Putting FrameNet Data into the ISO Linguistic Annotation Framework

Srinivas Narayanan **Miriam R. L. Petruck** **Collin F. Baker** **Charles J. Fillmore**
{snarayan, miriamp, collinb, fillmore}@icsi.berkeley.edu
International Computer Science Institute
1947 Center St., Berkeley, California

1 Abstract

This paper describes FrameNet (Lowe et al., 1997; Baker et al., 1998; Fillmore et al., 2002), an online lexical resource for English based on the principles of frame semantics (Fillmore, 1977a; Fillmore, 1982; Fillmore and Atkins, 1992), and considers the FrameNet database in reference to the proposed ISO model for linguistic annotation of language resources (ISO TC37 SC4)(ISO, 2002; Ide and Romary, 2001b). We provide a data category specification for frame semantics and FrameNet annotations in an RDF-based language. More specifically, we provide a DAML+OIL markup for **lexical units**, defined as a relation between a lemma and a semantic frame, and frame-to-frame relations, namely **Inheritance** and **Subframes**. The paper includes simple examples of FrameNet annotated sentences in an XML/RDF format that references the project-specific data category specification.

2 Frame Semantics and the FrameNet Project

FrameNet's goal is to provide, for a significant portion of the vocabulary of contemporary English, a body of semantically and syntactically annotated sentences from which reliable information can be reported on the valences or combinatorial possibilities of each item included.

A semantic frame is a script-like structure of inferences, which are linked to the meanings of linguistic units (lexical items). Each frame identifies a set of frame elements (FEs), which are frame-specific semantic roles (participants, props, phases of a state of affairs). Our description of each lexical item identifies the frames which underlie a given meaning and the ways in which the FEs are realized in structures headed by the word. The FrameNet database documents the range of semantic and syntactic combinatory possibilities (valences) of each word in each of its senses, through manual annotation of example sentences and automatic summarization of the resulting annotations. FrameNet I focused on governors, meaning that for the most part, annotation was done in respect to verbs; in FrameNet II, we have been annotating in respect to governed words as well.[1] This paper will explain the theory behind FrameNet, briefly discuss the annotation process, and then describe how the FrameNet data can be represented in RDF, using DAML+OIL, so that researchers on the semantic web can use the data.

2.0.1 Frame Semantic Background

In Frame Semantics (Fillmore, 1976; Fillmore, 1977b; Fillmore and Atkins, 1992; Petruck, 1996), a linguistic unit, in our case, a word (in just one of its senses), **evokes** a particular frame. An "evoked" frame is the structure of knowledge required for the understanding of a given lexical or phrasal item. The frames in question can be simple – small static scenes or states of affairs, simple patterns of contrast, relations between entities and the roles they serve – or possibly quite complex event types that provide the background for words that profile one or more of their phases or participants.

For example, the word *bartender* evokes a scene of service in a setting where alcoholic beverages are consumed, and profiles the person whose role is to prepare and serve these beverages. In a sentence like *The bartender asked for my ID*, it is the individual who occupies that role that we understand as making the request, and the request for identification is understood against the set of assumptions and practices of that frame.

[1]The National Science Foundation has provided funding for FrameNet through two grants, IRI #9618838 "Tools for Lexicon Building" (1997-2000, PI Charles Fillmore, Co-PI Dan Jurafsky) and ITS/HCI #0086132 "FrameNet++: An On-Line Lexical Semantic Resource and its Application to Speech and Language Technology" (PI Charles Fillmore, Co-PIs Dan Jurafsky, Srini Narayanan, and Mark Gawron). We refer to the two phases of the project as FrameNet I and FrameNet II.

2.0.2 Replacement: An Example Frame

A schematic description of the REPLACEMENT frame will include an AGENT effecting a change in the relationship between a PLACE (which can be a role, a function, a location, a job, a status, etc.) and a THEME. For example, in the sentence *Sal replaced his cap on his bald head*, *Sal* fills the role of AGENT, *his cap* instantiates the FE THEME, and *on his bald head* is the PLACE. The words defined in terms of this frame include *exchange.v, interchange.v, replace.v, replacement.n, substitute.v, substitution.n, succeed.v, supplant.v, swap.v, switch.v*, and *trade.v*.

The REPLACEMENT frame involves states of affairs and transitions between them such that other situations are covered: an "old theme", which we refer to as OLD, starts out at the PLACE and ends up not at the PLACE, while a "new theme", which we call NEW, starts out not at the PLACE and ends up at the PLACE (as in *Factory owners replaced workers by machines*).

