>
    </rdf:Seq>
    <StatementConclusion rdf:about='&the;Person' />
  </has>
</ConditionalStatement>
<ConditionalStatement rdf:ID='&the;ThirdStatement' asOf='2006-02-28'>
  <has>
    <rdf:Alt>
      <rdf:li><StatementHypothesis rdf:about='&the;Person' /></rdf:li>
      <rdf:li><StatementHypothesis rdf:about='&the;Person' /></rdf:li>
    </rdf:Alt>
    <StatementConclusion rdf:about='&the;Person' />
  </has>
</ConditionalStatement>
<ConditionalStatement rdf:ID='&the;FourthStatement' asOf='2006-02-28'>
  <has>
    <rdf:Alt>
      <rdf:li>
        <rdf:Seq>
          <rdf:li><StatementHypothesis rdf:about='&the;Person' /></rdf:li>
          <rdf:li><StatementHypothesis rdf:about='&the;Person' /></rdf:li>
        </rdf:Seq>
      </rdf:li>
      <rdf:li><StatementHypothesis rdf:about='&the;Person' /></rdf:li>
    </rdf:Alt>
    <rdf:Alt>
      <rdf:li>
        <rdf:Seq>
          <rdf:li><StatementConclusion rdf:about='&the;Person' /></rdf:li>
          <rdf:li><StatementConclusion rdf:about='&the;Person' /></rdf:li>
        </rdf:Seq>
      </rdf:li>
      <rdf:li><StatementConclusion rdf:about='&the;Person' /></rdf:li>
    </rdf:Alt>
  </has>
</ConditionalStatement>

```

In an XHTML context, tagging uses many techniques already developed. Conversion of this material to RDF/A does require that Legal-RDF's Alt, Seq, and li keywords are to be translated to the equivalent RDF elements.

```
<p about='&the;FirstParagraph'>
  <link about='' rel='legalRDF:has.DocumentParagraph' href='&the;FirstParagraph' />
  <span about='&the;FirstStatement'>
    <span property='legalRDF:eng'>If the Person has an X, then s/he must have a Y.</span>
    <link about='' rel='legalRDF:has.DocumentStatement' href='&the;FirstStatement' />
    <link rel='legalRDF:has.StatementSource' href='&the;FirstParagraph' />
    <span>
      <span property='legalRDF:has.StatementHypothesis.about' href='&the;Person' />
      <link property='legalRDF:Context.has' href='&legalRDF;X' />
    </span>
    <span>
      <span property='legalRDF:has.StatementConclusion.about' href='&the;Person' />
      <link property='legalRDF:Context.mustHave' href='&legalRDF;Y' />
    </span>
  </span>
  <span about='&the;SecondStatement'>
    <span property='legalRDF:eng'>If s/he has or had an X, then s/he must have a Y.</span>
    <link about='' rel='legalRDF:has.DocumentStatement' href='&the;SecondStatement' />
    <link rel='legalRDF:has.StatementSource' href='&the;FirstParagraph' />
    <span about='&the;FirstHypothesis'>
      <link property='legalRDF:isA' href='&legalRDF;StatementHypothesis' />
      <span property='legalRDF:about' href='&the;Person' />
      <link property='legalRDF:has' href='&legalRDF;X' />
    </span>
    <span about='&the;SecondHypothesis'>
      <link property='legalRDF:isA' href='&legalRDF;StatementHypothesis' />
      <span property='legalRDF:about' href='&the;Person' />
      <link property='legalRDF:had' href='&legalRDF;X' />
    </span>
    <span>
      <link property='legalRDF:has.Alt.li' href='&the;FirstHypothesis' />
      <link property='legalRDF:Context.li' href='&the;SecondHypothesis' />
    </span>
    <span>
      <span property='legalRDF:has.StatementConclusion.about' href='&the;Person' />
      <link property='legalRDF:Context.mustHave' href='&legalRDF;Y' />
    </span>
  </span>
</p>
```

3.6.1 Hypotheses

Hypotheses can identify a variety of conditions, from a generic event to a value of a property possessed by a resource.

