Knowledge Structures for Natural Language Generation [1]

Paul S. Jacobs
Knowledge-Based Systems Branch
General Electric Corporate Research and Development
Schenectady, NY 12301 USA

Abstract

The development of natural language interfaces to Artificial Intelligence systems is dependent on the representation of knowledge. A major impediment to building such systems has been the difficulty in adding sufficient linguistic and conceptual knowledge to extend and adapt their capabilities. This difficulty has been apparent in systems which perform the task of language production, i. e. the generation of natural language output to satisfy the communicative requirements of a system.

The *Ace* framework applies knowledge representation fundamentals to the task of encoding knowledge about language. Within this framework, linguistic and conceptual knowledge are organized into hierarchies, and *structured associations* are used to join knowledge structures that are metaphorically or referentially related. These structured associations permit specialized linguistic knowledge to derive partially from more abstract knowledge, facilitating the use of abstractions in generating specialized phrases. This organization, used by a generator called *KING* (Knowledge INtensive Generator), promotes the extensibility and adaptability of the generation system.

1 Introduction

The task of *natural language generation* is that of producing linguistic output to satisfy the communicative requirements of a computer system. The principal limitation of existing programs which perform this function is that they fail to realize a sufficiently broad range of requirements to demonstrate a convincing linguistic capability. This drawback seems founded in aspects of the systems which hinder the development of a large base of knowledge about language. A great deal of knowledge is required to produce any given utterance, yet much of this knowledge cannot easily be exploited across a range of utterances.

Partial success in generation systems is often achieved by applying linguistic knowledge to particular domains. Exemplary of this success are text generation programs such as PROTEUS [6], and Ana [15,14]; as well as generation components of on-line systems; for example, in HAM-ANS [5], UC [9,11], and VIE-LANG [4]. These systems, while embodying a variety of generation techniques, serve to illustrate the importance of the command of specialized constructs and the ability to utilize specialized knowledge in generation. A close examination of the knowledge used in such programs, however, reveals that a great deal of linguistic information seems to be encoded redundantly, thus impeding the use of generalizations in "scaling up" the systems.

The UNIX [2] Consultant system [22] is a program which answers questions from naive users about the UNIX operating system. Scaling up the user interface required a generator which could produce responses such as the following:

1. 'Chmod' can be used to *give* you write permission.
2. You don't *have* write permission on the directory.
3. You can't *get* write permission on the directory.
4. You *need* ethernet access.
5. You don't *have* ethernet access.

The PHRED generator initially used by UC [11] produced output such as the above by treating each verb use as an independent specialized construct. This allowed no benefit to the system of abstract knowledge about the use of the verbs, nor of applying its knowledge about one specialized construct to another. This difficulty proves to be a major handicap in building large-scale generation systems: A key element is to facilitate the exploitation of generalizations while still providing for specialized uses. In order for the UC system to have this capacity, the linguistic knowledge representation used had to be redesigned.

This knowledge-based approach has led to the design and implementation of the *Ace* knowledge representation framework [12]. Ace is a *uniform, hierarchical* representation system, which facilitates the use of abstractions in the encoding of specialized knowledge as well as the representation of referential and metaphorical relationships among concepts. A general-purpose natural language generator, KING (Knowledge INtensive Generator)[10], has been implemented to apply knowledge in the Ace form. The generator works by applying structured associations, or mappings, from conceptual to linguistic structures, and combining these structures into grammatical utterances. This has proven to be a simple but powerful mechanism, easy to adapt and extend, and has provided strong support for the use of Ace knowledge structures in generation.

While this presentation describes the Ace knowledge structures from the point of view of language production, the representation framework is designed to be unbiased with respect to language analysis or generation.

The discussion which follows focuses on the representation of linguistic and conceptual knowledge in Ace, using as an example knowledge about the verbs "give", "take", "buy" and "sell". The examples show briefly how information is encoded in Ace which enables the generator to produce dative constructs such as "John sold Mary a book" and specialized forms such as "John gave Mary a kiss", making use of abstract knowledge about events such as *giving*. These verbs provide a good testing ground for a representational framework, as they may be characterized by certain linguistic generalizations while appearing in a variety of specialized constructs. For further examples and a description of the generation algorithm used by KING, the reader is referred to [10].

2 Ace Fundamentals

I have suggested that the development of extensible and adaptable natural language systems depends on a knowledge representation framework within which generalizations are effectively exploited. This is the primary goal of the Ace framework. The starting point of Ace was an implementation of a knowledge representation called KODIAK [21], which was extended to include explicit structured relationships between language and meaning. This section presents the basic knowledge representation principles behind Ace, and provides an example of how conceptual knowledge is used to relate knowledge about *selling* to knowledge about *giving*.

