INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. · Suite 600 · Berkeley, California 94704-1198 · (510) 666-2900

The Meaning of Reference in Embodied Construction Grammar

Jerome A. Feldman

TR-02-011

September 2002

Abstract

The ECG formalism is quite general, specifying only the ways to write and combine the four basic structure types: schemas, constructions, maps, and spaces. Grammars in ECG are deeply cognitive, with meaning being expressed in terms of conceptual primitives such as image schemas, force dynamics, etc. The hypothesis is that a modest number of universal primitives will suffice to provide the core meaning component for the grammar. *Referent descriptors* entered the ECG formalism as the way of specifying the participants in a semantic specification This note discusses how to specify entity-like referents, focuses on the key issues in Reference, and treats some of the more problematic ones in some detail. It assumes a general knowledge of the NTL paradigm and is not self contained.

The Meaning of Reference in Embodied Construction Grammar

The ECG formalism is quite general, specifying only the ways to write and combine the four basic structure types: schemas, constructions, maps, and spaces. A grammar of English or any other language requires specific realization. The currently preferred way of thinking about grammars follows generally from Croft's Radical Construction Grammar [Croft 2000]. He suggests that the grammar for any language must capture three basic mechanisms: reference, predication, and modification. When we look beyond the individual sentence, a fourth mechanism which we can call discourse structuring, is also needed. This note focuses on the first of Croft's dimensions - reference and also deals with modification as it effects reference. Predication and Discourse are the subject of separate notes and will be mentioned only in passing.

This working note incorporates ideas from several members of the NTL group and has been fairly stable for about a year. It assumes a paradigm for language understanding comprised of two distinct phases. The first, *analysis,* phase takes an utterance in context and produces a semantic specification, the *SemSpec*, which is used by the second, *enactment,* phase in understanding the utterance. This is all described in various papers of which [BC 2002] and [Feld 2002] are the most recent. Within this paradigm, it appears that we can specify rather complete grammars using only four types of formal structures: *schemas, constructions, maps,* and *spaces.* A companion note ICSI TR02-10, specifies these structures and presents simple examples of their use.

The ECG notion of a grammar is rather broader and deeper than conventional grammars. The role of an ECG grammar is to specify everything needed for the analysis of an utterance (in context) into a SemSpec for subsequent *Enactment*. In addition to the grammar, we assume that there will be one or more external ontologies involved, with the obvious links between lexical items and ontology items (ExItem) and between ontology relations (ExRel) and the relations used in the grammar. In the grammar, external category constraints (ExCat) from the ontology can be used to specify role restrictions. External predicates in the grammar will be need to be computable in the associated external ontologies. For example, a role in some construction might require its filler to construable as valuable, dangerous, etc. We assume that predicates like these can be evaluated within the external ontology, returning logical or probabilistic answers.

This follows the general linguistic paradigm that a grammar of e.g., English, can be independent of much of the our detailed world knowledge and that people can learn new words and fields without changing the basic grammar. From an applied perspective, this means that we can build a core NLU system that can be used with novel applications by specifying interfaces to the ontology and Enactment modules for that domain. From the neural/psychological perspective, this says that only part of human knowledge is schematized for language analysis. Our model suggests that there is continual interaction between linguistic and more general world knowledge in language generation or understanding.

The immediate consequence of this stance is that we will NOT recreate all world knowledge as a collection of schemas and relations. Only the categories, predicates, and schemas needed for Analysis must be defined. It is not obvious that this separation of grammar and detailed meaning can be achieved, but that is our goal, for the reasons described just above. Some grammatical features (case, gender, etc.) will be quite like those of unification grammars such as HPSG [HPSG]. But there is an additional novel idea being explored in ECG.

Grammars in ECG are deeply cognitive, with meaning being expressed in terms of conceptual primitives such as image schemas, force dynamics, etc. The hypothesis is that a modest number of universal primitives will suffice to provide the core meaning component for the grammar. Specific knowledge about specialized items, categories and relations will be captured in the external ontology as ExItem, ExCat, and ExRel respectively. External items, etc. can appear in an ECG grammar and new ones can be freely added provided only that they are well defined in an external ontology. Language understanding is only effective when the underlying content is understood.

This note focuses on the key issues in Reference and treats some of the more problematic ones in some detail. It assumes a general knowledge of the NTL paradigm and is not self contained.

Referent descriptors (sometimes shortened to *RD)* entered the ECG formalism as the way of specifying the participants in a semantic specification or SemSpec. They typically describe Entities, but will sometimes describe Events, Discourse Segments and possibly other types. This note focuses on how to specify entity-like referents. The entire SemSpec, including the referent descriptors discussed here, is intended to bridge between linguistic constructs and the enactment of their meaning. Within the formalism, referent descriptors are *schemas*, the basic semantic unit of ECG. Any proposal for the ECG representation is tentative until we demonstrate how analysis can produce it and how enactment can simulate it.

