THE STRUCTURE OF THE FRAMENET DATABASE

Collin F. Baker: *International Computer Science Institute, Berkeley, California, USA* (collinb@icsi.berkeley.edu), Charles J. Fillmore: *International Computer Science Institute and University of California, Berkeley, California, USA* (fillmore@icsi.berkeley.edu) and Beau Cronin (bcronin@icsi.berkeley.edu) *International Computer Science Institute, Berkeley, California and Massachusetts Institute of Technology, USA*

Abstract

The FrameNet database contains descriptions of more than 7,000 lexical units based on more than 130,000 annotated sentences. The database and its related software are central to the process of entering lexical information, annotating sentences, displaying the results, and distributing the FrameNet data. This article discusses both how the design of the database follows the principles of frame semantics and also how the database provides appropriate access for project staff and users elsewhere.

1. Introduction: Conceptual Relations and Relations in Databases

The basic product of the FrameNet (FN) project is its data; in this chapter we will examine the form in which this data is stored and manipulated internally and discuss why this form was chosen. The project has received two NSF grants, each for a period of three years;[1] we refer to these phases of the project as FrameNet I and FrameNet II. The change from FrameNet I to II coincided with a major change in the representation of the data; this section will discuss what was changed at that time and why.

The data model used in FrameNet I (Lowe et al. 1997, Baker et al. 1998) represented all the annotation on the sentence as text with SGML mark-up[2], with the frame element (FE), phrase type (PT) and grammatical function (GF) of each labeled constituent marked as attributes on a general "constituent" element <C>, as in this marked-up version of the sentence *I am conscious that it is a difficult and complex subject* : [3]

<S TPOS = "81597120">… <C FE = "Cog" PT = "NP" GF = "Ext">I </C> <C TYPE = "SuppV" PT = "XFE" GF = "XFE">am </C> <C TARGET = "y"> conscious </C> <C FE = "Cont" PT = "Sfin" GF = "Comp">that it is a complex and difficult subject</C> . </S>

International Journal of Lexicography, Vol. 16 No. 3

In choosing a data model for the second phase of FrameNet, we were eager to move away from such a representation, for several reasons:

(a) Storing the data as text with mark-up meant that searching across the data was slow. In order to speed up access, we resolved to use a real database, where the off-the-shelf software would provide fast access, automatic indexing, and query optimization.

(b) We wanted to deal with situations in which the same constituent might represent more than one FE, as in *The internist* ***treated*** *the diabetic*, where the `Cure` frame is evoked by the word *treated*, and *the diabetic* is the filler of both the PATIENT FE and the DISEASE FE. In this case, we would need to have an element within an element, which causes problems for much of the software for SGML/XML. Therefore, we wanted a representation in which the FE, PT and GF labels would be on separate 'layers', distinct from each other and from the text itself. (This is sometimes referred to as 'standoff mark-up'.)

(c) Making across-the-board changes (such as renaming a frame element) in hundreds of separate SGML files was tedious and error-prone. We needed a representation in which the labels of the same type would refer to a single definition of the FE, PT or GF.

This last point suggests the answer which we adopted: a relational database. The chief advantage of a relational database is that data which occurs repeatedly is entered into the database only once. Each use of that value consists of a pointer to that one entry. Of course, this means that the representation is more indirect, and contents of the database are not directly interpretable. Suppose, for example, that we have the sentence *Carlos walked to the fountain*, where the verb *walked* evokes the `Self_motion` frame, and we want to mark the phrase *to the fountain* as the GOAL frame element. In the SGML used in FrameNet I, we would have the following (leaving aside the other marking that would occur in the sentence):

<S IDNUM = "1235">Carlos walked <C FE = "Goal">to the fountain </C></S>.

In FrameNet II, the label for the GOAL is represented as several entries in the database. One entry shows that there is a label extending from character 14 to character 28 of the text, another that this label has a certain set of colors, and that it represents the GOAL in the `Self-motion` frame (as we will discuss below in section 3.2). This representation is more complex than directly inserting SGML labels, but the advantage of having information such as FE names represented only once in the database far outweighs the inconvenience due to the increased complexity. If, for example, we wanted to translate the name of the FE into German, we would only need to edit the entry in the FrameElement table, substituting ZIEL for GOAL, and all appearances of the FE name, on screen or in reports, would be changed.