Syntactically, the role of AGENT can be expressed by a simple NP (e.g. *Margot switched her gaze to the floor*, a conjoined NP (e.g. *Margot and her admirer exchanged glances*), or two separate constituents, an NP and a PP (e.g. *Margot exchanged glances with her admirer*). Similarly, PLACE may be expressed as one PP or two. Compare *Ginny switched the phone between hands* and *Ginny switched the phone from one hand to the other*. And, if OLD and NEW are of the same type, they can be expressed as a single FE (e.g. *The photographer switched lenses*).

2.1 The FrameNet Process

Using attested instances of contemporary English, FrameNet documents the manner in which frame elements (for given words in given meanings) are grammatically instantiated in English sentences and organizes and exhibits the results of such findings in a systematic way. For example, in causative uses of the words, an expression about *replacing NP with NP* takes the direct object as the OLD and the oblique object as the NEW (e.g. *Nancy replaced her desktop computer with a laptop*), whereas *substituting NP for NP* does it the other way around (e.g. *Nancy substituted a laptop for her desktop computer*). A commitment to basing such generalizations on attestations from a large corpus, however, has revealed that in both UK and US English, the verb *substitute* also participates in the valence pattern found with *replace*, i.e. we find examples of *substituting* the OLD ***with*** the NEW (e.g. *Nancy subsitituted a laptop with her desktop computer*).

In their daily work, FrameNet staff members record the variety of combinatorial patterns found in the corpus for each word in the FrameNet lexicon, present the results as the **valences** of the words, create software capable of deriving from the annotations as much other information as possible about the words, and add manually only that information which cannot – or cannot easily – be derived automatically from the corpus or from the set of annotated examples.

2.2 Frame-to-Frame Relations

The FrameNet database records information about several different kinds of semantic relations, consisting mostly of frame-to-frame relations which indicate semantic relationships between collections of concepts. The two that we consider here are **Inheritance** and **Subframes**.

2.2.1 Inheritance

Frame **Inheritance** is a relationship by which a single frame can be seen as an elaboration of one or more other parent frames, with bindings between the inherited semantic roles. In such cases, all of the frame elements, subframes, and semantic types of the parent have equal or more specific correspondents in the child frame. Consider for example, the CHANGE_OF_LEADERSHIP frame, which characterizes the appointment of a new leader or removal from office of an old one, and whose FEs include: SELECTOR, the being or entity that brings about the change in leadership (in the case of a democratic process, the electorate); OLD LEADER, the person removed from office; OLD ORDER, the political order that existed before the change; NEW LEADER, the person appointed to office; and ROLE, the position occupied by the new or old leader. Some of the words that belong to this frame describe the successful removal from office of a leader (e.g. *overthrow, oust, depose*), others only the attempt (e.g. *uprising, rebellion*). This frame inherits from the more abstract REPLACEMENT frame described above, with the following FEs further specified in the child: OLD and NEW are narrowed to humans beings or political entities, i.e. OLD_LEADER and NEW_LEADER, respectively; and PLACE is an (abstract) position of political power, i.e. ROLE.

2.2.2 Subframes

The other type of relation between frames which is currently represented in the FN database is between a complex frame and several simpler frames which constitute it. We call this relationship **Subframes**. In such cases, frame elements of the complex frame may be identified (mapped) to the frame elements of the subparts, although not all frame elements of one need have any relation to the other. Also, the ordering and other temporal relationships of the subframes can be specified using binary precedence relations. To illustrate, consider the complex CRIMINAL_PROCESS frame, defined as follows: A Suspect is arrested by an AUTHORITY on certain CHARGES, then is arraigned as a DEFENDANT. If at any time the

DEFENDANT pleads guilty, then the DEFENDANT is sentenced, otherwise the DEFENDANT first goes to trial. If the VERDICT after the trial is guilty, then the DEFENDANT is sentenced. In the end, the DEFENDANT is either released or is given a SENTENCE by a JUDGE at the sentencing. For each step in the process, there is a separate frame in the database, including ARREST, ARRAIGNMENT, TRIAL, SENTENCING, and so on. Each of these frames is related to the CRIMINAL_PROCESS frame via the SubFrame relation in the frame editor. Moreover, subframes (of the same complex frame) are related to each other through their ordering.

We have recognized the need to deal with other types of relations among frames, and, so far, have identified two, **SeeAlso**, and **Using**. Currently, many Using relations are indicated in the FrameNet database.

2.3 The FrameNet Product

The FrameNet database contains descriptions of more than 7,000 lexical units based on more than 130,000 annotated sentences. This information is available for a wide range of natural language processing applications, including question answering, machine translation, and information extraction.