Nancy Chang made a fine start in her ECG toolbox where each referent could have roles:

Category
Number
Gender
Distribution
Predications

As we get into implicit referents, quantification, etc., we will need to refine and extend these roles. The Bergen and Chang ECG paper [BC2002] adds the good idea of a Resolved Referent role, which now seems better named as Reified Referent for reasons that will be outlined. Open questions there involve how and when the Reified Referent gets resolved using discourse context, etc. As is often the case, the connectionist story is cleaner; the effort to resolve references is ongoing and is parallel with analysis.

This note considers a wide range of issues involving referents descriptors in no particular order. We start by looking at some basic roles and constraints of the general referent schema.

Attributes and Restrictions

We will refine modifying Predications into the two traditional kinds of constraints: Attributes and Restrictions. The phrase:

"All the nerdy programmers"

could be talking about an *Attribute* of all programmers or could be *Restricting* consideration to a small subset of programmers. This is a crucial semantic distinction.

Both Attributes and Restrictions will be open class in the sense that any predication can potentially be used in these roles. Predicates can be arbitrarily complex and include any ECG Predications [BC2002] involving the entity being specified. We will use the special tag SELF to allow predicates like: Bigger(SELF, breadbox). Technically, a modifying predicate can either be a predication schema in the grammar or a fixed predicate (ExRel) of the supporting ontology. For example, the relative sizes of two entities can be essential for analyzing sentences like the classic: "The box is in the pen", but do not need to be explicitly encoded in the grammar. The model assumes that the external ontology can provide answers to specific queries like the example above.

One important function of Restrictions is to allow the SemSpec to include Referents that are not fully resolved - such as favorite restaurant or next year's champion. Unresolved referents will also be crucial in the treatment of pronouns and definite descriptions. The idea is that the Enactment can resolve these, ask about them, or execute with a partially unspecified entity. One current idea for handling wh-questions is to use tagged referent descriptors to specify what is known about the desired answers, following standard AI practice.

It seems that the two mechanisms of Attributes and Restrictions combined can handle all modification in Entity descriptions. In addition to these predications that are expressed directly in a referent descriptor, the RD itself can be used all sorts of other predicates, which further restrict it or expand its attributes. Following the general ECG practice, we consider any referent or predication descriptor as a specification of only *some* of the links involved; there will usually be several schemas bound together multiply in any SemSpec.

Restrictions will be stated in the CONSTRAINTS section of a referent descriptor (RD) schema. Constraints can also be declared to hold in only some phase of Enactment, using the :: notation. This seems to be a good way to handle the distinction between permanent and temporary predications as in the Spanish *ser* and *estar*. Keith Sanders has been looking into this under the name of stage versus individual properties. We exploit the idea of State-maintenance, one of the abstract event types. So "the castle on the hill" will lead to a referent with a restriction like: On(hill4, SELF). But "the bus on the hill" will lead to a referent with a restriction like: Interval:: On(hill4, SELF) using the :: notation for situational invariants. The :: notation encodes other important functionalities in the formalism and there is a recent paper describing this by Chang, Narayanan, and Petruck [CNP 2002].

Categories and Distribution

Categories and Distribution are crucial in the current design because they are going to carry much of the burden for Quantification and binding. Distribution will be closed class, with a fixed set of possibilities. Categories are divided into the standard count and mass types, plus an additional structure or compound type. The current design allows the following possible distribution types.

Individual of category T (generic, prototypical, ideal, etc.)
Set of individuals of category T
Count category T itself
Mass of category U
Mass category U itself
Structure – a schema that has named roles of type RD

The possible values for the Category role include any category from the external ontology (ExCat) and also any ECG schema type. This allows the free introduction of new categories of RD without structural change in the grammar. There was a question about whether Bounded Mass is a distinct distribution type, but there doesn't seem to be a need for this. The fact that a certain RD represents a bounded mass can be captured as a constraint. Of course, almost any count category can sometimes be construed as a mass.

We note in passing that the Gricean informativeness (I) principle here seems to be captured nicely by choosing a prototypical individual by default for unmarked expressions. The related manner (M) principle suggest that a more complex referring expression maps to a generic individual, not the prototype, with additional restrictions arising from the modifiers. As we will see later in this note, the Reified Referent role of an RD can be filled by a generic or other representative of its category.

One way to treat event-like referents (e.g., the last Derby) is to allow the category T to also cover event categories, whatever they may turn out to be. This is the current choice. With the new results on Discourse, we will also extend the possible categories to include various kinds of discourse segments.

A crucial part of the current design is what computer scientists call de-referencing of an RD. Any RD is a (partial) specification of some entity, event, etc. The basic unification (<->) operation of ECG requires that its two arguments have compatible types or categories. If we follow this rule strictly then an RD is never compatible with an explicit category such as Container. What is needed is an extra rule (de-referencing), which states that an RD whose category role has value C is unifiable with any role that unifies with C. That is, for purposes of unification or binding, an RD of Category C is treated as being of Category C. Linguists can see this as a metonymy where the referent for an entity can stand for that entity, as in: "The White House declared war" .

