# The Origin of Linguistic Categories

Luc Steels
SONY Computer Science Laboratory
6 Rue Amyot, 75005 Paris
and VUB AI Laboratory
Pleinlaan 2, 1050 Brussels
steels@arti.vub.ac.be
(draft Evolution of Language
Conference - London 1998

June 1, 1998

**Abstract**

The paper presents cognitive mechanisms and behavioral rules by which a group of distributed autonomous agents may develop a joined shared repertoire of grammatical conventions. The grammar includes an emergent internal meta-level ontology which is exploited for tightening the grammatical constraints. To illustrate the latter, it is shown how a prototypical form for members of a syntactic category may gradually emerge, thus making it easier to guess to which syntactic category an unknown form belongs.

## 1 Introduction

There is a growing body of work exploring the idea that language can be viewed as a complex adaptive system [8], [4]. The research explores formal models of language use in evolving inhomogeneous populations through computational simulations and experiments with physical robotic agents. Shared linguistic conventions have been shown to emerge as attractors of a language dynamics operating over autonomous distributed agents. Each interaction, or language game, involves a speaker and a hearer randomly drawn from a population. The speaker attempts to produce an utterance and the hearer attempts to parse it. When the interaction fails, the agents construct or change aspects of their language competence (if they play the role of speaker) or acquire parts they were missing (if they play the role of hearer). The power of the agent's cognitive architecture and the construction and acquisition operators they employ determine the complexity of the linguistic structures that will emerge in the simulations.

The more the dynamics incorporates characteristics of human physiology or cognitive capacity, and the more the possible universes of discourse approach the characteristics of human environments, the closer the attractors resemble human natural languages.

Different aspects of language can be studied by varying the nature of the game, the cognitive architecture of the participating agents, and their internal behaviors. For example, the origins of sound repertoires may be studied by setting up imitation games in which the agents come equiped with a synthetic articulator, modeled after the vocal tract, an auditory module modeled after the human hearing system, and an associative memory [2]. The origins of lexicons may be studied by setting up naming games, in which the agents name a topic in terms of features which distinguish the topic from other items in the context [7]. The origins of meaning may be studied through discrimination games in which agents attempt to distinguish items in their environment using an evolving hierarchial repertoire of categorisers operating over their incoming sensory data streams [9]. These different games may be coupled to have a co-evolutionary dynamics.

This paper contributes to the study of grammar within the complex adaptive systems framework. The first work in this area [3] has shown that structured syntactic representations may emerge in a group of agents which take turns in parsing and generating. The resulting grammars generate purely formal structures but do not relate them with meaning. Subsequent work has focused on the problem how syntactic structure may be used to express meaning. Computational experiments in this line, particularly as reported by [5], [1], and [8], have shown that hierarchical structure starts to develop as soon as agents try to reuse in some way earlier created form-meaning associations. This can either be because existing form-meaning chunks are assembled in larger wholes, as emphasised in [8], or because an existing chunk is decomposed into subparts, emphasised in [5].

The main contribution of the work reported in the present paper is to show how an internal emergent linguistic ontology can evolve alongside the grammar. The ontology introduces a meta-level allowing the agents to formulate and process representations about the language itself. The ontology makes it possible to tighten the constraints imposed by the grammar on utterances. It makes the language more streamlined and thus eases the parsing and production process. It makes the language more predictable and thus more easy to learn.

In contrast to nativist approaches, which assume that linguistic ontologies are innate, this paper explores the hypothesis that ontologies emerge as part of the natural process of language genesis. This implies that individual language users must construct their own ontology in agreement with the ontology underlying the language in their community.

Two observations justify an evolutionary approach to linguistic ontology. First, linguistic ontologies appear to be to a large extent language-specific, even though there are universal tendencies. This explains perhaps why linguistics

has not been able to arrive at a universally agreed upon terminology. For example, many syntactic categories fundamental to Indo-European languages, like articles, adjectives or prepositions, do not occur at all in non-European languages, like Chinese. And even closely related languages within the same family may make very different fine-grained categorial distinctions. Thus French has about fifteen adjectival classes, according to distributional criteria, which are not shared by other Romance language [?].

Second, research in diachronic linguistics, particularly in grammaticalisation phenomena [13], has shown that a language's ontology keeps changing as part of language evolution. For example, the syntactic category auxiliary (with members like will, can, would, etc.) apparently emerged only in middle English [?]. Before that period, the auxiliaries had the same syntactic constraints as regular verbs.