The FN database can be seen both as a dictionary and a thesaurus. As a dictionary, each **lexical unit (LU)** (lemma in a given sense) is provided with (1) the name of its frame, (2) a definition, (3) a valence description which summarizes the attested combinatorial possibilities, and (4) access to annotated examples. The FN database can also be seen as a thesaurus, associating groups of lexical units in frames and associating frames with each other (see below). The FrameNet database differs from existing lexical resources in the specificity of the frames and semantic roles it defines, the information it provides about relations between frames, and the degree of detail provided on the possible syntactic realizations of semantic roles for each LU.

While Ide, et al., (2002)(Ide et al., 2002) offers a representation scheme for dictionaries and other lexical data, the kind of information in the FrameNet database is not expressed in the same level of depth in any existing print dictionary or computational lexical resource. For instance, while WordNet describes semantic relations between words, it does not recognize conceptual schemas, i.e. frames, that mediate in these relations, and therefore does not have the means to link arguments of predicating words with the semantic roles they express. FrameNet also differs from WordNet in showing semantic relations across parts of speech, and in providing contextual information enriched with semantics (beyond the "Someone __s something" format of WordNet argument-structure representations). Thus, the complex relational structure inherent in the FrameNet frame element and frame-to-frame relations exercises and potentially extends the ISO TC37 SC4 standard (ISO, 2002). The rest of this paper describes our encoding of the FrameNet database in an RDF-based environment.

3 A Data Category Specification for Frame Semantics in RDF

The World Wide Web (WWW) contains a large amount of information which is expanding at a rapid rate. Most of that information is currently being represented using the Hypertext Markup Language (HTML), which is designed to allow web developers to display information in a way that is accessible to humans for viewing via web browsers. While HTML allows us to visualize the information on the web, it doesn't provide much capability to describe the information in ways that facilitate the use of software programs to find or interpret it. The World Wide Web Consortium (W3C) has developed the Extensible Markup Language (XML) which allows information to be more accurately described using tags. As an example, the word *crawl* on a web site might represent an *offline search* process (as in web crawling) or an exposition of a type of *animate motion*. The use of XML to provide metadata markup, such as for *crawl*, makes the meaning of the word unambiguous. However, XML has a limited capability to describe the relationships (schemas or ontologies) with respect to objects. The use of ontologies provides a very powerful way to describe objects and their relationships to other objects. The DAML language was developed as an extension to XML and the Resource Description Framework (RDF). The latest release of the language (DAML+OIL) (http://www.daml.org) provides a rich set of constructs with which to create ontologies and to markup information so that it is machine readable and understandable.

Framenet-1 has been translated into DAML+OIL. We developed an automatic translator from FrameNet to DAML+OIL which is being updated to reflect FrameNet2 data. With periodic updates as the FrameNet data increases, we expect it to become useful for various applications on the Semantic Web. DAML+OIL is written in RDF (http://www.w3.org/TR/daml+oil-walkthru/#RDF1), i.e., DAML+OIL markup is a specific kind of RDF markup. RDF, in turn, is written in XML, using XML Namespaces (http://www.w3.org/TR/daml+oil-walkthru/#XMLNS), and URIs. Thus, our framenet declaration begins with an RDF start tag including several namespace declarations of the form:

```
<?Xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE uridef[
  <!ENTITY rdf
"http://www.w3.org/1999/02/22-rdf-syntax-ns">
```

```
  <!ENTITY rdfs
"http://www.w3.org/2000/01/rdf-schema">
  <!ENTITY xsd
"http://www.w3.org/2000/10/XMLSchema">
  <!ENTITY daml
"http://www.daml.org/2001/03/daml+oil">
  <!ENTITY daml
"http://www.daml.org/services/daml-s/0.9/process">
]>

<rdf:RDF
  xmlns:rdf =     "&rdf;#"
  xmlns:rdfs =    "&rdfs;#"
  xmlns:xsd =     "&xsd;#"
  xmlns:daml =    "&daml;#"
  xmlns:CYC =     "&cyc;#"
>
```

So in this document, the rdf: prefix should be understood as referring to things drawn from the namespace called http://www.w3.org/1999/02/22-rdf-syntax-ns#. This is a conventional RDF declaration appearing verbatim at the beginning of almost every rdf document. The second and third declarations make similar statements about the RDF Schema and XML Schema datatype namespaces. The fourth declaration says that in this document, elements prefixed with daml: should be understood as referring to things drawn from the namespace called http://www.w3.org/2001/03/daml+oil#. This again is a conventional DAML+OIL declaration. We use the XML entity model to use shortcuts with referring to the URIs.[2] The other DAML+OIL ontologies used in the FrameNet description include the DAML-S (http://www.daml.org/services) service ontologies, the OpenCYC DAML ontology (http:// www.cyc.com/2002/04/08/cyc.daml), and the SRI time ontology (http:// www.ai.sri.com/ daml/ontologies/ sri-basic/1-0/Time.daml) which is currently being revised with the new DAML+OIL time ontology effort. http://www.icsi.berkeley.edu/ snarayan/frame-2.daml has a complete namespace and imported ontology list.