2.1 Basic Principles

Many knowledge representation systems, however different they appear superficially, may be shown to have the same

[1] This paper is based on the thesis research conducted while the author was at the University of California, Berkeley. The research was supported in part by the Office of Naval Research under contract N00014-80-C-0732, the National Science Foundation under grants IST-8007045 and IST-8208602, and the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3041, Monitored by the Naval Electronic Systems Command under contract N00039-82-C-0235.

[2] UNIX is a trademark of AT & T Bell Laboratories

formal expressive or inferential power. This discussion centers not on the question of formal power but on the *nature* of the knowledge which must be expressed. The Ace knowledge representation provides a framework for expressing essential linguistic knowledge in a form suitable for encoding within a representational formalism.

The following principles guide the encoding of knowledge important in the generation task:

- *Principle 1. Inheritance of Conceptual Relations.* Concepts in memory are organized into a hierarchy of categories, in which more specific concepts inherit "features" from more general concepts. This inheritance is a representational tool which has been employed throughout the history of Artificial Intelligence (*cf.*[18,19,2,3]). The question of what exactly is inherited, however, can be answered in a variety of ways. Ace takes advantage of *structured inheritance,* (*cf.* [3]), in which concepts linked to a particular structure may inherit from supercategories of that structure. For example, knowledge about the *seller* of a *selling* action may be inherited from knowledge about the *giver* of a *giving* action.
- *Principle 2. Proliferation of Conceptual Categories.* Individual concepts are themselves categories, and any concept about which there is particular knowledge is considered to form a category. Thus categories proliferate: Probably, there are far more conceptual categories than there are lexical items in the system. For example, it will be shown later in this section that it is reasonable to postulate a concept specifically for the action of paying money in exchange for merchandise, although there is no lexical item corresponding to this concept. The lexical term "pay" is associated with a more general concept, that of providing money in exchange for virtually anything. The lexical term "give" may be associated with a general *giving* concept, but giving to charity, giving an idea, and giving a chance are distinct concepts with distinct linguistic manifestations. For example, the use of the verb "give" without object or indirect object as in "Bill gave" and "I gave at the office" is a linguistic phenomenon which appears almost exclusively when referring to charitable giving.
- *Principle 3. Explicit Referential Relationships.* There are a range of conceptual relationships important in language use which are not easily described as factual or ontological relationships. The one which is considered here is the *view* relationship, which helps to determine how concepts may be used in expressing other concepts. The concepts of *giving* and *taking* may be related to the concept of a *transfer-event*, but the instantiation of the abstract *giving* and *taking* concepts cannot be factually inferred from the instantiation of *transfer-event.* For example, "John gave five dollars to charity" does not imply that a charitable organization *took* the five dollars from John. "Mary took the money from John" does not imply that John *gave* Mary the money. In many circumstances, however, the same event may be described using "give" or "take". For example, (1) "John gave Mary five dollars for the book" may imply (2) "Mary took five dollars from John for the book". Representing *giving* and *taking* as *views* of *transfer-event* permits the encoding of knowledge about describing *transfer-events* without requiring a given event to be classified as *giving* or *taking.* These views may thus represent the knowledge that "John took <x> from Mary" and "Mary gave John <x>" *might* be used to describe the same event. Such views will be shown to be useful in determining how linguistic structures are used to refer to events.

The next section describes the basic elements of the hierarchical framework of Ace.

2.2 Structured Associations in Ace

Ace makes use of a notation in which there are two types of entities: *objects* and *structured associations.*[3] A structured association is a relation among two or more objects which also relates corresponding objects associated with the related objects.

The most common structured associations in Ace, taken from the KODIAK representation, [4] are the *DOMINATE*, or "D", relation, which associates a subcategory with its parent category, and the MANIFEST or "m" relation, which associates a category with an *aspectual* or role. This hierarchical system is analogous to isa-links and slots in other similar representation systems; the motivation behind KODIAK was to preserve the ideas behind frame-based representations while clarifying the semantics of a "slot". For a comparison of KODIAK with other research, see Wilensky (forthcoming).

The term ROLE-PLAY, also taken from KODIAK, is used to indicate corresponding concepts across structured associations. For example, the assertion,

(DOMINATE *action selling*
with (ROLE-PLAY *actor seller*))

indicates that *selling* is a subcategory of *action,* with *seller* playing the role of *actor.* A graphical representation of this relation, with the ROLE-PLAY implicit, is illustrated in figure 1.