Structured referents are intended to handle everything from an organization (ICSI, Lakers) to formal compounds like conjunction or disjunction. Structured RD will also be used for representing coupled concepts like "bottle of beer" which is both a bottle and a quantity of beer. There will be a number of examples below.

There was an idea of making scales another basic category, but that doesn't seem to be right. Scales will be treated as specific kinds of parameterized schema, which can appear (inter alia) in restrictions or attributes of referent descriptions. Scales will be discussed later in this note. Similarly, the cardinality of a set will be represented as a constraint - restriction or attribute, as appropriate.

Reified Referent

As we have discussed, in general an RD can describe a referent without specifying a specific item (ExItem). For example, there could be quite a lot of discourse about the perpetrator of a crime without naming him or her. But when the specific ExItem is named or computed, we need a way to have that ExItem as a role of the RD – this is the function of the Reified Referent role. In this case, the old name, Resolved Referent, would be equally appropriate.

However, there is a related use of this role that does require the "Reified" terminology. This arises from the decision to capture detailed world knowledge in the external ontology. Quite often an RD will refer to an individual of some ontology category (e.g., café) without specifying a specific café. Now there is quite a lot known about cafés in general and even more about the prototypical café. Most of this knowledge is rarely used in grammar, but there are some predicates (like relative size, fragility, etc.) that need to be evaluated in deciding how well some construction fits the current discourse and context. For such purposes, an RD must provide a link to generic or prototypical ExItems as well as specific resolved referents. As one example, for an RD to unify with the protected role of a "safe" schema it must be construable as valuable and vulnerable. In the standard example "Harry walked into the café " there should be a lexical entries something like:

```
Lexical Construction café
  SUBCASE of Common Noun  / / default as neuter singular nominative
  FORM  "kaefe"
  MEANING
     Retail Business RD   / /  a general Referent Descriptor
     SUBCASE OF Building  / /  also a subcase of the Building schema
  CONSTRAINTS
     Category <- Restaurant
     Reified Ref <- ExItem( coffee-shop.proto) / /  knowledge about cafés

Lexical Construction Harry
  SUBCASE of Proper Noun  / / default as neuter singular nominative
  FORM  "hari"
  MEANING
     Person RD    / / a general Referent Descriptor
  CONSTRAINTS
     Category <- Fictional Person  / /  itself a subcase of Person
```

Gender <- Masculine
{Givenness <- Proper Name}   / / probably already in parent
Reified Ref <- ExItem(Harry Lime)   / / specific ontology item

The idea is that all lexical constructions in the grammar share some roles and constraints, including the constructional agreement features person, number, gender and case (PNGC). Specifically grammatical features, like PNGC, should be explicitly set in the RD. Other roles and default values come with more refined constructions (Proper Noun) and semantic schemas (Retail Business). In the case of a common noun, the default link is to a prototypical ExItem (cf. Grice), but when a specific ExItem is known this becomes a more informative Reified Referent, including for proper nouns. For example, when an Adjective-Noun construction parses something like grungy (elegant, etc.) café, it will not only add a modification schema to the RD, but will also replace the default prototype ExItem with a more appropriate one, often the generic item with the appropriate role~value set.

The main use of the Reified Ref during analysis is to help decide how well a given RD fills some role or (equivalently) unifies with a role. For example, knowing that a café is a building will suffice to allow it to unify with the container.inside role of "into" because buildings are known in the general grammar to be construable as containers. Most of the grammatical uses of café can be deduced from its status as a building and as a retail business. But there is also specific domain dependent knowledge about cafés that could be needed in a particular analysis decision. For example, "café society" could be a society interested in cafés, a decorating business, a 1995 movie, or any number of other things - as a Google search will reveal.

Consider the standard example "safe café", recalling that safe evokes a protection schema with three salient roles: protected, threat and barrier. The most natural reading would have café bind to the barrier role; this would arise from a constraint in the protection schema something like:

ExBarrier(self.barrier),

which would require the reified ref of any RD that unified with the barrier role of this schema to be construable as a barrier in the external ontology. This might follow from knowledge of a particular café as being a good place to meet in secret.

In the context of a performer trying out a new act, a safe café could be one that was not too threatening.
In this case, café would unify with the threat role as a weak threat. This appears to require force dynamics and contrast and is too complex for current techniques.

A simplified version might be captured by something like a constraint in the protection schema:

ExLessForce (self.threat, self.barrier)

saying that the force of the reified ref of the RD unifying with the threat role must be less than that of the barrier - here quite possibly a default for this performer. All of this is complicated and it only gets worse, we shouldn't expect language to be simple.