```
<!-- If a flooding occurs .... -->
<Hypothesis>
  <has>
    <StatementSubject rdf:about='&legalRDF;FloodingEvent' />
  </has>
</Hypothesis>

<!-- If the Person has an annual income that is less than $ 10,000 USD -->
<Hypothesis>
  <has>
    <StatementSubject rdf:about='&the;Person'>
      <has>
        <AnnualIncome>
          <has>
            <MaximumValue usd='9999.99' />
          </has>
        </AnnualIncome>
      </has>
    </StatementSubject>
  </has>
</Hypothesis>
```

3.6.2 Conclusions

Identifies a specific resource that is obligated, permitted, or prohibited from possessing certain properties or being of certain classes. The obliged resource is indicated using the 'rdf:about' attribute. The resource can be an instance of a

Class that is a subclass of the Actor, Role, Scene, Prop, Theme, Drama, or Script classes, or it can be a specific Thing that is an instance of one these classes.

```
<Conclusion>
  <has>
    <StatementSubject rdf:about='&a;Role'>
      <mustHave rdf:resource='&legalRDF;RoleName' />      <!-- obligation -->
      <mustBeA  rdf:resource='&legalRDF;SpeakingRole' />   <!-- obligation -->
      <mayHave  rdf:resource='&legalRDF;Understudy' />     <!-- permission -->
      <mayBeA   rdf:resource='&legalRDF;Boy' />            <!-- permission -->
      <mustNotHave rdf:resource='&legalRDF;Father' />     <!-- prohibition -->
      <mustNotBeA rdf:resource='&legalRDF;Female' />      <!-- prohibition -->
    </StatementSubject>
  </has>
</Conclusion>
```

4. Tagging Document Statements

These guidelines apply to statutory and other law documents; legal briefs and judicial opinions; regulatory documents; policy manuals; rule books; legal agreements, and other documents that contain narrative text.

Legal-RDF's ontology has a basic "Content" class that contains document text. This class has properties for each kind of statement made by text content, that is, definitional, factual, and conditional statements. Factual statements include subtypes assumptions, conclusions, and hypotheses. Conditional statements include obligations, permissions, and prohibitions. Definitional statements include those defining classes and those defining things.

A given stanza of document content can simultaneously have multiple statements of multiple types. The Legal-RDF architecture for properties participating in 1-to-many relationships such as this uses a "linked-list" approach whenever reasonable, thereby allowing explicit specification of the sort-order of members within a set (an architecture that satisfies XML best-practice recommendations). At the same time, the "next" and "previous" member properties share a common superclass which therefore allows unsorted collections.

4.1 Document Assumptions

Legal and regulatory documents routinely make assumptions about facts associated with other resources. The assumptions may be those necessary to resolve signals raised by editing of a contract against, for instance, statutory contract requirements.

In the example below, statutes associated with the jurisdiction in which the contract occurs, requires that every party must be an Adult. However the party "Joey Smith" was not indicated to be an Adult as a fact elsewhere in the document, so the application created an Assumption made by the (other) parties to the contract.

```
<Agreement rdf:about='&the;Agreement'>
  <has>
    <FirstAssumption rdf:about='&the;FirstAssumption'>
      <has>
        <StatementSubject rdf:about='&the;JoeySmith'>
          <isA rdf:resource='&legalRDF;Adult' />
        </StatementSubject>
      </has>
    </FirstAssumption>
    </NextStatement rdf:about='&the;SecondAssumption'>
  </has>
</Agreement>
```

4.2 Document Classes

A common device used in legal documents is to describe the obligations, permissions, and prohibitions of a particular party to the document, and then assign a person or company to the defined party-role. In Legal-RDF a class is defined for the role so that multiple agents can be assigned to the party-role, each with its own defining statement.