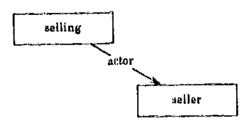

Figure 1: The *selling* action

Structured associations with ROLE-PLAYs are the basic mechanism for organizing knowledge in Ace. The next section describes how these associations are used to represent basic knowledge about *buying* and *selling.*

2.3 The Commercial Transaction Example

I have proposed that generalizing about linguistic constructs such as the dative form in "John gave Mary a dollar", "John told Mary a story", and "John sold Mary a book", seems to depend on the representation of concepts such as *giving, telling,* and *selling.* This section presents the foundation for the encoding of *selling* in Ace.

Consider the concept of the *commercial transaction* [7]. The *commercial-transaction* represents an event in which a merchandise object is exchanged for legal tender. The essential knowledge about this event may be represented by classifying the *commercial-transaction* as a *complex-event,* composed of at least two simpler events, *ct-merchandise-transfer* and *ct-tender-transfer.* Each of these two sub-events is a kind of *transfer-event,* and is thus used to associate roles of the *commercial-transaction* with roles of *transfer-event.* This knowledge is captured in figure 2.

Figure 2 illustrates the important knowledge that the *merchant* receives the *tender* from the *customer,* and the *customer* receives the *merchandise* from the merchant. Concepts such as *merchant, customer, merchandise,* and *tender* are aspectuals of the *commercial-transaction*; that is, they

[3] This term, and the idea of using general structured associations as a language processing tool, are due to Wilensky.

[4] These associations, as well as many of the ideas here, have evolved during a series of seminars among the Berkeley Artificial Intelligence Research group, led by Robert Wilensky. Other participants in these discussions were: Richard Alterman, Margaret Butler, David Chin, Charley Cox, Marc Luria, Anthony Maida, James Martin, James Mayfield, Peter Norvig, Lisa Rau, and Nigel Ward.

are specific concepts whose meaning is undetachable from the *commercial-transaction* event. However, much of the knowledge about these concepts, such as the *recipient* and *source* roles, is inherited from other concepts. As in other frame-like systems [19,2], this organization allows roles of a concept to be inherited in this manner. The ROLE-PLAY relationship in Ace, however, permits more than this simple form of inheritance: It allows for the semantics of aspectuals to be defined in terms of other aspectuals. For example, the meaning of the *merchant* aspectual of the *commercial-transaction* here is represented in part by the ROLE-PLAY relation which links this aspectual to the *source* of the *ct-merchandise-transfer* and that which links it to the *recipient* of the *ct-tender-transfer*.

The assertions above form an important core of knowledge about *commercial-transactions*. This knowledge is important

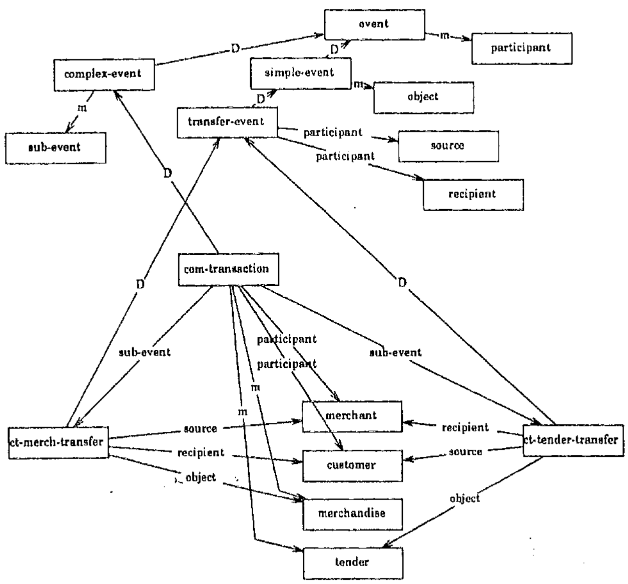

Figure 2: The *commercial-transaction* event

in the way language is used to describe such events. For example, it will be shown in section 4 that the knowledge that *merchandise* and *tender* play *object* roles is linked to knowledge about transitive verb forms, so that phrases such as "bought a book" and "paid five dollars" conform to a general rule.

The next section discusses how concepts such as *buying* and *selling*, used to refer to the *commercial-transaction* concept, are represented in Ace.

2.4 Actions as VIEWs of Events

I have proposed that verbs such as "give" and "take" refer to the actions *giving* and *taking*, and thus refer indirectly to the general *transfer-event* concept. The motivation for this analysis is to facilitate the representation of knowledge about the roles which *giver* and *taker* play, thereby enabling "John gave Mary a book" and "Mary took a book from John" to describe indirectly the same event, as "Ali gave Frazier a punch", and "Frazier took a punch from Ali" may indirectly describe the same event.

