Markup and the GOLD Ontology

Scott Farrar
Universität Bremen

D. Terence Langendoen
University of Arizona

1. Introduction and goals of this paper

The EMELD project is committed to providing recommendations to the endangered languages community for the markup of electronically encoded and archived data and analyses (henceforth 'datanalyses') of endangered languages. (Of course, any recommendations we make should be applicable for work on any human language whatever—flourishing, endangered, or extinct.) Our specific markup recommendations will take the form of how to represent commonly used linguistic data structures, such as glossed texts, annotated audio and video recordings, lexicons, paradigms, and grammars, and new types of structures that exploit the resources provided by new electronic technologies, particularly open-source Web technologies.

For the content of such structures, we rely on our community to tell us what is needed. We are now engaged in a large-scale effort to determine what consensus has been reached about the nature of linguistic datanalyses, and to systematize that knowledge in a fully sharable Web **knowledge base** or **ontology** we call GOLD (General Ontology for Linguistic Description, announced in Farrar and Langendoen (2003); see http://emeld.org/ for the current publicly available version of GOLD). We are attempting to do so in such a way that it will be accessible to everyone who wants to use it for annotating their own datanalyses, for incorporating it into tools for creating, validating, and interpreting such markup, and for doing "smart" Web searches including searches for comparable datanalyses across languages. To the extent that GOLD succeeds in being comprehensive, it will suggest to users what to call things and how to relate them to each other, but without requiring the use of any particular technical vocabulary. It will recommend that users **link** their annotated concepts (whatever they choose to call them) to the concepts in GOLD, and in this way the ability to search meaningfully across different datanalysis sets will be greatly enhanced.

The present-day computational environment provides several choices as to how to proceed to make linguistic datanalyses available electronically, including the one just described. Another is for different individuals or groups to create potentially large, separate datanalysis ‘warehouses’ of material analyzed systematically in accordance with their particular encoding framework, with their own tools for access and searching. This by and large has been the path that the field of linguistics has been following. This approach has several advantages. Individuals or groups who have the necessary resources (e. g. a competent team of linguists and programmers with adequate funding) can get their datanalyses online and periodically updated, along with controls on how they may be accessed. As a result, large amounts of carefully analyzed data about particular languages, linguistic features, and theoretical approaches are available and accessible, particularly if the datanalysis providers expose their metadata to OLAC or general-purpose search engines.

One of the disadvantages of this approach is the possibility that the datanalyses become inaccessible after the funding runs out or the encoding framework is no longer supported. Another is that the burden of comparing datanalyses from different sources is shifted to the end users, who are probably not in a position to fully interpret what they have found. Yet another is that this approach does not help to build a world-wide community of linguists, indigenous language communities, language teachers and language learners in the enterprise of documenting the world’s languages, providing for their future survival, and stimulating the research community with vast amounts of *comparable* linguistic datanalyses.

In committing ourselves to the first approach, we do not denigrate the work of electronic datanalysis warehouse providers. The datanalyses available on certain of these sites constitute a kind of ‘gold standard’ against which other datanalyses may be compared. To do so, however, requires that we figure out a way to integrate their work with that of individual researchers or communities who either simply post their work on websites in HTML or PDF format, or who use the resources that the EMELD project or others provide to make their datanalyses “smart” and directly usable by others.

In order to create resources that stand a chance of actually doing the tasks we have set out for them, they have to be carefully designed from the outset, so that they can be scaled up to represent the full complexity of language data and of the analyses that linguists, with all their ingenuity and insight, have already come up with and can be expected to come up with in the future. First, we propose that the incorporation of **feature structure systems** along the lines originally proposed by the Text Encoding Initiative (Sperberg-McQueen and Burnard 1994/2002) and as described by Maxwell, Simons, and Hayashi (2002) into the overall markup schema will not only enable any linguistic analysis whatever to be represented, but will also facilitate the cross-comparison of typologically diverse datanalyses. Next, we show how this is made possible by GOLD. We further illustrate the kinds of automated reasoning tasks that may become available in the near future when linguists become accustomed to using emerging Semantic Web technology as suggested by Berners-Lee, Hendler and Lassila (2001) and Simons (2003a, b). Finally, we focus on how automated reasoning can facilitate both markup creation tools and smart search engines.