The most general object of interest is a *frame*. We define the FRAME class as a *daml:class* We then define a bunch of bookkeeping properties on the FRAME class. An example of the name property is shown below.

```
<daml:Class rdf:ID="Frame">
  <rdfs:comment> The most general class </rdfs:comment>
</daml:Class>

<daml:ObjectProperty rdf:ID="Name">
  <rdfs:domain rdf:resource="#Frame"/>
  <rdfs:range rdf:resource="&rdf-schema;#Literal"/>
</daml:ObjectProperty>
```

In FrameNet, the basic relation between a word (Lemma) and a frame is the Lexical Unit (LU). The domain of the Lexical Unit is a Lemma or word and its range is a Frame. An LU is defined in DAML as a property.

```
<daml:ObjectProperty rdf:ID= "LU">
  <rdfs:domain rdf:resource="#Lexeme"/>
  <rdfs:range  rdf:resource="#Frame"/>
</daml:ObjectProperty>
```

[2]Note that all URIs are globally scoped, so without this the entire path has to be specified.

Roles are relations defined on frames ranging over the specific type of the *filler*. We use *daml:objectProperty* to define the roles of a frame. The domain of a role is its frame. We leave the type of the filler unrestricted at this level, allowing specific roles to specialize this further. Note that we use the *daml:samePropertyAs* relation to specify synonyms. The fragment below specifies that Frame Element, Role, and FE are synonyms.

```
<daml:ObjectProperty rdf:ID= "role">
  <rdfs:domain rdf:resource="#Frame"/>
  <rdfs:range  rdf:resource="&daml;#Thing"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="frameElement">
  <daml:samePropertyAs rdf:resource="#role"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="FE">
  <daml:samePropertyAs rdf:resource="#role"/>
</daml:ObjectProperty>
```

We use the various constructs *daml:maxCardinality*, *daml:minCardinality*, *daml:cardinalityQ*, etc. from DAML to specify cardinality restrictions on the fillers of a role property. The markup fragment below shows the specification of a single valued role.

```
<daml:ObjectProperty rdf:ID= "singleValuedRole">
  <rdfs:domain rdf:resource="#Frame"/>
  <rdfs:range>
   <rdfs:subClassOf>
    <daml:Restriction daml:maxCardinality="1">
      <daml:onProperty rdf:resource="#Role"/>
    </daml:Restriction>
   </rdfs:subClassOf>
</daml:Class>
```

The relation between frames (such as ARREST) and CRIMINAL PROCESS is often captured by a set of bindings between frame elements (such as the *arrested person* is the same individual as the *person charged* who is the same individual as the *defendant* in a criminal process). To capture such bindings, we introduce a special relation called *bindingRelation* whose domain and range are roles (either from the same or different frames).

```
<daml:ObjectProperty rdf:ID="bindingRelation">
  <rdfs:domain rdf:resource="#Role"/>
  <rdfs:range  rdf:resource="#Role"/>
</daml:ObjectProperty>
```

By far the most important binding relation is the identification of roles (i.e. they refer to the same value (object)). This can be specified through the relation *identify* which is a *subProperty* of *bindingRelation*. Note that in order to do this, we have to extend the DAML+OIL language which does not allow properties to be defined over other properties. We use the DAML-S ontology primitive *daml-s:sameValuesAs* to specify the *identify* relations.

```
<daml:ObjectProperty rdf:ID="identify">
  <rdfs:subPropertyOf rdf:resource="#bindingRelation"/>
  <rdfs:domain rdf:resource="#Role"/>
  <daml-s:sameValuesAs rdf:resource="#rdfs:range"/>
</daml:ObjectProperty>
```

In FrameNet, a frame may inherit (A ISA B) from other frames or be *composed* of a set of subframes (which are frames themselves). For instance, the frame CRIMINAL PROCESS has subframes that correspond to various stages (ARREST, ARRAIGNMENT, CHARGE, etc.). Subframe relations are represented using the *daml:objectProperty*.[3]

```
<daml:ObjectProperty rdf:ID="subFrameOf">
  <rdfs:domain rdf:resource="#Frame"/>
  <rdfs:range rdf:resource="#Frame"/>
</daml:ObjectProperty>
```