The *commercial-transaction* event is generally described using the verbs "buy", "sell", and "pay". "Sell" and "pay" behave similarly to the verb "give"; "buy", behaves more like "take". For example, "John sold Mary a book", and "Mary paid five dollars for the book" both use the dative form, and "John bought the book from Mary" exhibits a structure identical to "John took the book from Mary". The representation of the concepts *buying* and *selling* in Ace relates these concepts to *giving* and *taking* so that knowledge about expressing *giving* and *taking* may be used also for *buying* and *selling*.

The concepts *giving* and *taking* in Ace are related to the *transfer-event* concept by a structured association called a *VIEW*. Inspired by the notion of a view in earlier knowledge representations [17,2], the Ace VIEW is applied to metaphorical and analogical relationships, similar to those described in [16,8]. VIEWs are used to represent knowledge about concepts which may be used in expressing other concepts. The Ace network in figure 3 represents the basic knowledge about *giving, taking, buying*, and *selling*. In this hierarchy, the structured association *view1* between *transfer-event* and *giving* DOMINATEs the structured association *view3* between *ct-merchandise-transfer* and *selling*, and *view2* between *transfer event* and *taking* DOMINATES *view4* between *ct-merchandise-transfer* and *buying*.

The representation in figure 3 demonstrates on a small scale how the hierarchical arrangement of VIEWs is used in the encoding of structured associations. Structured associations such as *view1* between *transfer-event* and *giving* DOMINATE other more specific relations, such as *view3*. Note that this makes the explicit representation of ROLE-PLAY relations for *view3* unnecessary, as the relationship between *merchant* and *seller* in *view3* is specified by the relationship between *source* and *giver* in *view1*.

The representation of the *selling* concept is a simple example of how Ace encodes abstractions which may be used in language processing. The abstraction here is the relationship between a general category *giving* and a general category *transfer-event*. There are two ways in which this abstraction may be used: (1) A more specific association may be represented as a subcategory of the abstract association. This is the case in the *selling* example presented here. In this case, knowledge about the abstract association may be used in applying the specific association, thus knowledge about expressing an abstract concept may be used in expressing a more specific concept. This allows much of the same knowledge to be used for phrases involving "giving" and "selling". (2) A concept which is associated by another VIEW with the abstract concept may then also be expressed using the abstract VIEW. This is the case with expressions such as "give a punch" (cf. section 4.3), which takes advantage of the abstract *action as transfer-event* view in combination with the *transfer-event as giving* VIEW.

The next two sections discuss the application of the Ace framework to the representation of linguistic knowledge and to the representation of the knowledge which associates linguistic and conceptual structures.

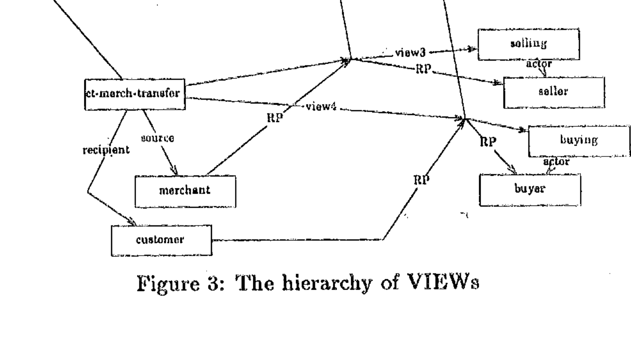

Figure 3: The hierarchy of VIEWs

3 Basic Grammatical Knowledge

The principles outlined in section 2.1 motivate a representation in which knowledge is dispersed throughout a hierarchy, with a greater number of structures, each containing more limited information. Linguistic knowledge in Ace is organized into a hierarchy which incorporates this type of organization. In fact, the same knowledge representation language is used to encode both linguistic and conceptual knowledge in Ace. The linguistic hierarchy provides for a simple set of basic linguistic templates, with little redundancy and relative ease of extension. The following principles of linguistic representation are suggested:

- *Principle 4. Inheritance of Linguistic Features.* Sets of features which are common to a certain class of templates need not be specified independently for each template in the class. Thus, if there is a set of features shared among passive sentences, or among prepositional phrases, these features belong by default to any template in the class. This eliminates the need for fully specifying the structure of phrases which have specialized properties or meaning.