To show how all this will work, we must discuss some the formalisms and computational power needed for the implementation. Specifically we need to show how GOLD can implement feature structure systems, using the mechanisms provided by the Web Ontology Language (OWL, a proposed standard for representing ontologies on the Web; for an introduction, see http://www.w3.org/2001/sw/.).

2.0 Background and Previous Work

In this section, we provide an overview of the following topics:

- systems of feature structures as they relate to current markup technologies;
- the theory and use of ontologies; and
- the development of a particular kind of logic known as **description logic** (Baader et al. 2003) for reasoning over very large knowledge bases of linguistic data.

Readers already familiar with these topics or who prefer to skip the theoretical details can proceed to section 3.

2.1 Features Structures and Their Markup

Feature structure theory, originally developed within linguistics by Kaplan (1975), Kay (1979; 1984), and Shieber and Uszkoreit et al (1983), provides a very general and expressive framework for representing the results of linguistic analysis, from the analysis of a single morpheme to an entire grammar. Although its systematic use is confined to certain theories of grammar such as HPSG (Pollard and Sag 1994), most linguists today, including most field linguists, use it extensively, albeit informally, so that we feel that its use within the EMELD project is entirely suitable.

The fundamental components of the theory are **features**, how features combine into **structures**, and how these structures are related in **systems**.

A feature is an ordered pair (also known as an **attribute-value pair**) consisting of a **name** (or **attribute**) and a **value**. Two simple examples of linguistic features are <NUMBER, PLURAL> and <GENDER, FEMININE>. For each name, we also specify its range of possible values. For example, NUMBER can have the values PLURAL and SINGULAR, and GENDER the values FEMININE and MASCULINE. In this way, we disallow features like *<NUMBER, MASCULINE> and *<GENDER, SINGULAR>.

A feature structure is a set of features, for example {<NUMBER, PLURAL>, <GENDER, FEMININE>}. Usually, feature structures are written as two-column matrices, with the feature names in the first column and their values in the second, as in Figure 1.

$$\begin{bmatrix}\text{NUMBER} & \text{PLURAL}\\ \text{GENDER} & \text{FEMININE}\end{bmatrix}$$

Figure 1. A feature structure written as a matrix

A feature structure can consist of a single feature, e.g. {<NUMBER, PLURAL>} = [NUMBER PLURAL], or of none at all. The **empty** feature structure we write as [ ].

Feature structures are related by **subsumption**, whereby a more general feature structure subsumes more specific ones. For example, the structure [NUMBER PLURAL] subsumes the structure in Figure 1, and the structure [ ] subsumes them both. If we allow a feature structure to subsume itself, then the subsumption relation is reflexive, antisymmetric, and transitive, i.e. a partial ordering.

A feature structure system is a pair consisting of a set of feature structures and the subsumption relation, for example <{[NUMBER SINGULAR], [NUMBER PLURAL], [ ]}, $\leq$>, where $\leq$ represents subsumption. This structure may be graphed as in Figure 2, where the arcs, read downward, represent subsumption, and the reflexive arcs are omitted. Such graphs are invariably **directed acyclic graphs** (**DAGs**).

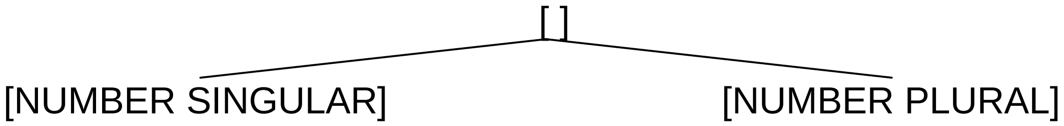

Figure 2. A feature structure system involving three feature structures

The system in Figure 2 is not a **complete** (or **boolean**) system since it lacks a ‘bottom’ (contradictory) feature structure $\perp$, which every feature structure in the system subsumes. Inserting $\perp$ in its proper place in Figure 2 along with the appropriate subsumption arcs, we observe that [ ] and $\perp$ are **negations** of each other, as are [NUMBER SINGULAR] and [NUMBER PLURAL]. We also observe that $\perp$ is the greatest lower bound (**conjunction**) of [NUMBER SINGULAR] with [NUMBER PLURAL], and that [ ] is the least upper bound (**disjunction**) of [NUMBER SINGULAR] with [NUMBER PLURAL]. In feature structure theory, the conjunction operation is given a special name, called **unification**. If the unification of two feature structures is $\perp$, then we say that their unification **fails**.