This paper is necessarily focused on only key aspects of the broad problem of the emergence of linguistic ontologies and their use to support systemic growth in a language. The next section of the paper sets the stage by introducing an associative grammar formalism and related parsing and production processes. Next, language construction and acquisition operators are proposed which agents can use to repair or adapt their grammars as they engage in language games. These operators lead to an ontology for syntactic and semantic categories and for syntactic and semantic functions. The final part of the paper shows one example how this ontology is then used to increase the systematicity in the grammar. New elements of a category are constructed by the agents so as to resemble existing members, thus gradually constituting a 'prototypical form' for that category and providing a strong hint to the hearer which novel item a category belongs to.

# 2 Associative grammar

## 2.1 Meaning and Form

We will study in this paper the origins of grammar in a purely formal way. It is assumed that the agents have an open-ended repertoire of forms, denoted as $f1$, $f2$, etc. and an open-ended repertoire of meanings, denoted as $m1$, $m2$, etc. The forms correspond to syllables, orderings, stress patterns, intonation contours, or any other kind of formal device a language may recruit. The meanings may correspond to properties, relations, identifiers, or any other meaningful element that may have to be communicated. As discussed in other papers, see particularly [8] and [11], we have been conducting experiments in which the meanings are also emergent and perceptually grounded in the experiences of the agents.

Both forms and meanings are considered in this paper to be conjunctions of atomic propositions. To get closer to the complexity of natural languages

it is undoubtly necessary that they are first-order formulae. For example, to describe the phrase "the woman", as pronounced with rising intonation, there might be a conjunction of form elements as follows:

```
word (x1,"the") & word (x2,"woman") &
preceeds (x3,x1,x2) & intonation (x3,rising)
```

In this paper, such an expression would simply be written down as ($f1$ $f2$ $f3$ $f4$).

## 2.2 Representation of the Grammar

The grammar consists of a set of associations between form and meaning schemata, and is therefore most strongly related to cognitive grammar [6]. An associative grammar is a transducer from form to meaning or from meaning to form, rather than a generator of all possible forms. The associations are bi-directional.

Two notations are employed. The *associational notation* shows the form / meaning pairs and the categories associated with the form and the meaning. An example of an association relating a form schema with elements $f1$ $f2$ to a meaning schema with elements $m3$ $m4$ $m5$ is written down as follows:

```
assoc1. [f1 f2] / [m3 m4 m5]
        cat1, cat11 / cat3, cat12
```

$Cat1$ and $cat11$ are syntactic categories to which the form-schema of association $assoc1$ belongs. A schema may belong to more than one category. $Cat3$ and $cat12$ are the semantic categories of the meaning-schema. Associations like $assoc1$ involving only basic meaning and form elements but no internal structure are equivalent to lexicon entries. There is however no strict distinction between lexicon and grammar, neither in the representation of the grammar nor its use during parsing and producing.

An example of an association relating structured schemata is as follows:

```
assoc2. [[slot1 cat1] [slot2 cat2]] / [[slot3 cat3] [slot4 cat4]]
         top, cat10 / top, cat5
```

$slot1$ and $slot2$ are syntactic slots to be filled by members of the syntactic categories $cat1$ and $cat2$. $slot3$ and $slot4$ are semantic slots to be filled by members of the semantic categories $cat3$ and $cat4$. $top$ is a special category denoting that the structure covers a complete utterance. The syntactic schema can fill a slot requiring an element of $cat10$, the semantic schema one requiring an element of $cat4$. Both can cover a complete utterance.

The ontological types implied by this grammar relate in the following way to traditional linguistic ontologies. The syntactic categories correspond to notions like Noun, Adjective, Noun Group, Verb Group, etc. They categorise parts of the form of an utterance. The syntactic slots correspond to grammatical functions or relations, such as Determiner, Adjunct, Head, etc. They are names of

possible slots in syntactic structures that may be filled by members of syntactic categories. The semantic categories label building blocks of semantic structures such as identifier, situation, property, connective, quantifier, etc. Semantic slots correspond to semantic roles or functions. An example would be case roles for situation schemata, like agent or patient.