- *Principle 5. Proliferation of Linguistic Categories.* In order to take advantage of the inheritance of features, there must be a wide range of classes of linguistic templates which share sets of features. Often these categories depart from the traditional syntactic classifications. Requiring that a template be a member of a unique category in the case of a gerund or nominalization can prove difficult, as these may inherit certain attributes from verbs and certain attributes from nouns. Thus any template may inherit features, including structural descriptions, from multiple categories. Categories are arranged hierarchically, so each category inherits from all its ancestors in the hierarchy.

- *Principle 6. Distinguishing Grammatical Relations from Grammatical Patterns.* A great deal of linguistic information seems associated with structural relationships between linguistic constituents which are dependent neither on their order in a surface structure nor on the precise nature of the structure in which they appear. For example, the relation between subject and verb retains its linguistic features regardless of how the subject and verb appear in any surface structure: The agreement between subject and verb in "John was given the book by Mary" is the same as in "Was John given the book by Mary?" as is the conceptual ***recipient*** role which John plays. Such information does not pertain to a particular surface structure, but to any surface structure in which a noun phrase and verb are in the ***subject-verb*** relation. In "John kissed Mary on the cheek" and "The kiss on the cheek pleased Mary", the role of "on the cheek" as it relates to "kiss" is independent of whether the prepositional phrase is part of a verb phrase or noun phrase. In general, structural linguistic relationships are not limited to those which are directly linked to constituent order, and thus a more general facility than a syntactic pattern is required to represent these relationships.

- *Principle 7. Uniformity of Representation.* Linguistic knowledge *is* knowledge, and thus can be encoded using the same representational framework as conceptual knowledge. The same structured associations used in the conceptual hierarchy can be used in the linguistic hierarchy. [5] Having such uniformity of representation has the practical value of facilitating the interaction of conceptual and linguistic structures.

[5]The use of a knowledge representation language to encode syntactic knowledge for the purpose of semantic interpretation has been practiced with KL-ONE [20] and its successors, also favoring uniformity of representation. Such systems have not been used, to my knowledge, to encode associations between conceptual and linguistic knowledge.

The discussion which follows demonstrates how linguistic knowledge may be encoded using the framework described in the previous section, and discusses the effect of property inheritance on linguistic knowledge representation.

3.1 Multiple Inheritance in the Ace Linguistic Hierarchy

This section shows how the structured associations of Ace, and particularly the capacity for multiple inheritance, are used to encode some of the linguistic knowledge used in the construction of simple sentences.

3.1.1 Verb phrases in Ace

Verb phrases provide a good example of the use of a linguistic hierarchy because they exhibit a variety of surface forms while obeying certain regularities. One type of verb phrase is the ***dative-vp***, a ***verb-phrase*** made up of a constituent ***dvp-indir***, which includes the verb and noun phrase corresponding to the indirect object, followed by the noun phrase corresponding to the direct object. The treatment of the ***dvp-indir*** as a separate constituent is done to facilitate the handling of other dative forms. Figure 4 shows how knowledge about the dative verb phrase pattern is encoded in Ace, as well as how this ***dative-vp*** is positioned in the verb phrase hierarchy. The pattern ***dvp-pattern*** represents the ordering of the constituents of the verb phrase, while the relation ***dvp-indir*** is used to represent the relationship between the dative verb and indirect object. Section 4 will show how this relation is associated with conceptual knowledge.

By allowing aspectuals which represent patterns and pattern constituents to play multiple roles, the representation of linguistic knowledge as shown in figure 4 shows how knowledge about infinitives, gerunds, and verb phrases is distributed in the Ace hierarchy. The various verb phrase patterns fall beneath the ***verb-phrase*** template, as do the gerund phrase, infinitive phrase, and finite verb phrase nodes. In order to