A feature structure system is complete if every structure in it has a negation, and every pair of structures in it has a unification and a disjunction. In a complete system, the structures that subsume only themselves and $\perp$ are the **generators** of that system, and if there are *n* generators, then the system contains 2^n distinct structures. For example, the system whose generators are the structures like that in Figure 1 with all possible values for NUMBER and GENDER, 4 altogether, contains 16 structures. In this system the unification of the structure [NUMBER PLURAL] with [GENDER FEMININE] is the structure in Figure 1, and in general if a feature structure consists of a set of features, it represents the unification of those features taken singly as feature structures. However, some of the structures in this system can only be expressed with the use of logical operators such as negation or disjunction, for example the structure [NUMBER

SINGULAR] **|** [GENDER MASCULINE], equivalently the negation of the structure in Figure 1. To accommodate such structures, the definition of feature structures must be generalized to allow for the use of these operators.

Feature structures are associated with linguistic **segments** (written expressions or utterances) of various sizes to form linguistic **units**. A set of units that express the members of a feature structure system constitute a **paradigm**. By their nature, paradigms are incomplete. For example a complete feature structure system generated by units with four case features, three gender features and two number features contains 2^{24} feature structures, but a typical paradigm using those features has far fewer members. Usually paradigmatic gaps are noticed only when an expected generator is missing, for example if there is no expression for a neuter dative plural.

When segments combine to form larger segments, their associated feature structures also combine. The default segment combinatory operation is **concatenation** and the default feature structure combinatory operation is unification. Both concatenation and unification may be overridden at least in part by other operations (deletion of segments and other 'morphophonemic' operations; replacement of one feature value by another, etc.). Legitimate combinations can generally be expressed using underspecified feature structures, where the features identified in those structures represent classes of linguistic units that can be combined. Since there are recursive combinatory operations in the grammars of natural languages, feature structures must be capable of being built up recursively. The mechanism for doing that is to permit feature structures or pointers to feature structures to occur as feature values.

In principle, it is possible to provide a full grammatical analysis of a language by specifying:

- what feature names it uses and what their associated values or value types are;
- what combinations of features are permitted in its feature structures (i.e., what feature structure systems it contains);
- how feature structures are associated with its elementary segments (i.e. how its lexicon is constructed);
- what its legitimate combinatory operations are.

The first set of recommendations for encoding feature structures in machine-readable documents was made by the TEI (Sperberg-McQueen and Burnard 1994); Langendoen and Simons (1995) provides a rationale for the specifics of those recommendations, which support all of the requirements just listed. These recommendations (which are included with little change in the 2002 XML version of the TEI recommendations) have not been as widely used as many of the other TEI proposals, in part because of their complexity, and in part because of the lack of tools to prepare and interpret documents using them. However, Black (1997) shows that the use of feature structures can ensure a high degree of systematicity in language description that aids in the construction of automated parsers and grammar checkers. Ide, Kilgarriff and Romary (2000) argue specifically for the use of feature structures in the design of machine-readable lexicons, and Maxwell (2002) and Maxwell, Simons and Hayashi (2002) propose a system for text glossing using feature structures.

One of our goals in the EMELD project is to streamline the TEI recommendations for encoding feature structure systems, to support the development of tools to create, validate, and interpret documents containing such encoding, and to integrate feature structures into GOLD.

2.2 Ontologies

Viewing linguistic datanalysis, for a moment, as a kind of knowledge, it has been recognized that ontologies are useful for many types of knowledge representation tasks. According to Gruber (1995: 908), an ontology is “an explicit specification of a conceptualization” where a conceptualization is the “set of objects, concepts, and other entities that are assumed to exist” in some domain, and according to Sowa (2000: xi), an “ontology defines the kinds of things that exist in the application domain”. Thus, an ontology makes explicit metaphysical commitments about what *is*, whence the term, which is based on Greek *einai* ‘to be’. An ontology is an engineering artifact, so one may speak of Aristotle’s or Peirce’s ontology. It should be noted that the term ‘ontology’ is often used to refer to simple taxonomies or to informally speak about the conceptual domain in a database system. We will avoid this usage and reserve the term ‘ontology’ to refer to a full formalization, in some logical language, of the allowable entities and relations in a domain.