The relational database which holds the FrameNet II data has been designed so that its structure, so far as practical, models the conceptual structure on which

the project is based. But the things we want to represent are rather disparate. On the one hand, we have hundreds of frames and thousands of frame elements (each specific to a given frame), defined 'by hand', as described in the article 'FrameNet in Action' in this volume. On the other hand we have more than 130,000 sentences with their associated annotations. Since there are typically two or more sets of FE, GF and PT labels on each annotated sentence, the tables containing the annotation are hundreds of times larger than those containing the frames and FEs. It is therefore convenient to consider separately the part of the database that represents the frames, FEs, lexical units, etc. from the part that contains the sentences and their annotation. We will refer to these as the 'lexical database' and the 'annotation database' respectively, and will discuss them separately below, although they are implemented in a single MySQL database. Note that the structure of the relational database has also changed in several relatively minor ways over the course of FrameNet II; these changes will not be discussed here.[4] Likewise, we refer to staff engaged in defining frames and lexical units and setting the parameters for extraction of example sentences as the 'vanguard', and those engaged in marking frame elements on sentences as 'annotators', even though most staff members do both sorts of work.

2. The Lexical Database

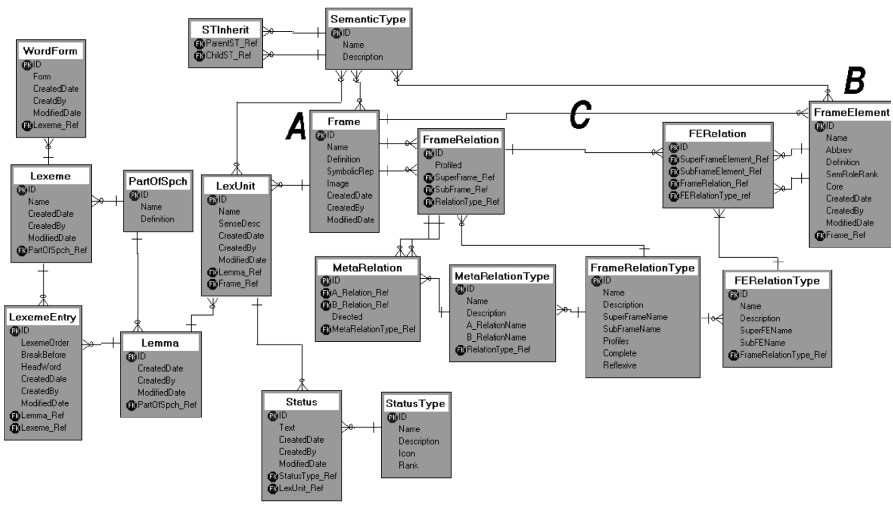

Figure 1

2.1. *Frames and Frame Elements*

The basic units of Frame Semantics (Fillmore 1976, 1977b, 1982, 1985) are frames and the frame elements (FEs) that comprise them. In Fig. 1 above, this situation is represented by the tables Frame (A) and FrameElement (B), and the one-to-many relation between them (C), which indicates that each frame

element (FE) is defined with regard to exactly one frame, and that frames are typically associated with more than one frame element. Because FEs are defined relative to frames, FEs in different frames may have identical names, without implying any relation between them. In practice, of course, we strive to give frame elements meaningful names, so it is not entirely accidental that more than 70 frames have FEs with the name AGENT. Nevertheless, one cannot conclude anything from this fact alone; if they are all related by a more general concept of agent, this will have to be stated explicitly, by entering FE-to-FE relations in the database, as will be discussed below.

2.2. *Lemmas, Lexemes, Word Forms, and Parts of Speech*

As different researchers use different terminology in this area, it is essential to define our terms at the start of the discussion. By *word form*, we mean one of the forms of a word differing by inflection; by *lexeme*, we mean any of the inflectional versions of a word, represented by the uninflected stem: the singular of a noun, the unmarked infinitive of a verb, the plain form of an adjective, or the single form of an uninflected word. Typically, one English noun lexeme is associated with two word forms (singular and plural) and one English verb lexeme with four (*need, needs, needed, needing*), although irregularities increase these numbers slightly. To handle multiword expressions, we posit a higher level or organization, the *lemma*, composed of one or more *lexemes*. In the left part of Fig. 1, we see the tables relating lemmas, parts of speech, lexemes, and word forms. As the connectors indicate[5], each lemma has one part of speech, as does each lexeme. Each lexeme is associated with one or more word forms, but each word form is associated with only one lexeme. (This entails some redundancy in the word form table, but is simpler than carefully maintaining the links that would be required for more parsimonious storage.)