A central relation between subframes is one of temporal ordering. We use *precedes* (in the sense of immediately precedes)) to encode this relation between subframes.

```
<daml:ObjectProperty rdf:ID="precedes">
  <rdfs:domain rdf:resource="#subFrame"/>
  <rdfs:range rdf:resource="#subFrame"/>
</daml:ObjectProperty>
```

We can define a property *temporalOrdering* that is the transitive version of *precedes*.

```
daml:TransitiveProperty rdf:ID="TemporalOrdering">
  <rdfs:label>TemporalOrdering</rdfs:label>
</daml:TransitiveProperty>
```

Note that the *temporalOrdering* property only says it is transitive, not that it is a transitive version of *precedes*. DAML+OIL does not currently allow us to express this relation. (see http://www.daml.org/2001/03/daml+oil-walkthru#properties).

Frame Elements may also inherit from each other. We use the rdfs:subPropertyOf to specify this dependences. For example, the following markup in DAML+OIL specifies that the role (Frame Element) MOTHER inherits from the role (Frame Element) PARENT. Note we can add further restrictions to the new role. For instance, we may want to restrict the filler of the MOTHER to be female (as opposed to animal for PARENT).

```
<daml:ObjectProperty rdf:ID="mother">
  <rdfs:subPropertyOf rdf:resource="#parent"/>
  <rdfs:range rdf:resource="#Female"/>
</daml:ObjectProperty>
```

With these basic frame primitives defined, we are ready to look at an example using the Criminal Process frames.

3.1 An Example: The Criminal Process Frame

The basic frame is the CRIMINAL PROCESS Frame. It is a type of background frame. CP is used as a shorthand for this frame.

```
<daml:Class rdf:ID="CriminalProcess">
  <rdfs:subClassOf rdf:resource="#Frame"/>
</daml:Class>

<daml:Class rdf:ID="CP">
  <daml:sameClassAs rdf:resource="#CriminalProcess"/>
</daml:Class>
```

The CRIMINALPROCESS frame has a set of associated roles. These roles include that of COURT, DEFENDANT, PROSECUTION, DEFENSE, JURY, and CHARGES. Each of these roles may have a filler with a specific semantic type restriction. FrameNet does not specify the world knowledge and ontology required to reason about Frame Element filler types. We believe that one of the possible advantages in encoding FrameNet data in DAML+OIL is that as and when ontologies become available on the web (uch as OpenCYC), we can link to them for this purpose.

In the example fragment below we use the CYC *Court-Judicial* collection to specify the type of the COURT and the CYC *Lawyer* definition to specify the type restriction on the frame element DEFENSE. For illustrative purposes, the DAML+OIL markup below shows the use of a different ontology (from CYC) to restrict the defendant to be of type PERSON as defined in the example ontology. This restriction uses the DAML+OIL example from http://www.daml.org/2001/03/daml+oil-ex)

```
<daml:ObjectProperty rdf:ID="court">
  <rdfs:subPropertyOf rdf:resource="#FE"/>
  <rdfs:domain rdf:resource="#CriminalProcess"/>
  <rdfs:range rdf:resource="&CYC;#Court-Judicial"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="defense">
  <rdfs:subPropertyOf rdf:resource="#FE"/>
  <rdfs:domain rdf:resource="#CriminalProcess"/>
  <rdfs:range rdf:resource="&CYC;#Lawyer"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="defendant">
  <rdfs:subPropertyOf rdf:resource="#FE"/>
  <rdfs:domain rdf:resource="#CriminalProcess"/>
  <rdfs:range rdf:resource="&daml-ex;Person"/>
</daml:ObjectProperty>
```

The set of binding relations involves a set of role identification statements that specify that a role of a frame (subframe) has the same value (bound to the same object) as the role of a subframe (frame). We could specify these constraints either a) as anonymous subclass restrictions on the criminal process class (see http://www.daml.org/2001/03/daml+oil-ex for examples) or b) we could name each individual constraint (and thus obtain a handle onto that property). We chose the later method in our DAML+OIL encoding of FrameNet to allow users/programs to query any specific constraint (or modify it). Note also that the use of the dotting notation (A.b) to specify paths through simple and complex frames and is not fully supported in DAML+OIL (see http://www.daml.org/services/daml-s/2001/10/rationale.html and also (Narayanan and McIlraith, 2002) for more info).