Capitalizing on recent progress in the understanding of the role of ontologies in knowledge systems, we have created a General Ontology for Linguistic Description (GOLD) (see section 1). GOLD provides a semantic framework for the representation of all kinds of linguistic knowledge. For example, it makes explicit the possible relations between meaning and form and the relations between linguistic and world knowledge. The former is useful in describing and reasoning about paradigmatic information across languages or even within a particular language. For example, one characteristic of an inflectional paradigm is that the all the elements express a core meaning. The ontology can be used to ensure this by explicitly defining the notion of ‘core meaning’ and various types of inflectional affixes. The latter (the relationship between linguistic and world knowledge) is important for grounding linguistic data in the broader sense. For example, in defining noun classification systems, the ontology provides the means to reference general concepts such as ‘things made by hand’, ‘round objects’, etc. This allows the linguist to construct theoretically or culturally relevant classes based solely on data. The rationale behind using an ontology as part of a markup system is that it gives linguists the maximum amount of freedom in the description of data while allowing them to maintain compatibility with other data. The resulting system is above all useful for the automatic comparison of marked up language data.

In its current state, the GOLD ontology may be divided into at least four conceptual sub-domains: WrittenExpression/Utterance, LinguisticUnit, and SemanticUnit. These correspond respectively to the common notions of form, mental representation, and meaning. The classes WrittenExpression and Utterance are the physical realization of a LinguisticUnit, where a LinguisticUnit is a mental object associated with the elements of a language. Therefore, linguistic data, of the type collected by field linguists, consists of instances of WrittenExpression in the case of orthographic form or instances of Utterance in the case of recorded speech. These classes realize instances of LinguisticUnit, which may be further divided as in Figure 3.

- LinguisticUnit
  - SublexicalUnit (bound)
    - InflectionalUnit
    - DerivationalUnit
    - Clitic
    - BoundRoot
    - BoundStem
  - LexicalUnit (free)
    - FreeRoot
    - ComplexLexicalUnit
      - FreeStem
      - Compound
  - PhraseUnit
    - NounPhrase
    - VerbPhrase
    - Sentence

Figure 3. A partial taxonomy of LinguisticUnit

A LinguisticUnit may be thought of as the mental encoding of a morpheme or combination of morphemes. It should be noted that the taxonomy is incomplete and may be expanding by individual linguists working on a particular language. For example, the concept BoundRoot is the class of all bound roots found in all languages. For a particular language, e.g., Biao Min, there could be a subclass for all Biao Min bound roots, which may be defined and linked to GOLD. We understand that not every category in GOLD may be acceptable to all linguists. It is hoped, however, that individual linguists may use at least a portion of the ontology and/or define new categories where GOLD is deficient. A SemanticUnit is essentially the meaning of some linguistic unit. The availability of such a construct allows, for example, the comparability of content expressions across languages, and for the construction of such semantic categories as HopiGod vs. EnglishGod which serve to define the concept ‘god’ in these languages (or cultures).

It is not difficult in principle to include within GOLD the constructs needed for the description of feature structure systems. The notion of a feature is already partially built into ontologies; simple features (those that do not have feature structures or pointers to them as values) embody the relation between a class (a feature name, or attribute) and an instance of that class (one of its values). Within GOLD, the possible values of an attribute constitute (or will constitute, once it is built out) a superset of the set of possible values in a given language (more precisely, in particular analyses of that language). (We put aside here the possibility that there may be in GOLD intervening classes between a feature and its associated values in a particular language.) In designing a tool for linguists to use to 'select' the possible values for a feature name, we can provide a drop-down list of the entire superset, or perhaps more usefully, a default list, with the option of adding others as needed.

Since the subsumption relation has the logical properties of the converse of the **is a** relation of ontological theory, it is (or should be) a straightforward matter to incorporate feature structures into an ontology, essentially as the unification of the individual features that they comprise. If such a structure is a possible value for a feature, then it becomes in effect an instance of that feature in the ontology.

We leave the characterization of a feature structure system in an ontology to another occasion.

2.3 Description and Automated Reasoning

Ontologies are meant to be used within a broader system of knowledge, that is, a knowledge base, which is defined using a particular logical formalism. One such formalism that has gained in popularity among the knowledge engineering community is known as 'description logic' (see Baader et al. 2003). The term 'description logic' is used to refer to a family of logical formalisms which is designed for implementation, that is, designed for both expressiveness and tractability. Description logics are essentially a way to constrain the power of full first/second-order logic in order to arrive at acceptable levels of decidability.