The LexemeEntry table is needed to represent *multiword expressions* (MWEs), such as verb+particle (*take off*), N-N compounds (*family practitioner*), and longer constructions (*Martin Luther King Day, have bats in one's belfry, an X's paradise*). For the sake of consistency, **all** lemmas, even those with only one lexeme, are connected to their lexemes via the LexemeEntry table. In the case of MWEs, the fields in this table indicate not only the order of the lexemes, but also which lexeme is the lexical head, and whether or not the MWE is separable. For example, the lemma *go broke* is comprised of the two lexemes *go* and *broke*; the first is the head and undergoes the usual inflection for the lexeme go while the latter is invariant (**went broken*). Therefore, the first lexeme will be associated with the appropriate five word forms (*go*, *goes*, *went*, *gone*, *going*), while the second lexeme has only one word form, not related (as least in our database) to the lexeme *break* (v). To give another example, there would be two lexical units (in quite different frames) containing the lexemes *take off*, one separable (*take your sweater off*) and one inseparable (*the plane took off*). In both of these, the lexeme *take* would be marked as the head of the

lemma. In the "wear" sense, the LexemeEntry table for *off* would also have the field called Break_before set to 'true', to indicate that light NPs usually break the lemma by being inserted before it (*took them off*).[6]

2.3. *Frames, Lemmas and Lexical Units*

FrameNet measures its progress in terms of Lexical Units (LUs)[7], which are defined as an association between a lemma and a frame. Since lemmas are units of form and frames represent meaning, lexical units correspond roughly to dictionary senses. Each LU thus has a link to a single frame and a single lemma. Many lemmas are associated with more than one frame, and thus constitute more than one LU; this is how FrameNet represents polysemy. The meaning of the LU is also expressed in words in the Sense Description field of the LexUnit table.

Each LU also has its own Name field, in addition to the name of the lemma, because the same lemma can appear twice in the same frame in different senses. A clear example is the noun *possession* which occurs in two different LUs in the `Possession` frame, one referring to the things possessed and the other to the state of possessing something (*Her most precious possession was a diamond ring* vs. *Possession is nine tenths of the law*). The names of these two LUs are *possession* and *possession of goods* respectively.

One or more statuses can be associated with each LU. Some of these are temporary, used for keeping track of the state of work on each lexical unit, e.g. whether subcorpora have been prepared for it, whether it is in the process of annotation, whether there is an unresolved problem regarding the frame and whether the LU belongs in it, etc. Others will be of interest to the end users of the FrameNet data, as they indicate the final disposition of the lexical unit, as shown in Table 1:

Table 1: Lexical Unit Statuses

Status	Definition
Finished	Enough examples were found and annotated during FrameNet II.
FN1 Sent	Annotated during FrameNet I.
Insufficient attestations	The LU has been defined in a frame, but we have not found enough good examples to annotate.
By analogy	This LU is one of a group of LUs that behave identically (such as the days of the week), and so only one or two of the group will be annotated; the others simply marked as similar.

2.4. *Frame-to-Frame and FE-to-FE Relations*

If the semantics of FrameNet were limited to a list of frames and their definitions, it would be impossible to express many generalizations. For example, most frames involving motion allow FEs representing the path followed by the moving object, from a SOURCE, along a PATH proper, to a GOAL; we need a formal way to express this generalization. In FrameNet I, one type of relation among frames was expressed: frames were grouped into semantic domains, such as Communication, Motion, Society, etc. But the theoretical basis of such groupings was never made explicit; as time went on, we realized that we needed to represent notions such as one frame being a subtype of another, or one event frame being composed of a series of smaller *scenes* (Fillmore 1977a). When two frames are related, their frame elements are also often related to each other in more or less predictable ways. In the FrameNet II database, instead of the domains of FrameNet I, we have defined a variety of relations between frames and between FEs, and we have made it possible to define new relations relatively easily, by adding the tables FrameRelationType and FERelationType to the database. Each Frame-to-frame relation is associated with exactly one FrameRelationType, which may be directed or undirected, and FE-to-FE relations are handled similarly. We will discuss below some of the more important of these relations.