```
<Agreement rdf:about='&the;Agreement'>
  <has>
    <a:VendorParty rdf:about='&the;Corporation' />
    <a:VendeeParty rdf:about='&the;Person' />
  </has>
</Agreement>
```

This architecture provides the capacity to define information classes of any type, e.g., a committee and chairperson.

```

<Statute rdf:about='&the;Statute'>
  <has>
    <StatuteClass rdf:ID='&a;MedalOfHonorCommittee'>
      <isA rdf:resource='&legalrdf;Committee' />
      <isA rdf:resource='&legalrdf;SingleInstance' />
      <has>
        <ClassProperty rdf:about='&legalrdf;FormalTitle'>
          <has>
            <PropertyValue eng='Medal Of Honor Committee' />
          </has>
        </ClassProperty>
        <ClassProperty rdf:about='&legalrdf;FormalAcronym'>
          <has>
            <PropertyValue eng='MoHC' />
          </has>
        </ClassProperty>
      </has>
    </StatuteClass>
    <StatuteClass rdf:ID='&a;MoHCChairperson'>
      <isA rdf:resource='&legalrdf;Chairperson' />
    </StatuteClass>
  </has>
</Statute>

```

Instantiations follow of the two MoHC-specific classes defined above.

```

<a:MedalOfHonorCommittee rdf:about='&the;MedalOfHonorCommittee'>
  <has>
    <StatutoryCitation rdf:about='&the;Statute' />
    <FormalTitle eng='Medal Of Honor Committee' />
    <FormalAcronym eng='MoHC' />
    <FirstChair rdf:about='&the;FirstMoHCChairperson' />
    <LatestChair rdf:about='&the;FirstMoHCChairperson' />
    <LatestMeeting date='20060331' rdf:ID='&the;MoHC20060331'>
      <has>
        <MeetingMinutes rdf:about='&the;MoHC20060331.html' />
      </has>
    </LatestMeeting>
    <LatestRule rdf:ID='&the;Rule-01' />
  </has>
</a:MedalOfHonorCommittee>

<a:MoHCChairperson rdf:about='&the;Person' rdf:ID='&the;FirstMoHCChairperson'>
  <has>
    <StatutoryCitation rdf:about='&the;Statute' />
    <FirstChairmanship rdf:about='&the;MedalOfHonorCommittee' />
    <NextChairperson />
    <PreviousChairperson />
  </has>
</a:MoHCChairperson>

```

4.3 Document Conclusions

Documents may contain conclusions that are based upon other facts contained in or referenced by the document. Conclusions can be inserted by reasoning agents during an editing and validation process against the document. In the example below, the agent has determined that "Joey Smith" is an Adult from the fact recounted from elsewhere that he is a parent of a child.

```

<Agreement rdf:about='&the;Agreement'>
  <has>
    <FirstConclusion rdf:ID='&the;FirstConclusion'>
      <has>
        <StatementSubject rdf:about='&the;JoeySmith'>
          <isA rdf:resource='&legalRDF;Adult' />
        </StatementSubject>
        <FirstReason>
          <has>
            <StatementSource rdf:about='&another;Document'>
              <StatementSubject rdf:about='&the;JoeySmith'>
                <isA rdf:resource='&legalRDF;Parent' />
              </StatementSubject>
            </has>
          </FirstReason>
        </has>
      </has>
    </FirstConclusion>
  </has>
</Agreement>

```

```

    </FirstReason>
  </has>
</FirstConclusion>
</NextStatement rdf:about='&the;SecondConclusion'>
  <has>
</Agreement>

```

4.4 Document Facts

Documents make factual statements about resources (classes and things) which are not defined within/by the document. These statements are simply recounting information retrieved from elsewhere, ensuring that subsequent agents processing the document have the same information as the agent did.