The primitives in a description logic include 1) 'concepts' (the types), such as WORD or TENSE, and 2) 'individuals' (the tokens), such as the word *pig* or the grammatical category PASTTENSE, and 3) elementary 'roles' such as HASVALUE or DENOTES. Concepts may be thought of as unary predicates and roles as binary predicates. Unlike unrestricted first-order logic, Description Logic is variable-free. Therefore the concept HOPIWORD in a Description Logic would be equivalent to ∀x HopiWord(x) in first-order logic. There are several concept-forming operations in a description logic, but space precludes discussion here.

The use of Description Logic is advocated for the Semantic Web (Berners-Lee, Hendler and Lassila 2001), and its XML serialization is known as Web Ontology Language (OWL) (W3C 2002). OWL is a markup language for representing ontologies and knowledge bases for a variety of domains.

3. The Proposed Markup Framework

In this section we discuss the proposed markup framework which consists of three interrelated components: the feature system, the ontology, and the logical formalism. We first give a brief synopsis of how an ideal feature system would be constructed and then motivate the use of the an ontology as a part of the system. Finally with the semantically enriched system in place, we discuss various reasoning capabilities, including semantic search.

3.1 Feature System

The use of a feature system provides a powerful and expressive mechanism for describing language data and building tools for automated processing. The first step in constructing a feature system for a particular language is the enumeration of syntactic categories for the particular language. For example, language x has {*noun, determiner, verb*, *adjective*, …}. The second step is to enumerate the possible features for each syntactic category along with their possible values. For example, verbs in language x have *tense* {*pastTense*, *nonpastTense*} and *mood* {*indicative, optative*}.

Using the above system, it is possible to enrich language data on the Web. Complications arise, however, due to a number of factors. First, linguists use a variety of labels to describe the same concept (i.e., feature or value). When referring to the ‘perfective aspect’, one author may use *PF* while another may use *PERF*. These two elements may have exactly the same meaning, and even pertain to the same language data, but a simple search for one would fail to return examples of the other. Also, identical strings can mean different things. A search for *PA* intended to mean PARTITIVE could return documents and examples containing *PA* meaning PERFECTIVEASPECT. Second, feature systems and the available set of features must be broad enough to allow for all the variation in language. This requirement may be impossible to fulfill until all languages are described. However, it is possible to provide the mechanism for defining new features as detailed language description becomes more common. And the individual features of particular languages can be systematically compared. This is related to the problem of nuances of feature/value meaning. That is, there may be no universal notion of pastTense. Instead, pastTense in English is different from pastTense in Dyirbal. The problem, then, is to determine how like features and their values differ in the context of particular languages. Finally since feature systems are power tools for representing a variety of linguistic knowledge, a particular system should be cross-compatible with other levels of description. A feature system for phonology should be compatible with one for morphosyntax. That is, the same formalism should be usable for a variety of linguistic phenomena.

3.2 Incorporating the Feature System with GOLD

The key to solving all of the above problems is to provide a basis for defining the meaning of features/values as they are encountered in markup. We argue that the General Ontology for Linguistic Description (GOLD) is adequate for this task as it provides the semantic machinery for making the elements of a feature system explicit within the broader framework of knowledge. We have created with GOLD a relatively small set of well-defined, general linguistic concepts. The aim is to create tools with which linguists, in particular field linguists, can construct their own categories to suit a particular language.

First of all it is necessary to define the notion of a 'feature'. Ontologically a feature may be thought of as a defining quality relevant to some particular domain. Other names for features in general include quality, feature name, and attribute. In the broader domain, a feature may be thought of as a quality associated with some entity. So, instances of features include COLOR, SIZE, SHAPE, etc. (Shieber 1986: 12; Gärdenfors 2000; Masolo et al. 2002). There is an ontological separate of physical things (objects and processes), features, and abstract things as shown in Figure 4.

- Entity
  - Physical
  - Feature
  - Abstract

Figure 4. Upper taxonomy for the ontology

For linguistics we are concerned with the grammatical qualities of instances of LinguisticUnit, or those qualities which determine how instances of LinguisticUnit behave in the grammar of a language. Instances of MorphoSyntacticFeature include: tense, aspect, mood, number, person, partOfSpeech, etc. A MorphoSyntacticUnit is said to stand in a hasGramInfo relationship to particular instances of MorphoSyntacticFeature.