2.4.1 *Frame Inheritance*. The relation mentioned above among frames involving motion is a good example of *frame inheritance*, which we define as *full* and *monotonic*, with the possibility of *multiple inheritance*:

(a) *Full* inheritance means that if frame `B` inherits from frame `A`, there must be an FE in `B` corresponding to each FE in `A` ; the FE in `B` can have a different name from that in `A`, but there must be a binding between them. Child frames can have additional FEs not found in their parent(s).

(b) *Monotonic* means that if a parent frame or FE has a semantic type, the semantic type of the child frame or FE must be the same as or a subtype (elaboration) of the semantic type of the parent. (cf. Semantic types, below)

(c) *Multiple* inheritance means that a child frame (and therefore its FEs) can have any number of parents (though we rarely posit more than two in practice).

2.4.2 *The Subframe Relation*. Many frames express concepts which have natural, well-defined subparts. For example, many complex events break down into a series of smaller events, occurring in a particular order. Since all semantics is expressed in terms of frames[8], each subevent will also be a frame, and we call them subframes of the complex event. Thus, the complex frame `Picnic` will have subframes of preparing food, going to a suitable spot, eating a meal there, and returning home. Subframes typically contain FEs that are bound to FEs of their complex frame, and thus indirectly to each other. For example, in the subframes of `Picnic`, the subframes of going to the spot and returning home will contain an FE MOVER, and the eating subframe will contain an FE EATER, all of which will be bound to the FE PICNICKER of the complex frame, and thus, transitively, to each other, since one who participates in a picnic

must participate in each of those stages. The FE for the PREPARER of the food in the first subframe, on the other hand, will not be bound to the PICNICKER, since the action of preparing food is not necessarily performed by the one who goes on the picnic.

If a complex frame A is inherited by another frame B, our notion of full inheritance requires that all the subframes of A be bound to subframes of B, that is, subframe structure is also inherited. If B also inherits from another complex frame, the details of the bindings can be complex.

2.4.3 *The Uses Relation*. This is a relation like inheritance, but less strictly defined. If frame B *uses* frame A, frame B need not have an FE corresponding to each FE of frame A. We have defined this relation mainly because working out full inheritance lattices among frames has proven to be extremely contentious and time-consuming. The Uses relation allows us to define a similar relation without getting bogged down in details. For example, the Conversation frame is clearly related to the more abstract Communication frame, but the Communication frame contains an FE MESSAGE, which is not an element in the Conversation frame. We cannot say that Conversation *inherits* Communication, or we would violate the principle of full inheritance, but we can say that Conversation *uses* Communication and avoid this dilemma.

2.4.4 *The 'See also' relation.* Although distinguishing among closely related frames is often difficult, defining the differences among frames is crucial to the FrameNet enterprise. We need a way of explaining the distinctions among a group of frames in one place, without repeating this explanation for each frame in the group. We do this much as dictionaries distinguish among closely related headwords, by writing a discussion of the distinctions in a group (a synonymy), making it part of the entry for a single headword, and then referring to that entry from the definitions of the other members of the group. The 'see also' relation is a pointer from one or more frames to another frame whose definition includes a detailed discussion of the differences among the frames in the group. For example, we treat the two uses of words like *load* in *She loaded the wagon with hay* and *She loaded the hay onto the wagon* in the two frames Filling and Placing respectively. The detailed discussion of the distinction between these two frames is located in the definition of Filling and there is a 'see also' relation from Filling to Placing.

3. The Annotation Database

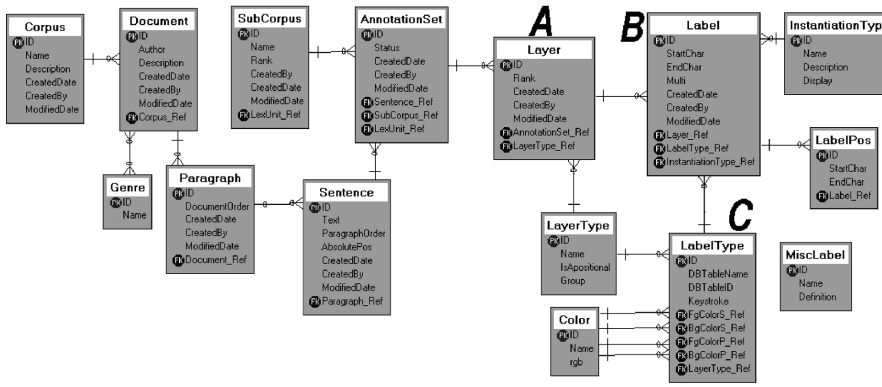

Figure 2

The day-to-day work of FrameNet consists mainly of annotating sentences chosen from a corpus as examples of a particular lexical unit. Once again, the structure of the tables used to store the sentences and annotation on them (shown in Fig. 2) reflects the structure of the concepts involved.