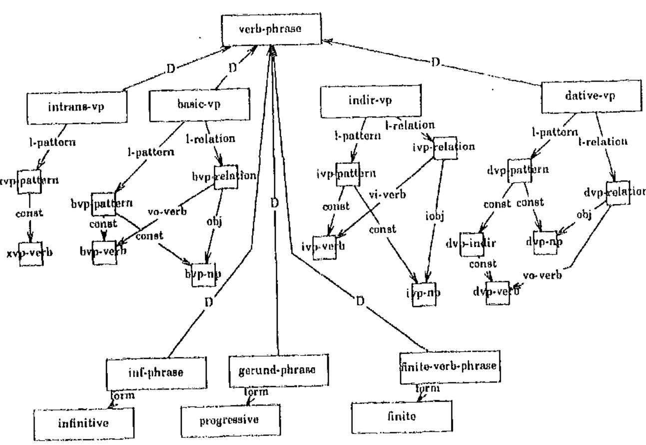

Figure 4: The verb phrase hierarchy

produce a finite verb phrase, a node lower in the hierarchy must be instantiated. This hierarchical organization permits the gerund phrase and infinitive phrase to have the same linguistic structure as the verb phrase, modulo the form of the verb part. An instantiated verb-phrase in Ace thus inherits most of its internal structure from one category, for example ***dative-vp***, and its external behavior from another, for example, ***finite-verb-phrase***. The ***verb-phrase*** category itself plays no external syntactic role—there is no pattern in which a constituent belongs to the ***verb-phrase*** category and to no lower category, although theoretically there could be. The effective organization of information about verb phrases stems directly from the application of the basic knowledge representation principles of Ace to linguistic knowledge.

This section has shown how a hierarchical representation is used to encode linguistic relations. The next section concentrates on how the representation of explicit referential relationships links this linguistic knowledge to conceptual knowledge.

4 Associating Language and Meaning

Section 2.1 presented the foundations of the Ace framework and its application to meaning representation, proposing that concepts be hierarchically organized, linked together via structured associations. Section 3 described how linguistic knowledge can also be hierarchically organized within this framework, and how features can thus be inherited through linguistic categories. The idea of a linguistic relation was proposed, to distinguish structural relationships from ordering relationships in grammatical constructs. Naturally, the knowledge required to produce correct and appropriate utterances includes both of these classes in addition to the links which bind the classes together.

4.1 Basic Principles

The goal of taking advantage of explicit referential knowledge, discussed in section 2.1, along with the framework presented in the previous sections, suggests two principles directed towards the association of linguistic and conceptual knowledge:

- *Representation Principle 8. Correspondence of Linguistic and Conceptual Structures.*
  Linguistic structures, such as lexical terms, linguistic categories, and grammatical structures, are directly associated with conceptual categories. General linguistic categories in Ace, such as ***verb-indir-relation***, correspond to general conceptual categories, such as ***transfer-event***. Specific lexical terms, such as "buy" and "sell", are linked to more specific concepts, such as ***buying*** and ***selling***. Grammatical structures, such as the ***modifier-noun*** relation, are associated with conceptual relations, such as MANIFEST.
- *Representation Principle 9. Association of Linguistic Features with Conceptual Attributes.*
  The structured association permits the relation of linguistic aspectuals with conceptual aspectuals via ROLE-PLAY. Linguistic features, such as ***dvp-pattern*** and ***ivp-verb*** as described in section 3 are represented as aspectuals in Ace, as their status depends on the template of which they are a part. The association of these features with conceptual attributes goes along with the association of the template with a concept. For example, the indirect object feature ***iobj*** is an aspectual of ***verb-indir-relation***. The linguistic feature ***iobj*** is linked to the conceptual attribute ***recipient***, an aspectual of the ***transfer-event*** concept. The direct-object feature ***obj***, an aspectual of ***verb-obj-relation***, is linked to the conceptual ***object*** concept, an aspectual of the ***simple-event*** concept. Thus linguistic features have conceptual correlates.

The next section describes how these principles are realized in the association of language and meaning in Ace.

4.2 Linking Linguistic and Conceptual Structures Using REF

The main tool for representing relationships between linguistic structures and conceptual structures in Ace is a structured association called REF. The REF association, designating "referential" relationships, is similar to VIEW, except that it joins language to concepts instead of concepts to concepts.

The linguistic knowledge presented earlier has included information about how verbs and their objects or indirect objects may appear in surface structure. The knowledge essential to build these structures, however, is contained in the correspondence between linguistic relations and conceptual entities. This information is presented in figure 5.

The diagrams in figure 5 illustrate how the syntactic structure of the dative verb phrase is associated with its conceptual structure. The verb part and noun phrase in the ***ivp-pattern*** in the top diagram belong to the ***verb-indir-relation***, which associates with the indirect object the concept of ***recipient***. This diagram presents a slightly abbreviated version of the Ace representation given in section 3, as ***ivp-verb*** and ***ivp-np*** are associated only indirectly with the ***verb-indir-relation*** via the ***ivp-relation*** aspectual.

The structured association between the ***verb-obj-relation*** and the ***simple-event*** concept in the lower diagram of Figure 5 links the object of the verb in the dative verb phrase to the object of the simple event. In this association, as in the association of the ***verb-indir-relation*** with the ***transfer-event***, the verb part ***dvp-verb*** is associated via ROLE-PLAY with the event itself, rather than with any aspectual of the event.