Features have as their values instances of the class FeatureValue. That is, specific features have specific values associated with them, e.g., the feature tense has as its possible values {past, present, ..., future}. Values can be thought of as points in a particular space or region, in philosophical terms feature values are known as ‘qualia’. Tense values are points in tense space which are represented in the ontology as the class TenseValue, where TenseValue defines all the allowable values for tense in human language (see Gärdenfors 2000 for a discussion of this notion). Subtypes in the value hierarchy do exist. Using TenseValue as an example, we may pose the following taxonomy, in Figure 5.

- MorphoSyntacticValue
  - TenseValue
    - AbsoluteTense
      - AbsolutePastTense
        - simplePastTense (instance)
        - recentPastTense (instance)
      - …
    - RelativeTense
      - …

Figure 5. The TenseValue taxonomy in GOLD

One thing should be clarified concerning the feature values of specific languages. In Yandruwandha (Dieric, Australian), there are five different instances of tense values which correspond to the English past tense: ‘very recent past’, ‘within the last couple of days’, ‘within the last few days’, ‘weeks or months ago’, and ‘distant past’ (Comrie 1985: 98). The instances in Figure 5 correspond to canonical values only. Individual researches will be able to specify instance for particular languages using the next generation of markup tools, e.g., FIELD, under development by the Michigan EMELD group.

The resulting picture of the relevant relations in ontology is given in Figure 6 as a conceptual graph, where the nodes are concepts and the arcs are relations between those concepts.

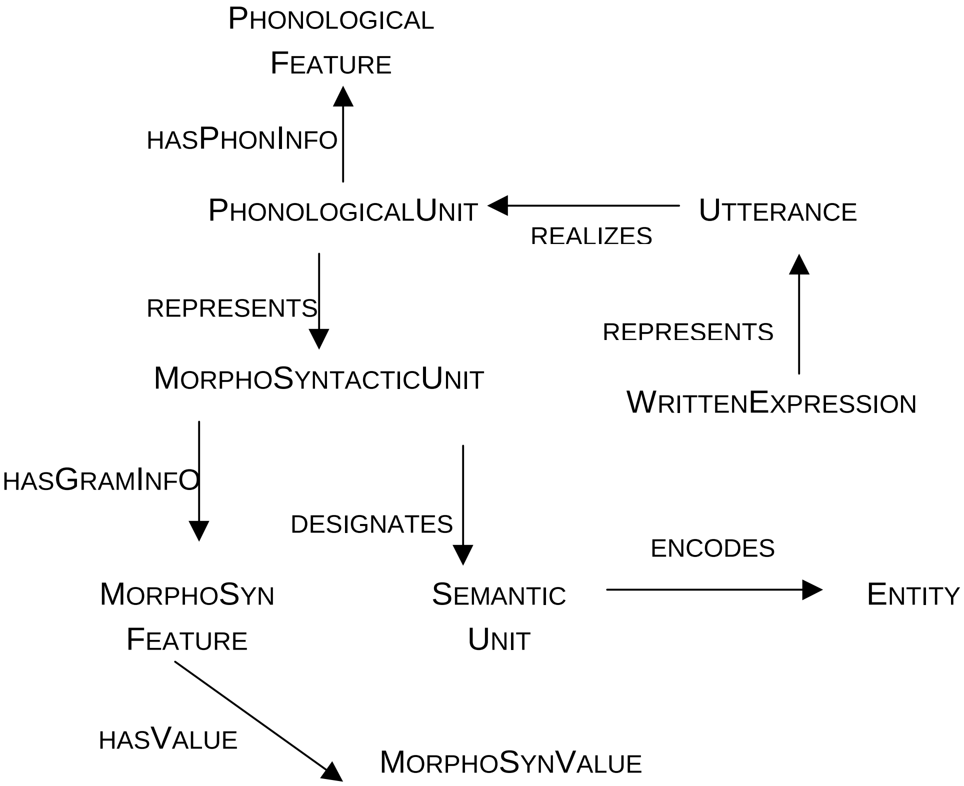

Figure 6. Relations among the various GOLD entities

The figure represents several relations and a full explanation can be found in the actual documentation of the ontology. What is relevant here are the relations between MorphoSyntacticUnit, MorphoSyntacticFeature, and MorphoSyntacticValue.