3.1. *Subcorpora and Sentences*

The software used to extract the example sentences for an LU allows the vanguard to exclude certain collocates (as likely to indicate the wrong sense of the lemma—"out of frame" uses), to include certain collocates, and to select for certain syntactic patterns that are believed (on the basis of some preliminary corpus searches and introspection) to be important for this LU. When presented to the annotators by the annotation software, the sentences are grouped according to these patterns into *subcorpora* and the lemma under study (the *target*) is already marked.

The idea is that (1) annotation will be easier if sentences are presented in some sort of reasonable order and (2) we want to be sure to annotate a few examples of each different pattern in which the lemma occurs; having patterns grouped together helps annotators keep track of how many of each pattern have been done. Of course, the vanguard cannot always foresee all the relevant patterns, so there are fairly large subcorpora for 'other' sentences which contain the target but don't fit any of the patterns requested by the vanguard.

The simplest database representation for subcorpora would be to create a link directly from the sentences to the subcorpus. But the situation is actually a little more complicated. For each target, there is a set of *annotation layers* for the FEs, phrase types, grammatical functions, etc.; each such set is represented by an entry in the AnnotationSet table, linking a sentence, a subcorpus, and (via the subcorpus) an LU. This makes it possible for one sentence to be annotated more

than once, with regard to more than one target word; the two targets will be linked to different annotation sets (and typically to different LUs).

3.2 *Annotation Sets, Layers, and Labels*

The annotation process is conceived of as labeling portions of the example sentences with labels indicating various relevant syntactic and semantic properties. A constituent of a sentence may represent a particular frame element; for example, in the sentence *Helmut **saw** a tall, black figure against the shining snow*, with *see* as the target in the `Perception_passive` frame, *Helmut* expresses the FE PERCEIVER-PASSIVE, *a tall, black figure*, the FE PHENOMENON, and *against the shining snow*, the FE GROUND. We further specify the phrase types of each of these constituents as NP, NP and PP respectively. We also describe their grammatical function relative to the target as subject (which we refer to as the *external argument*[9] of *see*), object, and complement respectively. We think of these three sets of labels as being on three more or less independent *layers*, called FE, PT and GF, as shown below:

Table 2: Layered Annotation

(TEXT)	*Helmut*	***saw***	*a tall, black figure*	*against the snow*
FE	PERCEIVER-PASSIVE		PHENOMENON	GROUND
PT	NP		NP	PP
GF	EXT		Obj	COMP

These correspond fairly directly to the Layer and Label tables in the database (A and B respectively in Fig.2). Each layer has a type with a name (here FE, PT, and GF) contained in the LayerType table, and each layer contains zero or more labels. Most labels are contiguous, and their extent is indicated simply by the startChar and endChar fields, but others may be discontiguous, in which case, the additional spans will be indicated by entries in the LabelPos table. Each label also has a type in the LabelType table (C in Fig. 2), which contains the fields DBTableName and DBTableID which together form a pointer to the actual content of the label. For FEs, the DBTableName is 'FrameElement' and the ID points to the appropriate FE in the FrameElement table. For all other types of label, the DB table name is 'MiscLabel'; this table contains the names of the PTs, GFs, etc.

Some FEs are considered to be conceptually present in the context, even though not expressed in the sentence; this is referred to as *null instantiation*. Information about null instantiated FEs is stored in the InstantiationType table.[10]

In addition to the FE, GF, and PT layers, annotators also add labels on other layers, all of which are represented similarly. Certain syntactic information is represented by adding labels on the part-of-speech-specific layer. The most

notable among these is marking of support verbs for nouns and adjectives. For example, in *He made a **comment** about my work*, with the target *comment* in the `Statement` frame, *made* would be labeled 'Supp' on the part-of-speech-specific layer. This layer also contains labels which can be used to indicate what word is the governor of a target noun or adjective. There is also a layer marking the part of speech label for each word, which are imported along with the text when the sentences are imported; this layer is normally not used by the annotators.[11]

If two FEs for the same target partially overlap, annotators can add another FE layer on which to place the label. For example, in *His arm was **broken** in the accident*, with *break* as the target in the `Experience_bodily_harm` frame, *his arm* is marked with the FE BODY_PART, but within that, *his* can be additionally labeled with the FE EXPERIENCER on a second FE layer.