To communicate the grammar to linguists used to a generative grammar formalism, a *generative notation* is derived from the set of associations. Each association is then written as a rewrite rule with the number of the rule equal to the number of the association. Each rule rewrites the categories associated with a schema to the basic elements or the slots' categories. This can be done either for the syntactic part of each association or for the semantic part. The syntactic and semantic rules from the associations above are as follows:

```
1. cat1, cat11 -> f1 f2
2. top, cat10 -> cat1 cat2
3. cat2 -> f100
```

```
1. cat3,cat12 -> m3 m4 m5
2. top, cat5 -> cat3 cat4
3. cat4 -> m7 m8
```

Note that the generative notation contains less information than the associative notation. The generative notation does not name the slots, and, more importantly, does not show the two-way mapping from syntactic to semantic schemata. The grammar is not used in a generative mode neither for parsing nor for producing.

During language processing complex internal structures are created which are printed out as a standard constituent tree in parenthesised notation. Again much information is left out but such representations are easy for us to follow what is going on.

```
(top
  (slot1 (cat1 f1 f2))
  (slot2 (cat2 f100)))
```

```
(top
  (slot3 (cat3 m3 m4 m5))
  (slot4 (cat4 m7 m8)))
```

## 2.3 Parsing and Producing

Parsing and production are mirror processes. Parsing takes a set of forms as input and constructs a syntactic structure by cycling through the grammar. Each time a part of the syntactic structure is built, the corresponding semantic structure is built. Production takes as input a set of meanings and constructs a semantic structure by cycling through the associations in the grammar. Each time a part of the semantic structure is built, the related syntactic structure is built as well. Ordering is a formal device as any other one. It is described explicitly (using a predicate like *preceeds*). The order in which the form elements are given does not play a role at all in processing.

In a complementary mode of the parser, there is both a form and an expected meaning. The form *and* meaning schema of an association must then both

match for a successful application of a schema. In case of incompleteness of the grammar the expected meaning is used to repair the grammar.

It often happens that only part of a schema matches with the given input. In that case, a *promissory note* is constructed, which captures the information that a structure is still to be expected. When it arrives, it is incorporated in the structure already built and the promissory note is removed. The promissory note is used for repairing the grammar when a structure is not found at the end of processing.

In the experiments conducted so far, both parsing and production are assumed to be deterministic and the grammar is assumed to evolve in such a way as to keep this property intact. Determinism is only possible when there is at all times enough information to decide whether an association is applicable. Because semantic structures are built at the same time as syntactic structures, semantic constraints can be exploited early. When this is not the case, constraints on the grammar should be tightened. Additional formal elements could be employed to help disambiguate the utterance or the constraints on the schema could be tightened by restricting the scope of a category. Note that deterministic parsing does not require that the grammar is deterministic in a generative sense. It is allowed that a certain non-terminal symbol is rewritten in more than one way or that the same structure occurs as the right hand side of more than one rule.

# 3 Construction and Acquisition

The key of the system resides in the operators performing construction and acquisition of the grammar. This section presents some example operators and their application during construction and acquisition illustrated with traces taken from the computer simulations. Situations may arise which are too complex. Rather than attempting a fix, the speaker or hearer then waits until simpler cases allow a progressive build up of the parts.

## 3.1 Lexicalisation

*Construction* When a set of meanings is given and no schema is matching at all, a new form is constructed by the speaker and a new association between the meanings and the form is added to the grammar. The category of the form and meaning schemata is *top*. Such a lexicalisation operation has happened in the following example. Agnt-1 attempts to express the meaning set ($m1$). The EXPAND task which performs grammar lookup does not yield any result. A REPAIR task is triggered which constructs a new form ($f3\ f2\ f1$) associated to $m1$.

```
Produce agnt-1: (m1)
Next task: EXPAND
```

```
Next task: REPAIR
 <= New association assoc1 in agnt-1: [f3 f2 f1] / [m1]
```

*Acquisition* When the hearer receives a form and no schema is matching, he performs the dual of the lexicalisation operation by constructing a new association between the given forms and the expected meaning set. This is shown in the following trace:

```
Parse agnt-2: (f3 f2 f1) (m1)
Next task: EXPAND
Next task: REPAIR
<= New association assoc2 in agnt-2: [f3 f2 f1] / [m1]
```