In section 2, it was noted that a feature structure system is a useful device for describing the grammatical properties of a language. With the notions of FEATURE and a VALUE in place, we include in the ontology the class FEATURESYSTEM and define it as the class of grammatical systems which uses features and values to represent grammatical information. For now it can be assumed that only one feature system exists per language data project (Maxwell, Simons and Hayashi 2002). A FEATURESYSTEM consists of a set of FEATUREVALUEASSOCIATION, which is a set of features and their allowable values, and a set of FEATURECONSTRAINTS, which dictate which features can co-occur in a feature structure. Keep in mind that a FEATUREVALUEASSOCIATION relates a particular feature of language with a subset of values from value space. That is, some language may only use two values from all possibilities in value space. At this point we have not worked out the details of FEATURECONSTRAINT. But it may be stated that certain features are constrained to co-occur with certain instance of the feature PARTOFSPEECH. For example, HOPINOUN and PASTTENSE may never be found on the same HOPIMORPHSYNUNIT. A pairing of a feature with a value is a FEATURESPECIFICATION. And a set of FEATURESPECIFICATIONS is a FEATURESTRUCTURE. In this respect and most others, our model is virtually identical to that of Maxwell, Simons and Hayashi (2002).

3.2 Utility of the system

One of the most useful aspects of having enriched data on the Web is the ability to execute ‘semantic searches’. Semantic search is defined as the matching of the intended meaning of the search string(s) with the meaning of the data. For example, a semantic search for PROGRESSIVE returns only data that actually **means** PROGRESSIVE. The data may not contain any literal strings such as *prog* or *progressive,* but, if the source document contains pointers to the corresponding ontological concept, then a search will produce a hit. Another useful kind of purely semantic search, particularly for typological research, is to ask how specific languages grammaticalize/lexicalize certain concepts. For example, a search could be carried out for all examples of data that encode some kind of SPATIALRELATION. Ideally, this would return data from languages which employ adpositions, such as German and French, and data from languages which employ SPATIALCASE such as Archi (Dagestan) and Hungarian. Another kind of search pertains to how grammatical categories in certain languages seem to form subclasses of those in another language. That is, pure semantic search would be useful in surveying cross-linguistically the potential granularity of a certain class, e.g., the allowable TENSEVALUES in Hopi versus those in Navajo. Also, a query could be constructed to search for portmanteau expressions, that is where one form designates two or more grammatical categories. To execute such a search, it would be necessary to search for instances of LINGUISTICUNIT which carry two or more instances of FEATURE. The type of FEATURE could also be specified. If properly marked up, a language's status as either agglutinating, isolating, etc. could be determined automatically, that is, if rules for defining each of these language types were given.

Other than search we propose that the ontology can be used to enhance markup tools, such as FIELD. Key to such a tool is the ability to guide the user in selecting the correct categories for a language. For example, it would be inappropriate and tedious to sort through all available feature values {perfective, past, future, …, indicative} when the only the values for tense are relevant. Furthermore, the ontology allows one to define a canonical feature system. That is, FIELD first presents the user with the most common, canonical categories for the user to choose from. Another kind of tool under development seeks to the semi-automate the markup process of existing (legacy) data (Lewis 2003). The ontology will be utilized to make predictions about the meaning of tags in, for example, interlinear text, provided that at least something is known about the existing markup. The widespread use of informal standards (passed on from generation to generation) is indeed the case, as Lewis shows in mining of existing legacy data.

A rich markup system using the semantic power of an ontology allows for many unprecedented capabilities. Certain possibilities which are at least conceivable, however, are not yet realizable due to constraints on reasoning power and the status of our understanding of linguistics in general. For example, an expert system for linguistics is not currently possible. That is, it is not possible to simple feed a computer program field data and have that data automatically analyzed. Implemented systems such as FIELD will still require a high degree of supervision. As more and more knowledge is encoded in the ontology, however, less and less human supervision will be required provided appropriate reasoning systems are employed. One characteristic of FIELD is its ability to give grammatical predications based on a preliminary sketch of the data. For example, given a part of speech inventory, the program might be able to predict which grammatical categories are possible in the language. Due to the current broad-brush approach in the ontology, fine-grained grammatical (syntax/semantic/phonological) comparisons are still in the future. But as we have argued, even course-grained comparisons will be immensely useful to researchers. Finally, the ontology represents one of the few efforts towards universal semantics applicable to all languages. There is an enormous task ahead of linguistics and notable efforts toward a solution include Peterson (2000) and Bluhme, Nickels and Zaefferer (2003).