As mentioned above, sentences can be annotated with regard to more than one target, which means that they will contain more than one annotation set, and probably be associated with more than one LU; each annotation set has all of the features described above, with its own layers, labels, etc. In this way, a sentence can be annotated with respect to several frame-evoking expressions, each of which contributes to the meaning of the whole. For example, *Matilde **learned** that Rufus had **gone** to Cairo* might be annotated once with the target *learn*, in the `Becoming_aware` frame and again with respect to the target *go* in the `Self-motion` frame. *That Rufus had gone to Cairo* would be labeled simply as the CONTENT FE in `Becoming_aware`, but more detailed characterization of the content would come from the `Self-motion` frame. The FrameNet project does not undertake to develop a theory as to how the semantic structure of the lower clause is to be integrated with the semantic structure of the matrix clause, but merely to describe the two pieces separately.

3.3. *Corpora, Documents, and Paragraphs*

Although the primary purpose of FrameNet is to collect annotated examples for lexicographic purposes, it is important to demonstrate that a Frame Semantic analysis can be extended to full running text as well. Therefore, we have created tables (seen at the left side of Fig. 2) that allow sentences in the database to be contained in paragraphs, paragraphs in documents, and documents in corpora, and software for importing continuous texts. We have done this so far only for 'toy' corpora, but the mechanism is ready if whole documents are to be imported and fully annotated.[12]

4. Semantic Types

Certain semantic generalizations are not easily made in terms of frames and relations between them. For example, some FEs require that their fillers be human. Some LUs in the `Judgment` frame express positive judgments and

others negative judgments; we want to distinguish these without creating two separate frames. Some set of frames might be associated with a particular technical domain (such as law or biochemistry), yet not related through inheritance or uses relations. These notions can, of course, be expressed in words as part of the definitions of the respective frames, FEs, and LUs, but we would want them to be clear-cut and easily retrievable. For this purpose, we have created a very general table of semantic types, which can be attached to LUs, frames, or FEs. Thus, an entry can be made in the SemanticType table for 'human' and connected with certain FEs, or for 'positive' and associated with certain LUs. In some cases, we want to create a small semantic type hierarchy, such as identifying human as a subtype of animate; this can be expressed through the STInherit table. We plan to incorporate a semantic type hierarchy to aid in a proposed system for semi-automatic FE recognition which is now under development.

5. Notes and Note Types

We have found it convenient for a variety of purposes to have the ability to attach notes to objects and sets of objects within the FN database. We want to record questions and problems that arise in the course of defining frames and LUs, or during the annotation process. The note table allows staff to write notes to each other and attach them to the object(s) in question, so that a different person working in the same area later does not need to start from scratch on the same questions.

6. User interfaces to the Database

The database structure described above is implemented as a MySQL database running under Solaris 8 OS on Sun hardware. Three types of interface have been developed in the FrameNet project for browsing, searching, and updating the databases.

6.1. *Web-based Browsing*

The FN home pages (http://framenet.icsi.berkeley.edu/~framenet) display nearly all the data in the FN database in a more human-readable format. These are similar to the set of web reports used by the FN staff, which are generated from the database, both automatically each day and on demand.

6.2. *Web-based Searching*

The FrameSQL program is a web-based search engine created and maintained by Prof. Hiroaki Sato of Senshu University, in Kawasaki, Japan. It has a browse mode and two search modes, designed for beginning or experienced users,

although using the search tool effectively depends on a thorough understanding of the nature of the data. We have found it invaluable for discovering unusual syntax-semantics combinations, some of which represent errors in the data (which we then correct) and some of which are simply interesting facts about the language. Prof. Sato has generously made FrameSQL available from the FrameNet website through a link to his website in Japan.[13]

6.3 *The FrameNet Desktop software*

A single GUI (Graphical User Interface) written in Java allows the users to interact with the lexical database, the annotation database and the report system, combining the frame editing functions and the annotation functions. The software is split into client applications, which can be running concurrently and an application server which mediates between the clients and the database. Its operation is discussed in this volume in the article 'FrameNet in Action'; some proposed changes are discussed in the next section of this article.