[3]The *subFrameOf* relation has a direct translation to a richer semantic representation that is able to model and reason about complex processes (such as buying, selling, reserving tickets) and services on the web. While the details of the representation are outside the scope of the this paper, the interested reader can look at (Narayanan and McIlraith, 2002) for an exposition of the markup language and its operational semantics.

```
<daml:ObjectProperty rdf:ID="prosecutionConstraint">
  <rdfs:subPropertyOf rdf:resource="#identify"/>
   <rdfs:domain rdf:resource="#CP.prosecution"/>
   <rdfs:range rdf:resource="#Trial.prosecution"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="defendantConstraint">
  <rdfs:subPropertyOf rdf:resource="#identify"/>
   <rdfs:domain rdf:resource="#CP.defendant"/>
   <rdfs:range rdf:resource="#Arrest.suspect"/>
</daml:ObjectProperty>
```

Subframes of the CRIMINALPROCESS frame are defined by their type (LexicalFrame or a BackgroundFrame). For example, ARREST and ARRAIGNMENT are Lexical Frames while TRIAL is a BackgroundFrame (all are subframes of CRIMINALPROCESS. We subtype the *subFrameOf* property to specify the individual subframe relations (shown below for the relation subframeOf(Criminal Process, Arraignment)).

```
<daml:Class rdf:ID="Arrest">
<rdfs:comment> A subframe </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#LexicalFrame"/>
</daml:Class>

<daml:Class rdf:ID="Arraignment">
<rdfs:comment> A subframe </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#LexicalFrame"/>
</daml:Class>

<daml:Class rdf:ID="Trial">
<rdfs:comment> A subframe </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#BackgroundFrame"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="arraignSubFrame">
  <rdfs:subPropertyOf rdf:resource="#subFrameOf"/>
   <rdfs:domain rdf:resource="#CP"/>
   <rdfs:range rdf:resource="#Arraignment"/>
</daml:ObjectProperty>
```

To specify the the relation *precedes(Arrest, Arraignment)* we restrict the property *precedes* within (the domain of) the ARREST frame to have as one of its range values the frame (class) ARRAIGNMENT. This is done using the property restriction feature with DAML+OIL as follows.

```
<daml:Class rdf:about="#Arrest">
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#precedes"/>
      <daml:hasClass rdf:resource="#Arraignment"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

With this markup of the ontology, we can create annotation instances for examples with targets that belong to the CRIMINALPROCESS (or its associated) frames.

At the current stage, we have converted all of FrameNet 1 data (annotations and frame descriptions) to DAML+OIL. The translator has also been updated to handle the more complex semantic relations (both frame and frame element based) in FrameNet 2. We plan to release both the XML and the RDF-based DAML+OIL versions of all FrameNet 2 releases.

4 Examples of Annotated Sentences

4.1 Basic Annotation of Verb Arguments and Complements as Triplets

Consider the following sentence, which is annotated for the target *nab*, a verb in the ARREST frame; the frame elements represented are the arresting AUTHORITIES, the SUSPECT and the TIME when the event took place:

> [$_{Authorities}$ Police] **nabbed** [$_{Suspect}$ the man], who was out on licence from prison, [$_{Time}$ when he returned home].

The phrase *who was out on licence from prison* provides additional information about the SUSPECT, but it is not syntactically an argument or complement of the target verb *nab*, nor semantically an element of the ARREST frame, so it is not annotated.

How do we intend to represent this in XML conforming to the proposed standards? The header of the file will refer to the FrameNet Data Category specification discussed in the last section, but hereafter we will omit the domain name space specifications and use a more human-readable style of XML. The conversion to the full ISO style should be straightforward.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 [DOCTYPE definitions like those shown in
   the preceding section go here ]
3 <lexunit-annotation name="nab" frame="Arrest" pos="V">
4   <definition>COD: catch (someone) doing something
               wrong. </definition>
5  <subcorpus name="V-001-all">
```

The entity <lexunit-annotation>, which comprises the rest of the file includes attributes giving the name of the lexical unit (nab), the name of the frame (ARREST), and the part of speech of the lemma (verb). The first included element is a definition of the lemma within the frame, seen on line 4.

The entities contained within the lexunit-annotation are called subcorpora; each represents a particular syntactic pattern, combination of collocates, etc. In the case of *nab*, there are so few instances of the word that we have lumped them all into one subcorpus as indicated by the subcorpus name "all" on line 5. It might seem logical that the entities within the subcorpus should be sentences, but in fact, we recognize the possibility that one sentence might be annotated several times, for several targets. There might even be several instances of the **same** target lemma in the same sentence in the same frame (e.g. *The FBI* ***nabbed*** *Jones in NYC, while the Mounties* ***nabbed*** *Smith in Toronto*), each with its own set of FEs. Therefore, the next smaller entity is the **annotation set** (line 6).