Quantification

We need to add a Quantification role to Referent descriptors. This will include much more than the traditional logic quantifiers. It will be closed class with values like the following:

All, most, a few, none // Quantifier-scale
Choose 1, with (any) or w/o replacement
Enumerate one at a time (each)
Whole set or collective, as opposed to all of its elements
Specific individual, not necessarily resolved

The first Quantification option fits with set (collective) or mass referent descriptors. The next three work with set referent descriptors and the final option is for individuals. This brief presentation of Quantification is misleading. The treatment of quantifiers strictly within referent descriptors is the crucial decision for the whole approach. A full discussion is beyond the scope of this note, but here is the general idea.

We propose to eliminate global quantifiers from the SemSpec and to capture the required functionality by relations among referent descriptors. For this purpose, the formalism must include a way to name the element chosen by one of the Choose or Enumerate quantifiers; we will use the notation:

RefA.choice

to specify the element of RefA (which must be a set) that is currently selected in Enactment. There is a related mechanism for specifying conditions on an arbitrary element of a set-like referent descriptor that will be described below. The examples section shows how these mechanisms enable us to treat "all" the hard cases.

Givenness and Indexicals

Another piece of the Referent story is the treatment of linguistic indexicals such as here, then, or that cat. Obviously enough, the temporal indexicals will need to be worked into the general tense story, which I am guessing will not need much change from the Gildea, et al. paper [CGN] . Spatial indexicals like "here" can involve quite a lot of context and construal, but presumably resolve to some location associated with the speaker.

This seems to go beyond Lambrecht's classical work on accessibility. Gundel et al. [Gun] have a nice categorization of Givenness into six levels: in focus, activated, familiar, uniquely identifiable, referential, and type identifiable. There are at least two more levels at the lower end: proper names and qualified proper names like "the band Chicago". We will have such a Givenness / Accessibility role; it will help a lot with reference resolution. Once the referent is resolved, Enactment need not worry about Givenness, but it does still play a role in further constructions, for example in finding best antecedents.

Intonation and Focus

There is still more to do, but the optimal answer is clear - ideally there will be no unresolved indications of undifferentiated intonation or focus in the referent descriptor remaining for Enactment to deal with. We will need some kind of intonation role in the referent descriptor during Analysis. There seems to be a generally agreed upon range for intonation markings (H+, etc.) and we should probably follow that.

The goal is to work out the various uses of intonation and their constructional effect on the referent descriptor. For example, a stressed prenominal adjective seems to always denote restrictive rather than attributive modification .e.g., the NERDY programmers. Many cases of stress also introduce contrast; the example above suggests evoking a contrasting set of cool programmers as another referent likely to be required in analysis.

Mats Rooth (and maybe others) [Rooth] suggests that focus in general introduces a base class for comparison. His examples are like: "John introduced Bill to SUE" , which entails a set of people who Bill was not introduced to. There may be terminological problems with using "focus" to refer to this general contrast property. But we should be able to handle such sentences by introducing two referent descriptors, one for Sue and one for the set of contextually relevant people, less Sue, to whom Bill was not introduced. We will see in the next section how stress contrast interacts with negation.

Another important use of intonation is to mark newness. There seem to be fairly well understood links between intonation and the Givenness marking. By having each referent descriptor retain its intonation marking we should be able to deal with hard cases like:

> Fred called Bill a Republican and then HE insulted HIM.

Givenness can be marked in English by “deaccenting”, overriding the standard stress placement in a phrase. Other languages (including Korean and Japanese) have focus and newness marked by “dephrasing” - the reversal of the usual process of combing familiar word sequences into one unbroken phrase. And there are additional uses of stress and other intonation patterns that are yet to be considered. This may well work out best in combination with the ECG treatment of morphology .

Negation

Referent descriptors will also play a central role in our treatment of negation. More or less following Horn , [Horn], we will divide negation into three conceptually distinct notions. This discussion only covers negation within referent descriptors - negation is also a crucial part of the Predication formalism and metalinguistic negation also needs to be treated.

As expected, ordinary negation can appear in the Attributes or Restrictions of any Referent. But there is another negation primitive that will also be common in Referents. The classical names for these two ideas are Contradictory and Contrary Negation, but this is confusing and we will tend not to use these terms. The standard minimal pairs are like “unhappy” and “not happy”. The specific concept “unhappy” is semantically more precise (Contrary case) than the general negation “not happy”, which could represent anything from desperate to ecstatic, omitting only a range around happy on the happiness scale. Scales will be discussed in detail the next section.

We introduce an explicit function, Contrast, to represent the idea of a value that is only specified as contrasting with some stated value on a scale. Typically a Contrast term will appear as a Restriction in a Referent descriptor. For example, a phrase like “the man who was not friendly” would yield a Referent having a restriction:

Contrast (Friendliness-scale, friendly).