7. Future Directions

7.1. *Distributing the FrameNet data in XML*

FrameNet is committed to making our data available for research and teaching by users with different needs and different facilities, including those who do not need or want to duplicate our software system. Although we have moved away from using SGML/XML as the basic data representation, XML is the logical format in which to distribute our data to researchers who want in integrate it into an existing software system.

For example, a number of research projects on machine translation have expressed interest in using FrameNet frames. The assumption is that certain types of activities (such as cooking, reading, and replacing one thing with another) are similar enough across cultures that the FrameNet frames can be used to group the vocabulary associated with them in both the source language and the target language. People working on such projects can download the XML version of the lexical database and use it to populate the English side of a bilingual lexicon organized by frames. Researchers building systems for natural language understanding and question-answering in English might want to download both the lexical database and the annotation database, the latter serving as a resource for analyzing the contexts (both syntactic and lexical) in which words are used in particular senses. Table 3 shows a (slightly edited and condensed) sample of the XML for the annotated sentence from the beginning of this article, *I am conscious that it is a difficult and complex subject.* The structure of the XML basically follows the structure of the relational tables, although many references are resolved, and the names of many fields are abbreviated. The XML format is not always easy for people to read, but should not be too difficult to parse automatically, using readily available XML parsers.

Table 3: Sample of Export XML Format

```
<sentence ID="327024" aPOS="81597120">
<text>As I rise to speak to the House on this important matter, I
am conscious that it is a complex and difficult subject.</text>
<layers>
  <layer ID="2615969" "lexUnitRef="155" name="FE"
frame="Awareness" lemma="conscious">
    <labels>
     <label name="Cognizer" ID="8189607" start="59" end="59"/>
     <label name="Content" ID="8189622" start="74" end="115"/>
    </labels>
  </layer>
  <layer ID="2615970" "lexUnitRef="155" name="GF"
frame="Awareness" lemma="conscious">
    <labels>
     <label name="Ext" ID="8189609" start="59" end="59"/>
     <label name="Comp" ID="8189624" start="74" end="115"/>
    </labels>
  </layer>
  <layer ID="2615971" "lexUnitRef="155" name="PT"
frame="Awareness" lemma="conscious">
    <labels>
     <label name="NP" ID="8189608" start="59" end="59"/>
     <label name="Sfin" ID="8189623" start="74" end="115"/>
    </labels>
  </layer>
  <layer ID="2615974" "lexUnitRef="155" name="Target"
frame="Awareness" lemma="conscious">
      <labels>
     <label name="Target" ID="8189613" start="64" end="72"/>
    </labels>
  </layer>
 </layers>
</sentence>
```

7.2 *Distribution of the Frame Editing and Annotation Software for Collaboration*

We welcome offers of cooperation in the work of the project, both with regard to the definition of new frames and the lexical units in them and the annotation of sentences, in English and in other languages. However, in our own work together as a team, we find that extensive face-to-face consultation is required if the definitions of frames and frame elements are to be consistently applied, and considerable effort and time has to be spent on quality control. Maintaining

consistency and coherence over work done in several different sites seems likely to be even more difficult. Nevertheless, we are eager to see the FrameNet methodology applied to the widest possible range of semantic domains and types of texts. Suggestions as to how this can be accomplished cooperatively are welcome.

7.3 *Possible changes in the Database*

The relational database has served us well in FrameNet II, and we plan to make only minor changes to it. We expect to spend most of the remaining time in this phase of the project filling in gaps and correcting errors in the data. There are many sorts of inferences and generalizations that can be drawn from the FrameNet data (e.g. Chang et al. 2002); we simply do not have the resources to work on most of these. Instead, we have concentrated on finding out the facts about the valences of lexical units and how we can fruitfully generalize about these at the frame level.

One sort of generalization that we are currently working on is adding frame-to-frame relations in the lexical database. At the moment, only a few samples of these are entered in the database, partly because earlier versions of the software made the process tedious. The new GUI includes much better tools for viewing and editing frame-to-frame and FE-to-FE relations, and handles a much wider range of such relations, so we are making rapid progress in this area.