The annotation set[4], shown below, consists of the <sentence>, which contains only the <text> of the sentence, and a set of layers, each consisting of a set of labels. Each label has attributes start and end, giving the stating and ending position in the text to which it is applied. This sentence is typical of the basic FrameNet annotation style, in that there are three main layers, one for frame elements ("FE", line 8), one for the phrase type (PT) of each FE (line 22), and one for the grammatical function (GF) of each FE (line 15). In each case, there are three coextensive labels; thus the word *Police*, in text positions 0-5 expresses the FE AUTHORITIES (line 10), has the phrase type "NP" (line 24) and is the subject of the verb *nab*, which we refer to as external argument "Ext" (line 17). The other two frame elements are shown by similar triplets, SUSPECT-NP-Obj and TIME-Swh-Comp, the latter meaning a complement of the verb consisting of a clause (S-node) introduced by a WH-relative.

```
 6 <annotationSet status="MANUAL">
 7  <layers>
 8    <layer name="FE">
 9      <labels>
10        <label name="Authorities" start="0"
               end="5" />
11        <label name="Suspect" start="14" end="20" />
12        <label name="Time" start="61" end="81" />
13      </labels>
14    </layer>
15    <layer name="GF">
16      <labels>
17        <label name="Ext" start="0" end="5" />
18        <label name="Obj" start="14" end="20" />
19        <label name="Comp" start="61" end="81" />
20      </labels>
21    </layer>
22   <layer name="PT">
23      <labels>
24        <label name="NP" start="0" end="5" />
25        <label name="NP" start="14" end="20" />
26        <label name="Swh" start="61" end="81" />
27      </labels>
28    </layer>
29    <layer name="Sent" />
30    <layer name="Other" />
31    <layer name="Target">
32     <labels>
33       <label name="Target" start="7" end="12" />
34     </labels>
35   </layer>
36   <layer name="Verb" />
37 </layers>
38 <sentence aPos="34400709">
39   <text>Police nabbed the man, who was out on
   licence from prison,  when he returned home.
    </text>
40 </sentence>
41 </annotationSet>
```

[4]The XML shown here is somewhat simplified from the representation being distributed by FrameNet, which includes attributes on each label giving an ID number, the date and time of creation, the name of the annotator, etc. In these examples, we use several XML tags without defining them. Without going into unnecessary detail, we note here that they can be defined in the DCS and the Dialect specification as described in (Ide and Romary, 2001a). We are also using a condensed notation with multiple attributes on entities for reasons of space, although proper RDF requires that they be split out.

There are three other layers shown in the example, none of which contain labels, called Sentence, Verb, and Other. The layer Target contains the single label Target; the fact that *nab* is the target word is indicated in the same way as the information about FEs.

Note that this XML format is "standoff" annotation in the sense that the labels refer to text locations by character positions (allowing any number of labels on various layers, overlapping labels, etc.), but that the text and the annotations appear in the same document. This is contrary to the general sense of the ISO standard, which uses indirect pointers to an entirely separate document containing the primary data. The indirect approach has certain advantages, and where the primary data is audio or video, is virtually unavoidable. But in the case of the current FrameNet data, where the annotations all apply to individual sentences, there seem to be some advantages, at least for human readers, of having the text of the sentence and the annotation contained within a fairly low-level XML entity, allowing the reader to glance back and forth between them.[5] In formulating standards for linguistic annotation, it might be wise to take these advantages and disadvantages into consideration; perhaps either situation might be allowable under the standard.

4.2 Other Types of Annotation

As the basic unit of annotation is the label, which can be applied to anything ranging from a single character to an entire sentence, and there are no a priori constraints on labels overlapping, a great variety of information can be represented in this way. We will not be able to demonstrate all the possibilities here, but we will give a some representative examples.

In FrameNet, event nouns are annotated in the same frame (and hence with the same FEs) as the corresponding verbs; the main differences are that the syntactic patterns for the FEs of nouns are more varied, and (with rare exceptions), no FEs of nouns are **required** to be expressed. Consider the noun *arrest*, also in the ARREST frame, in the sentence:

> Two witnesses have come forward with information that could lead to [$_{Suspect}$ the killer 's] **arrest** .