This captures the notion that “friendly” is an element on the friendliness scale and that the man has some other value on that scale. We will also add primitives for Contrast-up and Contrast-down to cover the cases where the language specifies that one end of the scale is precluded like “the man who was not at all friendly”. We could also allow Contrast to operate on discrete sets (e.g. colors), but the focus mechanisms of the previous section allow this use as a special case - the referent is some basis set less the contrast element.

In general, negated predications without focus do not make clear positive statements. For example: "John did not introduce Bill to Sue" could mean any number of things and Enactment will often need to ask for clarification, as people do. If there is stress on Sue, then the meaning becomes - there are other people to whom John did introduce Bill - and this can be represented using a referent descriptor for this currently unresolved set. The ECG treatment of predication will need to handle both the underdetermined and the explicit contrast possibilities.

The third flavor of negation is what Horn calls (appropriately) meta-linguistic uses of negation. These include references to discourse like the famous line: " That was no woman, that was my wife". Another example would be: "He is not happy, he is ecstatic". There is a wide range of these meta-linguistic constructions including some that we have looked at, like one anaphora. The current idea on meta-linguistic constructions involves a variant on EVOKES (maybe RETRIEVES) that matches a construct from a (not too much) earlier analysis. This all seems to be separable and will be treated later.

Scales

The preliminary ideas on scales are also simple. Scales are treated as a specific category of parameterized schema. All scales are assumed to have an underlying quantitative basis in the real interval [0,1] although this and other quantitative matters are more a matter for Enactment and not of much import in Referent descriptions. Scales also have a linguistically preferred end, usually the top. So one talks about how hot, tall, happy something is. But we can also ask how cold something is, where the preferred end is the bottom. There are good arguments for linguistic scales being one sided, but we could add bi-directional ones if needed. Infinite scales are metaphorical.

A place on a scale will be identified with Gaussian distribution (Bell Curve) with a center and spread. For example, "happy" might be broadly identified as Happiness-scale(.6, .3) and "ecstatic" more narrowly as Happiness-scale(.9, .1). We will NOT attempt to use Fuzzy Logic with these interpretations, but there are nice questions of how to apply Belief Nets.

This general formulation seems to support a uniform treatment of many modification phenomena. For example, "very happy" could take the values for happy and both raise them and narrow the range yielding Happiness-scale(.65, .25) and so on for iterated "very"s.

The current design involves a general linear scale schema with roles *bottom* ,*top* *category*, and *property*. For the weight of cows we might have, *category* = cow , and *property* = weight, *bottom* = 100 lbs, *top* = 2000 lbs. A constituent of every linear scale property will be a "bell map", which maps modifier terms onto (center, spread) places on the absolute scale [0,1]. For example, as a weight modifier "heavy" might map on to (.6,.3) . This would yield an estimate for a heavy cow at 1140 lbs, with a spread of 570 lbs.

We will also need scales for Modification in Predicates, for examples such as "pushing (very) hard". Again, there doesn't seem to be any inherent problem in having an Effort-scale, etc. The resulting numerical estimates become parameters to Enactment routines. This needs to be discussed as part of Predication.

An interesting combination occurs with scaled quantifiers like "many", etc. It seems that we can define, as one of the fixed Quantifier types, a specific Quantifier-scale that obeys the usual scale composition rules, e.g.,. "very many". Interestingly, the Gricean maxim of quantity (Q) comes almost for free with this treatment. If "many" corresponds to (.6,.2) on the Quantifier-scale then using the word "many" implies that the quantity specified is not nearly all of the possibilities. George Lakoff points out that there are linguistically different scales for count (many) and mass (much) referents, and these should be included.

Another combination arises with contrasts and scales in expressions like "not at all happy". This becomes:

Contrast-down(Happiness-scale(.6, .3)).

One could imagine a calculus that would then yield Happiness-scale(.2, .2), but this is not a current concern. We will also need to handle "not at all unhappy" as Contrast-up(Happiness-scale(.6, .3)). It seems straightforward to write the constructions for all this, but it hasn't yet been done.

Presuppositions

I haven't yet found the equivalent of Horn for presuppositions and so might not understand the problem. At least some simple cases seem to be handled nicely by lexical constructions. The classic example "regret" would add to the SemSpec two predications, one saying the protagonist did something (possibly unknown) and another expressing that he was sorry for it. This issue also comes up with referents in a case like "Mary's winning essay was short" which adds a Restriction won(Contest6, SELF) to the referent descriptor for the essay.

One classical worry that will not concern us is the status of "The King of France is bald" - does it presuppose the existence of said king? With mental spaces, it is fine to populate them as we like. Considering cases like "Not even one boy laughed.", gets much harder. This entails (?presupposes) that the hearer would have expected some boys to laugh -- over to Keith. We could try a version of the schema trick and have a “failed expectation” schema with the expected and experienced predications as roles. Schank built a theory of learning on this idea.