Like the pattern-concept pair [**11**] or the unification grammar template [13,1], REF is a means of associating linguistic structure with conceptual structure. The template, however, is replaced with an explicit structural link in the knowledge

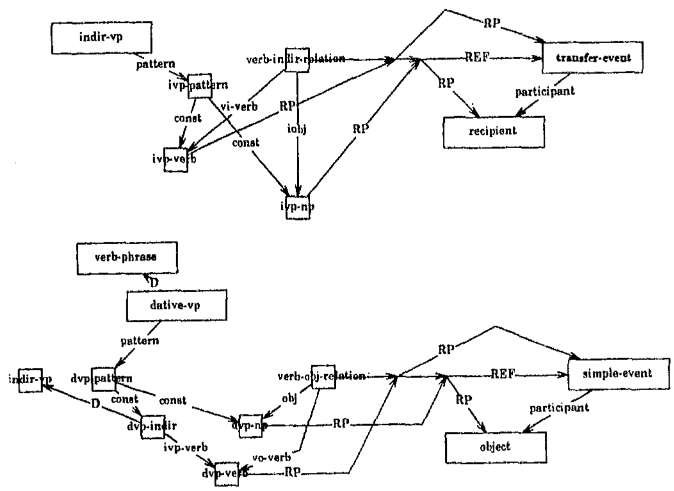

Figure 5: Knowledge linking verb phrases to events

network. This makes it easier to perform the knowledge-driven aspects of generation because no querying or complex matching is necessary. The use of the REF association also facilitates incremental generation by encoding knowledge about referential relationships as structured associations at various levels in the Ace hierarchy: The construction of a complete sentence involves the combination of structures derived from a number of REF associations, each of which refines those structures already active. [6]

4.3 Knowledge About Specialized Construc

The examples given in the previous section are simple illustrations of the use of associations between language and meaning at a variety of levels. These levels become especially apparent in the representation of knowledge about specialized constructs, whose meaning can be only partially represented as associations from more general linguistic structures.

[6] Those familiar with KL-ONE and other similar representation languages will observe that the Ace representation tends to avoid matching by assuming that the concept being generated from is assigned to a specific category. Thus some of the work done by template matching in systems such as PHRED is done during the classification of a concept in Ace. This is consistent with the Proliferation of Conceptual Categories principle: the generator need not obtain conceptual information that is more specific than that contained in the category in which a concept has been classified.

The means in which specialized knowledge is encoded varies according to the particular construct. The "give a hug" expression is interesting because there is no single syntactic structure which can be identified with the specialized interpretation, yet intuitively the specialized meaning seems tied to the use of the verb "give" in conjunction with the object "hug". While the metaphorical connection between the *hugging* action and the *giving* action is dependent on the "acting upon is giving" metaphor, this "hugging is hug-giving" metaphor must also be associated with the particular lexical items "give" and "hug". Figure 6 shows how both objectives may be accomplished: The link between ***hug-giving*** and ***hug-transfer*** is labeled ***view1'***, to indicate that it is DOMINATEd by the structured association between ***giving*** and ***transfer-event***. This association, ***view1***, relates the *source* of the *transfer-event* to the *giver* or *actor* of the *giving*. By inheritance and ROLE-PLAY, ***view1'*** associates the *giver* of the ***hug-giving*** action with the *source* of the ***hug-transfer*** event. The views ***view2*** and ***view3*** represent metaphorical associations that *actions* may be VIEWed as *transfer-events* from the *actor* to the *object*. ***View3*** indicates the correspondence between the *object* of an action or event and the *recipient* of the transfer, and between the event and the *object* of the transfer. ***View2*** represents the relationship between *source* and *actor*.

The association between the ***hug-transfer*** event and the ***hugging*** action, labeled ***view2,3'***, allows the inheritance of the knowledge that the *source* of the ***hug-transfer*** corresponds to the *actor* of the ***hugging*** action, that the *object* of the ***hug-transfer*** corresponds to the ***hugging*** action itself, and that the *recipient* of the ***hug-transfer*** corresponds to the *object* of the ***hugging***. The ***view2,3'*** association is DOMINATEd by both ***view2*** and ***view3*** and thus inherits their ROLE-PLAYs.

Knowledge about specialized constructs, such as "giving a hug", makes use of abstract structured associations between language and meaning just as knowledge about *selling* makes use of knowledge about *giving*. In this way the Ace representation is used to take advantage of generalizations in the encoding of specialized knowledge.