```

<Agreement rdf:about='&the;Agreement'>
  <has>
    <FirstFact rdf:ID='&the;FirstFact'>
      <has>
        <StatementSubject rdf:about='&the;JoeySmith' asOf='20060228'>
          <has>
            <GivenName eng='Joey' />
            <FamilyName eng='Smith' />
            <PersonalAddress eng='123 Main Street'>
              <has>
                <AddressLocale rdf:about='&legalRDF;Zip/98368' />
              </has>
            <PersonalAge yr='38' />
            <PersonalGender rdf:about='&legalRDF;Male' />
            <PersonalIncome usd='65000' />
            <PersonalSpouse rdf:about='&the;KatySmith' />
            <PersonalEducation yr='16' />
          </has>
        </StatementSubject>
        <NextStatement rdf:about='&the;SecondFact' />
      </has>
    </FirstFact>
  <has>
</Agreement>

```

4.5 Document Hypotheses

In Legal-RDF a hypothesis is information that is neither true nor false, but is presumed to be true so as to evaluate its consequences. A hypothesis would be created in a document for instance to identify 'what-if' conditions pertinent to any other content within the document.

The example below marks the contract agreement as a hypothetical scenario, that is, the content of the agreement presumes that Joey Smith has a \$95,000 income. Normally, a notarization agent would not allow the document to contain any hypotheses whatsoever.

```

<Agreement rdf:about='&the;Agreement'>
  <has>
    <DocumentHypothesis>
      <has>
        <StatementSubject rdf:about='&the;JoeySmith'>
          <has>
            <PersonalIncome usd='95000' />
          </has>
        </StatementSubject>
      </has>
    </DocumentHypothesis>
  <has>
</Agreement>

```

4.6 Document Obligations

All statements of obligation are conditional statements, that is, each obligation must include both a hypothesis (the 'if' part) and a conclusion (the 'then' part). The conclusion *must* use the <mustHave> and <mustBeA> predicates for the subject of the conclusion, in order that these may be recounted without modification in the context of the resource that is the subject of the conclusion.

In contract scenarios for instance, accepting a contract offer creates the obligation to record receipt of a purchase deposit. In the example below, the agreement document identifies a legal contract which it is associated. This relationship allows multiple agreements to pertain to a single contract; the ContractAgreement property for &the;Contract identifies &the;Agreement as the basis for the contract.

```

<PurchaseAgreement rdf:about='&the;PurchaseAgreement'>
  <has>
    <LegalContract rdf:about='&the;PurchaseContract'>
      <isA rdf:resource='&legalRDF;PurchaseContract' />
      <has>
        <ContractAgreement rdf:about='&the;PurchaseAgreement'>
          </has>
      </LegalContract>
    <DocumentObligation rdf:about='&the;ContractObligation'>
      <has>
        <StatementHypothesis>
          <has>
            <StatementSubject rdf:about='&the;Vendor'>
              <has>
                <VendorContract rdf:about='&the;PurchaseContract'>
                  <has>
                    <ContractAcceptance />
                  </has>
                </VendorContract>
              </has>
            </StatementSubject>
          </has>
        </StatementHypothesis>
        <StatementConclusion>
          <has>
            <StatementSubject rdf:about='&the;Vendor'>
              <mustHave>
                <VendorReceipt rdf:ID='&the;ContractAcceptance'>
                  <has>
                    <ReceivedItem rdf:about='&the;Deposit' />
                    <ReceiptDate />
                    <ReceiptLocation />
                    <ReceiptMethod />
                  </has>
                </VendorReceipt>
              </mustHave>
            </StatementSubject>
          </has>
        </StatementConclusion>
      </has>
    </DocumentObligation>
  </has>
</PurchaseAgreement>

```

4.7 Document Permissions

All statements of permission are conditional statements, that is, each statement must include both a hypothesis (the 'if' part) and a conclusion (the 'then' part). The conclusion *must* use the <mayHave> and <mayBeA> predicates for the subject of the conclusion, in order that these may be recounted without modification in the context of the resource that is the subject of the conclusion.

The example below shows tagging for a renewal of a listing contract that has expired. The renewal agreement permits the broker to hold an open-house for the property a maximum of one Saturday per month; it does not obligate the broker to hold any open-houses for the property.