Some Standard Examples

It will take a lot of work to see if the proposed mechanisms for quantifiers, scales, modification, negation, and contrast will handle all the required cases of referents, but there are no killer counter examples yet. Here are some introductory examples to help convey the ideas, in a simplified notation. Consider the two readings of the standard example:

“Every boy danced with a girl.”

In both cases, the SemSpec will include some specification of couple-dancing (say dance4) with two referents, and the boy referent, B, is the same for both readings:

Predication:	Referent Descriptors:
dance4	Referent B; Category Human
roles:	Distribution: SET
ref1: B	Gender: MALE / / Comment this is a shorthand
ref2: Gn	Restriction: x in SELF => attended(x, party11)
	Quantification: Enumerate (each)

The restriction on B introduces the notation for describing properties of elements of a SET type. The => notation species that the predicate on the right must hold for the variables named on the left. Notice that we could handle an example involving “most “ boys by changing the Quantification role value to be Quantifier-scale(.7,.2). For the case where only one girl was involved, the referent Gn is:

Referent: G1; Category Human
Distribution: individual
Gender: FEMALE
Restriction: attended (SELF, party11)
Quantification: specific

The Entity G description for the other reading, involving several girls, is:

Referent: G2; Category Human
Distribution: SET
Gender: FEMALE

Restriction: x in SELF => attended(x, party11)
Quantification: Choose 1

The idea is that Enactment uses this bit of SemSpec to model each boy dancing with some girl or another. To get "Each boy danced with a different girl" we just have the Quantification of G2 be "Choose 1 without replacement". The details of how to actually do such simulations remain to be worked out and this is the major open issue.

Suppose that we had "Every boy but one danced with a girl". This seems to Involve three referents: Bx, the exceptional boy plus B and G from above. Taking the narrow, G1, reading, we would get something like:

Referent Bx; Category Human
Distribution: individual
Gender: MALE
Restriction: attended (SELF, party11) & NOT dance4(SELF, G1)

The descriptor for the remaining set of boys, B, would have its Restriction become:

Restriction: x in SELF => attended(x, party11) & NOT Bx

The other reading would have G1 replaced by G2, the descriptor for all the girls at the party.

One of George's examples was: "All the boys Nancy danced with were similar" This could be written as the usual set of boys with the restriction:

x, y IN SELF => similar(x, y).

With this mechanism, we could rewrite the referent descriptor for a set of boys abbreviated as Gender MALE to be a restriction:

x in SELF => Gender(x, MALE).

As we will see, relations among elements of set-like referents are central to our treatment of tricky cases.

Unresolved referent descriptors could also be the key to enacting questions. Specially tagged referents could denote desired answers and their restrictions. For example, a question : "Which boys danced with Nancy? " will yield the query referent:

Query Referent Bz; Category Human
  Distribution: SET
  Gender: x in SELF => Gender(x, MALE).
  Restriction: x in SELF => attended(x, party11) & dance4(x, Nancy)

Another example: “Two boys ate three pizzas.” There is a reading where they shared the pizzas and one where they ate three each. The first reading could be treated as:

Referent: B2; Category Human
 Distribution: SET
 Gender: MALE
 Restrictions: SIZE(SELF) = 2
 Quantification: Whole Set

Referent: P3
 Distribution: SET of PIZZAS
 Restrictions: SIZE(SELF) = 3
 Quantification: Whole Set

Saying that the boys as a group ate all the pizzas. For the second reading we could introduce distinct referents for the various boys and pizzas so we can deal with follow on statements like: “Billy had two pepperoni”. A third reading, which would leave everything anonymous, parallels the final donkey example below. A somewhat different issue also comes up: “The boys ate pizza or burgers”. Here we want to preserve the disjunction in either reading and so should use a collective referent of type disjunction.

Referent: D7
Distribution: OR2

Schema OR2
  SUBCASE OF STRUCTURED RD
  Category: FOOD
  Distribution: INDIVIDUAL
  Disjunct1: pizza
  Disjunct2: burgers

As a final illustration, let’s consider a humane version of the famous donkey sentence: If a farmer owns any donkeys, he feeds most of them. The classical problem is to capture the fact that each farmer feeds his own donkeys. Our solution involves two referent descriptors as usual and one predicate:

```
  feeds
    feeder: F
    fed: D
    food:
```

with,

Referent: F; Category FARMER
Distribution: SET

Referent: D; Category DONKEY
Distribution: SET

Restrictions:
x IN SELF => owns(F.choice, x)

Quantification: Enumerate *Quantification: Most*

Here the referent D has a restriction that depends on the current value, F.choice, of the enumeration over the set of farmers, F. Simulation proceeds by first choosing a particular farmer. Them the referent D, all the donkeys owned by that farmer is computed. Finally, the predicate feeds(F,D) simulates the chosen farmer feeding most of his donkeys. In fact, there is almost never a reason to enact such general statements and they would normally be stored for later use in a particular situation.