In the longer term, in a third phase of the project, we might consider using a different sort of database to handle information about semantic types and frame-to-frame relations. These tables typically contain relatively few entries, each entered "by hand", so there is not much need for the speed provided by automatic indexing of tables in the relational database. Also, changes in our definitions of relations often require substantial changes in the structure of the relational database and the software which accesses it. For these reasons, it may be advantageous to use a hybrid approach, with a relational database for the 'heavy lifting' required in the annotation process, accessing the lexemes, etc., and a more flexible database (such as lisp or prolog clauses) to handle the hand-tailoring of relations among frames, FEs, semantic types, etc.

8. Conclusion

We hope that we have shown that, despite its complex appearance, the FrameNet database is, in fact, a fairly straightforward representation of the concepts underlying Frame Semantics and the annotation process, expressed in the style of a relational database. Fully understanding the database does require a certain amount of experience, but the Java GUI makes it easy for the staff to use it in their day-to-day work, while the MySQL back end makes searching relatively efficient. The HTML reports and the XML export format make the data accessible to users elsewhere. We look forward to the increasing use of the

FrameNet data by other teachers and researchers[14], and to increasing integration of the FrameNet data into various types of NLP systems.

Notes

[1] See the foreword of this volume for details.

[2] SGML stands for Standard Generalized Mark-up Language, and has been widely used for marking up texts for information retrieval, electronic publishing, data exchange, etc. SGML is now largely superseded by Extensible Mark-up Language, XML. See http://www.w3.org/XML/ for more information.

[3] The annotation shows the frame elements cognizer and Content, and the support verb *to be* as well as the target word itself, *conscious*. The other abbreviations are interpreted as follows: Ext-external argument (see note 7), XFE-not a frame element, Comp-complement (a term we use very broadly to cover both arguments and adjuncts), NP-noun phrase, Sfin-finite clause.

[4] The database structure discussed in this paper is the latest version, implemented along with a new software suite that was nearing completion as this article was drafted (April 2003). This means that it differs somewhat from the structure at the time of the interim FN data release (Release 1.0) in October 2002; the next data release will reflect the new structure. See Fillmore et al. (2001) for a more detailed discussion of the database changes from FrameNet I to FrameNet II.

[5] In the diagram, connections between tables are shown as lines; the ends of the lines indicate whether the relation is one-to-one, one-to-many, etc., in a fairly iconic way. A single cross-bar represents the digit 1, a small oval represents the digit 0.

[6] Heavy NPs tend not to be inserted between the two parts, even when the MWE is separable; thus it is preferable to say *She took off the sweater which her grandmother had given her for Christmas* rather than *She took the sweater which her grandmother had given her for Christmas off*. We do not consider the `flying' sense of *take off* to be separable, even though certain adverbials can be inserted, as in *took rapidly off into the wind*.

[7] Our terminology here follows Cruse (1986:77).

[8] Fillmore (1977a) talks about *frames* and *scenes*, but we do not want to postulate two fundamentally different types of entity. A scene is simply a frame considered in its relation to a larger event, i.e. a subframe. Related concepts are Schank's scripts (Schank and Abelson 1977) and Minsky's (1974) frames.

[9] Actually, external object is defined more broadly than subject. Besides regular subjects of finite verbs, we include subjects or objects of certain control verbs including support verbs and even prepositions, e.g. (from the Revenge frame, with targets marked in boldface):

(1) She had urged [GF="Ext" him] to **retaliate** for his humiliation.

(2) [GF="Ext" He] had [Supp wreaked] **vengeance** on those who humiliated him.

(3) [GF="Ext" He] had done it [Supp in] **retaliation** for his humiliation.

[10] See the "FrameNet in Action" article in this volume for a discussion of null instantiation.

[11] These will be either BNC tags or Penn TreeBank tags. They are distinct from those contained in the PartOfSpeech table, which are simply N, V, Adj, etc. and apply to lemmas or lexemes.

[12] For a worked example of full-text annotation, see Fillmore and Baker 2001. The current project which is closest to such an undertaking is probably PropBank (Kingsbury et al. 2002). A similar project for German, involving automatic labeling of all FEs in continuous text is underway at U Saarlandes (Erk et al. 2002).

[13] http://163.136.182.112/searchfn2/notes/index.html

[14] As of this writing (April 2003), more than 60 groups had downloaded or requested access to the FrameNet data, for a great variety of uses: in machine translation systems, for semantic parsing of texts for question answering systems, as a conceptual ontology for common-sense reasoning, as a resource for teaching lexical semantics, etc.