```

<ListingAgreement rdf:about='&the;ListingRenewal2'>
  <isA rdf:about='&legalRDF;ContractRenewal' />
  <has>
    <LegalContract rdf:about='&the;ListingContract'>
      <isA rdf:resource='&legalRDF;ListingContract' />
      <has>
        <ContractAgreement rdf:about='&the;ListingAgreement'>
          <ContractAgreement rdf:about='&the;ListingRenewal1'>
            <ContractAgreement rdf:about='&the;ListingRenewal2'>
          </has>
        </LegalContract>
      </has>
    </LegalContract>
  </has>

```

```

<DocumentPermission rdf:about='&the;ContractPermission'>
  <has>
    <StatementHypothesis>
      <has>
        <StatementSubject rdf:about='&the;Vendor'>
          <has>
            <VendorContract rdf:about='&the;ListingContract'>
              <has>
                <ContractAcceptance/>
              </has>
            </VendorContract>
          </has>
        </StatementSubject>
      </has>
    </StatementHypothesis>
    <StatementConclusion>
      <has>
        <StatementSubject rdf:about='&the;RealProperty'>
          <mayHave>
            <OpenHouse rdf:ID='&the;OpenHouse'>
              <has>
                <PresentationAgent rdf:about='&the;Broker' />
                <PresentationItem rdf:about='&the;RealProperty' />
                <PresentationDate>
                  <has>
                    <DateDay rdf:about='&legalRDF;Saturday' />
                    <MonthlyFrequency count='1' />
                  </has>
                <PresentationDate>
                  </has>
              </OpenHouse>
            </mayHave>
          </StatementSubject>
        </has>
      </StatementConclusion>
    </has>
  </DocumentPermission>
</has>
</PurchaseAgreement>

```

4.8 Document Prohibitions

All statements of prohibition are conditional statements, that is, each statement must include both a hypothesis (the 'if' part) and a conclusion (the 'then' part). The conclusion *must* use the <mayNotHave> and <mayNotBeA> predicates for the subject of the conclusion, in order that these may be recounted without modification in the context of the resource that is the subject of the conclusion.

This type of statement follows the same pattern as described for Document Obligations and Document Permissions.

4.9 Document Things

Legal documents often define specific things which otherwise do not exist in the absence of the document. For instance, a statute might define a standing committee of government.

```

<Statute rdf:ID='&the;Statute'>
  <has>
    <StatuteContent>
      <eng>The "Medal of Honor Committee" is hereby established.</eng>
    </StatuteContent>
    <StatuteThing rdf:ID='&the;MedalOfHonorCommittee'>
      <isA rdf:resource='&a;MedalOfHonorCommittee' />
      <has>
        <FormalTitle eng='Medal Of Honor Committee' />
        <FormalAcronym eng='MoHC' />
      </has>
    </StatuteThing>
  </has>
</Statute>

```

5. Notes and Bibliography

^[1] Technically, text 'entities' are retrieved, not text 'files'.

^[2] More generally, deontic statements are discussed here which relate to obligations had by a party. The general principles apply however to conditional statements of all kinds.

^[3] 'CitizenType' is for illustrative purposes; it is not a property defined by Legal-RDF.

^[4] Technically, the xml:id is the fragment part of a full URI formed from the URI for the XHTML entity, e.g., http://www.dot.com/someFile.html#xml_id, an approach consistent with the normal treatment of xml:id in the XHTML context.

[RDFA] [RDF/A Primer 1.0](#), produced by the [RDF-in-HTML Task Force](#) of the [Semantic Web Best Practices and Deployment Working Group](#).

[XHTML2] [XHTML, Version 2](#), produced by the [HTML Working Group](#).