It is worth also looking at the ECG version of the Mental Space solution to this problem, as described in the original book [Fau]. Fauconnier sets up an if-then mental space mapping, which we can represent as a schema. In the simple case where each farmer owns one donkey, the target sentence can yield the following elegant bit of SemSpec:

```
SCHEMA: if_then_feeds
  SUBCASE of if_then
   if:     owns (F,D)
   then:  feeds(F.choice, D.choice)
```

where :

Referent: F; Category FARMER	Referent: D; Category DONKEY
Distribution: SET	Distribution: SET
Quantification: Enumerate	Quant: Choose 1 w/o replacement

This does not extend nicely to the "most" case and that seems to require setting up a full mental space and a mapping from each farmer to his donkeys. Interesting. In any case, the resulting SemSpec is a again general statement and might be used in various ways by Enactment.

Structured Referents

So far the entire discussion has focused on separate referents. But many referring expressions involve multiple interacting referents, for example, measure expressions (flask of vodka), other descriptive phrases (picture of Paris, cross of gold) or noun compounds (coast road). These can interact in complex ways: "She grabbed and drank the small delicious flask of vodka". The SemSpec needs to specify that it was a small flask that was grabbed and delicious vodka that was drunk.

The mechanisms already described allow us ways to do this. For example, we could have a referent for a bounded mass of delicious vodka bound to some contents role of a container referent for the large flask. But there are reasons to use a different method.

An alternative way of capturing the same information is to have Structured RD schemas that can describe instances of e.g., the containment relation. It could be:

```
SCHEMA Contains
  SUBCASE OF  Structured RD
  ROLES: container: COUNT RD
         contents : MASS RD
         fullness: SCALE
```

This separate schema approach has the advantages that it is symmetric in its two main roles and also that instances of the schema itself could be used in certain bindings. This version is also more like the triangle node realization that we envisage for the connectionist version. Of course these relations can be nested - “a case of small delicious flasks of vodka”. As we discussed earlier, Structured RD is the general type for complex referent descriptors that can be used themselves as a kind of referent.

Possessives present an interesting case for coupled referents. Of course the possessive marker can encode many different relations including part-of, colleague, relative, etc. It seems fine to capture these as structured schemas binding the two referents, which again can be complex and modified in various ways. This is discussed further in the section on modifiers when we consider “Ben’s best bike”. Again possessives do have conventional heads and so can also be treated as such.

We call these bindings of two or more referents to a relational schema something like a *structural* RD. As George points out, only some of these relational schemas (we call them *coupling*) form a unit that itself can serve as a referent. The treatment of complex referents as a structure linking referent descriptors appears to have many advantages. Another one is that each individual RD can retain its discourse roles such as intonation and givenness for later constructions like one-anaphoras to work with. Collapsing everything into some kind of head seems to preclude this.

Structured versus Collective(set) Referent Descriptors

We have introduced both structured (bottle of beer) and set or collective (six-pack) referent descriptors. The difference is that a set RD denotes a multiplicity that has multiple similar and undifferentiated elements, while a Structured RD denotes a single referent made up of (possibly disparate) entities, each having a separate role and referent descriptor of its own. Of course the role of one structured RD can, recursively, be another structured RD like when a company has groups, which have divisions, etc. This is completely analogous to the distinction in programming languages between arrays or list of similar items and records or structures with fields and values of disparate types.

The Referent Descriptor Schema – current design

Following our current rules for formalism, the referent descriptor should be a SCHEMA of the following form. Recall that a role name ending in * can have multiple fillers.

```
SCHEMA Referent Descriptor
  SUBCASE OF Grammatical Primitive
   ROLES:
     Category    // through de-referencing, RD can unify with this category
     Distribution
     Agreement roles*  //  whichever are needed PNGC, classifiers, etc.
     Quantifier
     Givenness
     Intonation
     Attributes*
     Reified Referent
   CONSTRAINTS:
     Restrictions*  // restrictive predications on SELF
```

The reasoning is that these properties specify what the referent *is* - other properties can be expressed by simple relational schemas as well as by complex predications involving this referent. The design also entails a SUBCASE lattice on different kinds of referent descriptors, still to be worked out in detail. One specific constraint that will be needed is:

Distinct(<Referent>*)

which specifies that Self does not specify the same reified referent as any of the arguments of Distinct. This is needed for constructions like “other shoe” and many others.

It is important to list all the restrictive predications as CONSTRAINTS, because these are used to resolve the referent. But it does not matter which attributive predications appear directly in the Referent Descriptor and which appear separately. This captures the connectionist notion that all of the predications involving a particular referent will mutually activate one another. From the computational perspective, the SemSpec will work with all the attributes of each entity, whether or not they were specifically mentioned in the current discourse segment.

An intriguing possibility is to also specify some of these semantic roles, and perhaps others without direct semantics as “unification features” that are automatically unified. This would exploit the elegant implied unification device of conventional unification grammars like HPSG. Something like this is included in the current ECG effort on morphology and agreement.