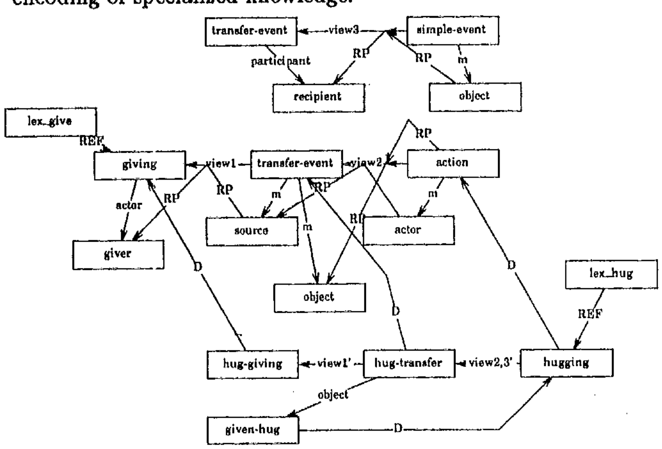

Figure 6: Knowledge used for "giving a hug"

5 Summary and Conclusion

The knowledge framework for language generation presented here shows how generalizations may be effectively exploited in the representation of linguistic knowledge. The application of the Ace framework to the representation of knowledge about *selling* shows how knowledge about *giving* may be used to represent information used in constructing utterances with the verb "sell". The example of "giving a hug" is used to demonstrate how the notion of a *view* captures knowledge about metaphorical constructs. The representation embodies a uniform, parsimonious encoding of conceptual and linguistic knowledge which seems to promote the extensibility of natural language systems.

References

[1] D. Appelt. TELEGRAM: a grammar formalism for language planning. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, Massachusetts, 1983.

[2] D. Bobrow and T. Winograd. An overview of KRL, a knowledge representation language. *Cognitive Science*, 1(1), 1977.

[3] R. Brachman. *et. al. Research in Natural Language Understanding.* Technical Report 4274, Bolt Beranek and Newman, 1979.

[4] E. Buchberger and H. Horacek. The generation component of the system VIE-LANG. In *First International Workshop on Language Generation*, Burg Stettenfels, West Germany, 1983.

[5] S. Busemann. Topicalization and pronominalization: extending a natural language generation system. In *Proceedings of the Sixth European Conference on Artificial Intelligence*, Pisa, Italy, 1984.

[6] A. Davey. *Discourse production.* Edinburgh University Press, Edinburgh, 1979.

[7] C. J. Fillmore. Topics in lexical semantics. 1977.

[8] D. Gentner. Structure-mapping: a theoretical framework for analogy. *Cognitive Science*, 7, 1983.

[9] P. Jacobs. Generation in a natural language interface. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, 1983.

[10] P. Jacobs. *A knowledge-based approach to language production.* PhD thesis, University of California, Berkeley, 1985. Computer Science Division Report UCB/CSD86/254.

[11] P. Jacobs. PHRED: a generator for natural language interfaces. *Computational Linguistics*, 11(4), 1985.

[12] P. Jacobs and L. Rau. Ace: associating language with meaning. In *Proceedings of the Sixth European Conference on Artificial Intelligence*, Pisa, Italy, 1984.

[13] M. Kay. Functional Unification Grammar: a formalism for machine translation. In *Proceedings of the Tenth International Conference on Computational Linguistics*, Stanford, California, 1984.

[14] K. Kukich. *Knowledge-Based Report Generation: A Knowledge-Engineering Approach to Natural Language Report Generation.* PhD thesis, University of Pittsburgh, 1983.

[15] K. Kukich. Semantic and syntactic constraints in natural language stock reports. In *proceedings of the 45th annual meeting of ASIS*, 1982.

[16] G. Lakoff and D. Johnson. *Metaphors we Live By.* University of Chicago Press, Chicago, 1980.

[17] J. Moore and A. Newell. How can MERLIN understand? In L. Gregg, editor, *Knowledge and Cognition*, Erlbaum Associates, Halsted, New Jersey, 1974.

[18] M. R. Quillian. *Semantic Memory.* Bolt Beranek and Newman, Cambridge, Massachusetts, 1966.

[19] R. B. Roberts and I. P. Goldstein. *The FRL Manual.* Technical Report AIM-408, MIT AI Lab, 1977.

[20] N. Sondheimer, R. Weischedel, and R. Bobrow. Semantic interpretation using KL-ONE. In *Proceedings of the Tenth International Conference on Computational Linguistics*, Palo Alto, 1984.

[21] R. Wilensky. KODIAK - a knowledge representation language. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, Colorado, 1984.

[22] R. Wilensky, Y. Arens, and D. Chin. Talking to UNIX in English: an overview of UC. *Communications of the Association for Computing Machinery*, 27(6), 1984.