4. Conclusions

The recent advances in Web technology combined with the development of a variety of markup languages have allowed for unprecedented research capabilities for linguists. The dream of an expert system, a kind of omniscient artificial linguist, is perhaps only science fiction. However, taking advantage of the current technology can at least make the rudiments of such a system possible. One requirement, to be sure, is the consistent use of markup recommendations promulgated by such groups as EMELD, DOBES, AILLA, and SIL. Going a step further, it is possible through the create of ontologies like GOLD to merge the efforts of these organizations to achieve data compatibility across the Web. We are at a crucial junction in this endeavor as the next task is to develop tools to “make it all happen”. That is, it is the responsibility of individual communities (like field linguistics) to develop and maintain Web archives and their own Web tools to realize the benefit of the Semantic Web.

References

Baader, F., D. Calvanese, D. McGuinness, D. Nardi and P. Patel-Schneider (2003) *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge: Cambridge University Press.

Berners-Lee, T., J. Hendler and O. Lassila (2001) The Semantic Web. *Scientific American,* May 2001.

Black, C. (1997) A PC-PATR implementation of GB syntax. *SIL Electronic Working Papers*.

Bluhme, S., M. Nickels and D. Zaefferer (2003) Cross-linguistic reference grammar: An XML-based internet database for general comparative linguistics. Presented at *DGfS-CL-2003*, 27 February 2003, Munich.

Comrie, B. (1985) *Tense*. Cambridge: Cambridge University Press.

Farrar, S. and D. T. Langendoen (2003) A linguistic ontology for the Semantic Web. *GLOT International* 7 (3), 97-100.

Gärdenfors, P. (2000) *Conceptual Spaces: The Geometry of Thought*. Cambridge, MA: The MIT Press.

Gazdar, G. and C. Mellish (1989) *Natural Language Processing in LISP: An Introduction to Computational Linguistics*. Menlo Park, CA, Addison-Wesley.

Gruber T. R. (1995) Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies 43*, 907-928.

Ide, N., A. Kilgarriff and L. Romary (2000) A formal model of dictionary structure and content. *Proceedings of Euralex 2000*, 113-126, Stuttgart.

Kaplan, R. (1975). On process models for sentence comprehension. *Explorations in Cognition*, ed. by D. Norman and D. Rumelhart. San Francisco: W. H. Freeman.

Kay, M. (1979) Functional grammar. *Fifth Annual Meeting of the Berkeley Linguistics Society*.

Kay, M. (1984) Functional unification grammar: A formalism for machine translation. *10th Annual International Conference on Computational Linguistics* (COLING-84), Stanford.

Langendoen D. T. and G. Simons (1995) A rationale for the TEI recommendations for feature-structure markup. *Computers and the Humanities* 29,191-209.

Lewis, W. D. (2003) Mining and migrating interlinear glossed text. Paper presented at this workshop.

Masolo, C., S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, and L. Schneider (2002) WonderWeb deliverable D17 (version 2.0). The WonderWeb Library of Foundational Ontologies and the DOLCE ontology.

Maxwell, M. Resources for morphology learning and evaluation. *LREC 2002: Third International Conference on Language Resources and Evaluation* vol. III, 967-974. Las Palmas.

Maxwell, M., Simons, G. and Hayashi, L. (2002) A morphological glossing assistant. Paper presented at the LREC Workshop on Resources and Tools in Field Linguistics, Las Palmas.

Peterson, J. (2000) *Cross-Reference Grammar Project* 2.0. Ludwig-Maximilians-Universität zu München.

Pollard, C. and I. A. Sag (1994). *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press.

Shieber, S. M. and H. Uszkoreit, et al. *The Formalism and Implementation of PATR-II*. Menlo Park: SRI International.

Shieber, S. M. (1986) *An Introduction to Unification-Based Approaches to Grammar*. Chicago: University of Chicago Press.

Simons, G. (2003a) A metaschema language for the semantic interpretation of XML markup in documents. OLAC draft recommendation. <http://www.language-archives.org/REC/metaschema.html>

Simons, G. (2003b) Developing markup metaschemas to support interoperation among resources with different markup schemas. Presented at the ACH/ALLC Joint Conference, 29 May - 2 June, Athens, GA.

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

Sperberg-McQueen, C. M. and L. Burnard, eds. (1994/2002). *TEI P3/ TEI P4: Guidelines for Electronic Text Encoding and Interchange*. P3 (SGML version), Oxford and Chicago: ACL/ACH/ ALLC; P4 (revised XML version), Oxford, Providence, Charlottesville, and Bergen: Text Encoding Initiative Consortium.