Modifiers

We should say more about the modifiers of referents, prototypically adjectival phrases. We talked earlier about scales, such as weight, and that is part of the story. In keeping with the current approach, there should be schemas for modification relations, e.g.,

```
SCHEMA Color of
    SUBCASE of  Nominal Modification
    ROLES:
            landmark: physical object
             hue: color
             intensity: scale
             modification: adverb or intonation
```

If the color of some referent is a *restriction*, this fact must be expressed as part of the Restriction role in the descriptor, if not the schema expressing this fact can be an Attribute of the Referent. There is no semantic distinction between adding a attributive modification to the referent descriptor and having the descriptor RD appear in a separate predication, it is just notational convenience. This captures the fact that, from the connectionist perspective, all uses of a given referent are mutually activating.

Notice that the relational schema way of representing modification facilitates comparisons across referents as in one-anaphora. Another nice feature is that qualified or emphasized descriptors can have that fact noted with a modification feature in the relational schema descriptor. So we can combine modifying expressions with adverb scales to get things like “brilliant blue sky”. Some cases can be treated directly as scales, but others will involve complex nesting of modifying relations.

Comparatives and superlatives have an interesting formalization in the current scheme. Each involves two referent descriptors - one for the basis set and another

for the item being described. For example “Ben’s best bike” involves one referent descriptor R33 for the set of all Ben’s bikes and a second, R44, which is an individual bike with the restrictions that it is in R33 and that
x IN R33 => better(SELF, x), using the convention above for naming set elements.

Adding some detail, we see that R33 can be described as a set of bikes with the restriction that these are all owned by Ben. This restriction is an OWNS relational schema binding R33 to a descriptor for Ben. To do “Ben’s second best bike” we would add a restriction that R33 is ordered by “better” and change the restriction on R44 to ORDINAL(2,R33,SELF) or some equivalent notation.

Notice that this use of contrasting basis sets interacts nicely with the treatment of semantic focus. For example, “ Ben’s SECOND best bike is new” entails a referent for some other set of Ben’s bikes that are specified to be not new.

Open Issues

This seems to be enough machinery to handle all the examples raised by Lakoff, Langacker, etc. modulo other representational problems that we have not yet solved, mainly concerning the related issues of context, construal and reference resolution.

But any treatment of reference only makes sense in the context of mechanisms for predication and discourse structure. The ECG formalism contains consistent suggestions for treating metaphor, mental space, and blending phenomena but these need to be evaluated in detail. The general representation ideas such as the ones outlined here need to be extended to include constructions that can produce SemSpecs and manipulate the representation. Finally, all of the ECG mechanisms are directed towards enacting the SemSpec and propagating the consequences back to update the current belief state and that is still far from settled.

References

BC 2002 Bergen, B.K. and N. Chang (2002). *Embodied Construction Grammar in Simulation-Based Language Understanding.* To appear in Ostman and Fried (eds) Construction Grammar(s) Cognitive and Cross Linguistic Dimensions. *John Benjamins, to appear.*

CFPS Chang, N. J. Feldman, R. Porzel, and K. Sanders (2002). *Scaling Cognitive Linguistics: Formalisms for Language Understanding*. Proc. First ScaNaLU Workshop, Heidelberg, June 2002

CGD N. Chang, D. Gildea, S. Narayanan (1998). *A Dynamic Model of Aspectual Composition*. In Proceedings of the 20th Annual Conference of the Cognitive Science Society (COGSCI-98), pages 226-231, Madison, WI, 1998.

CNP Chang, N., S. Narayanan, and M.R.L. Petruck (2002). *Putting Frames in Perspective*. Proceedings COLING 2002.

CROFT Croft, W. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford 2001

DW 2002 Dodge, E. and Abby Wright *Herds of Wildebeest, Flasks of Vodka, Heaps of Trouble: An Embodied Construction Grammar Approach to English Measure Phrases.* Proc. Berkeley Linguistics Society, 2002

Fau Fauconnier, G. 1985. Mental Spaces. Cambridge, Mass.: MIT Press.

Feld 2002 Feldman, J. *A Proposed Formalism for ECG Schemas, Constructions, Mental Spaces, and Maps* ICSI TR-02-010, Berkeley CA, September 2002

GHZ Gundel, J., Nancy Hedberg and Ron Zacharski. 1993. *Cognitive Status and the Form of Referring Expressions in Discourse.* Language 69.274-307.

Horn Horn, Laurence R. *A Natural History of Negation*. Center for the Study of Language and Information. 2001

HPSG Sag, I. and Tom Wasow; *Syntactic Theory: A Formal Introduction*, CSLI Press, 1999.

Rooth Rooth, Mats *A theory of focus interpretation* , Natural Language Semantics 1, 75-116, 1992.