In this case the SUSPECT is expressed as a possessive (*the killer's*; it could equally well have been in a PP headed by *of* (*the arrest of the killer*).

```
<annotationSet status="MANUAL">
```

[5]The location of the sentences in the original corpora is still recoverable from the **aPos** attribute, which gives the absolute position from which the sentence was abstracted. The name of the corpus is given in another attribute which has been omitted in the example.

```
  <layers>
    <layer name="FE">
      <labels>
        <label name="Suspect" start="68" end="80" />
      </labels>
    </layer>
    <layer name="GF">
      <labels>
        <label name="Gen" start="68" end="80" />
      </labels>
    </layer>
    <layer name="PT">
      <labels>
        <label name="Poss" start="68" end="80" />
      </labels>
    </layer>
    <layer name="Sent" />
    <layer name="Other" />
    <layer name="Target">
      <labels>
        <label name="Target" start="82" end="87" />
      </labels>
    </layer>
    <layer name="Noun" />
  </layers>
  <sentence aPos="102536044">
    <text>Two witnesses have come forward with
information that could lead to the killer's arrest.
    </text>
  </sentence>
</annotationSet>
```

In addition to marking the FE SUSPECT from ARREST, we could also annotate the same sentence again in the CAUSATION frame with the target **lead**, which would create an annotation set listed under the the LU *lead_to*:

> Two witnesses have come forward with [$_{Cause}$ information that] could **lead** [$_{Effect}$ to the killer's arrest].

The same sentence would be annotated in two different frames, and the semantics of the two frames could (in theory) be combined compositionally to get the semantics of the phrase *information that could lead to the killer's arrest.* Similar processes of annotating in multiple frames with targets **come_forward** (and possibly **witness** as well) should yield a full semantics of the sentence. [6]

[6]The qualification "in theory" is included because the present phase of the FrameNet project is not undertaking to implement a system of semantic composition; we are just trying to annotate enough examples in enough frames to provide a basis for semantic parsing (in this context, automatic FE recognition) and composition of annotation sets.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In ACL, editor, *COLING-ACL '98: Proceedings of the Conference, held at the University of Montréal*, pages 86–90. Association for Computational Linguistics.

Charles J. Fillmore and B.T.S. Atkins. 1992. Towards a frame-based lexicon: The semantics of RISK and its neighbors. In Adrienne Lehrer and Eva Feder Kittay, editors, *Frames, Fields and Contrasts*. Lawrence Erlbaum Associates.

Charles J. Fillmore, Collin F. Baker, and Hiroaki Sato. 2002. The framenet database and software tools. In *Proceedings of the Third International Conference on Languag Resources and Evaluation*, volume IV, Las Palmas. LREC.

Charles J. Fillmore. 1976. Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, volume 280, pages 20–32.

Charles J. Fillmore. 1977a. The need for a frame semantics in linguistics. In Hans Karlgren, editor, *Statistical Methods in Linguistics*. Scriptor.

Charles J. Fillmore. 1977b. Scenes-and-frames semantics. In Antonio Zampolli, editor, *Linguistic Structures Processing*, number 59 in Fundamental Studies in Computer Science. North Holland Publishing.

Charles J. Fillmore. 1982. Frame semantics. In *Linguistics in the Morning Calm*, pages 111–137. Hanshin Publishing Co., Seoul, South Korea.

Nancy Ide and Laurent Romary. 2001a. A common framework for syntactic annotation. In *Proceedings of ACL 2001*, pages 298–305, Toulouse. ACL.

Nancy Ide and Laurent Romary. 2001b. Standards for language resources. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 141–149, Philapdelphia. IRCS.

Nancy Ide, Adam Kilgarriff, and Laurent Romary. 2002. A formal model of dictionary structure and content. In *Proceedings of Euralex 2000*, pages 113–126, Stuttgart. EURALEX.

ISO. 2002. Iso tc 37-4 n029: Linguistic annotation framework. Internet. http:// www.tc37sc4.org/ document.htm.

John B. Lowe, Collin F. Baker, and Charles J. Fillmore. 1997. A frame-semantic approach to semantic annotation. In Marc Light, editor, *Tagging Text with Lexical Semantics: Why, What and How?* Special Interest Group on the Lexicon, Association for Computational Linguistics.

Srini Narayanan and Sheila McIlraith. 2002. Simulation, verification and automated composition of web services. In *Proc. Eleventh International World Wide Web Conference (WWW2002)*, May.

Miriam R. L. Petruck. 1996. Frame semantics. In Jef Verschueren, Jan-Ola stman, Jan Blommaert, and Chris Bulcaen, editors, *Handbook of Pragmatics*. John Benjamins.