## 3.2 Grouping

*Construction* When a speaker has been able to cover the set of meanings, but the final structure has disconnected parts, he constructs a new schema which combines these parts. The new schema has slots for each of the parts being combined. New categories are introduced for the slot fillers and the parts are categorised in terms of the new categories. This grouping operation is illustrated in the next example. ($m1$) and ($m2$ $m3$ $m4$) are both individually covered but the total is not covered yet.

```
Produce agnt-1: (m1 m4 m3 m2)
Next task: EXPAND
 =+= Meaning Match with assoc3
  Added form: (f5 f4) Covered meaning: (m2 m3 m4)
 =+= Meaning Match with assoc1
  Added form: (f3 f2 f1) Covered meaning: (m1)
Next task: REPAIR
  Syntactic structure:
   (top f3 f2 f1)
   (top f5 f4)
  Semantic structure:
   (top m1)
   (top m4 m3 m2)
<= Categorise assoc3: (f5 f4) as Cat1 - (m2 m3 m4) as Cat3
<= Categorise assoc1: (f3 f2 f1) as Cat2 - (m1) as Cat4
<= New association assoc5 in agnt-1:
  [[slot1 Cat1] [slot2 Cat2]]/[[slot3 Cat3] [slot4 Cat4]]
```

The final structures after repair are as follows:

```
  Syntactic structure:
   (top
    (slot2 (Cat2 f3 f2 f1))
```

```
    (slot1 (Cat1 f5 f4)))
  Semantic structure:
   (top
    (slot4 (Cat4 m1))
    (slot3 (Cat3 m4 m3 m2)))
```

The grammar in associative notation is now like this:

```
assoc5
  [[slot1 Cat1] [slot2 Cat2]] / [[slot3 Cat3] [slot4 Cat4]]
  top / top
assoc3
  [f5 f4] / [m4 m3 m2]
  top cat1 / top Cat3
assoc1
  [f3 f2 f1] / [m1]
  top Cat2 / top Cat4
```

The form and meaning grammar in generative notation is like this:

```
5. Top -> Cat1 Cat2                    5. Top -> Cat3 Cat4
3. Top, Cat1 -> f5 f4                  3. Top, Cat3 -> m4 m3 m2
1. Top, Cat2 -> f3 f2 f1               4. Top, Cat4 -> m1
```

*Acquisition* A similar operation is performed by the hearer, when his final structure consists of disconnected parts. Moreover the grouping operator can be applied on arbitrary complex parts and combining an arbitrary number of parts.

### 3.3 Reuse

*Construction* When a structure is partially matching, and another structure is left disconnected, the disconnected structure may be recategorised to fit within the partial match. As a general rule the structure is attached as low as possible in the hierarchy. This operation is illustrated in the following example, where an attempt is made to express ($m1$ $m5$). $m1$ and $m5$ are both already lexicalised. $m1$ fits in $slot4$ of $assoc5$ and a promissory note is constructed because the syntactic slot $slot1$ and the semantic slot $slot3$ are not filled.

```
Produce agnt-1: (m1 m5)
Next task: EXPAND
 =+= Meaning Match with assoc7
  Added form: (f8 f7 f6)
  Covered meaning: (m5)
 =+= Meaning Match with assoc1
  Added form: (f-3 f-2 f-1)
```

```
  Covered meaning: (m1)
 =+= Dual Partial Match with assoc5
  Covered Form: ((slot2 (assoc1 f-3 f-2 f-1)))
  Covered Meaning: ((slot4 (assoc1 m1)))
  Missing: ((slot1 Cat1)) ((slot3 Cat3))
```

The grammar is repaired by categorising [$m5$] as belonging to $cat3$ and [$f8$ $f7$ $f6$] as belonging to $cat1$:

```
Next task: REPAIR
<= Categorise assoc7: [f8 f7 f6] as Cat1 - [m5] as Cat3
```

The form grammar in generative notation is now as follows.

```
7. Top, Cat1 -> f8 f7 f6
5. Top -> Cat1 Cat2
3. Top, Cat1 -> f5 f4
1. Top, Cat2 -> f3 f2 f1
```

*Acquisition* A similar operation is performed by the hearer when confronted with a part that is missing and another part that could not be fit in. He recategorises so that the disconnected part can fill the missing slot.

## 3.4 Recursive Application

Formation, grouping, and reuse operators might have to be applied in turn to fully repair a structure. When the number of repairs is too complex, the agent ignores the case. When several alternative repairs are available, the agent will attempt to attach the disconnected structure as low as possible in the hierarchy.

Here is a more complex example, starting from a grammar with the following generative representation of its meaning structures:

```
22. TOP, Cat11 -> Cat27 Cat28
19. TOP, Cat28 -> m13 m12
16. TOP -> Cat19 Cat20
12. TOP, Cat20 -> Cat11 Cat12
7. TOP, Cat3 -> m6 m5 m4
5. TOP, Cat12 -> Cat3 Cat4
3. TOP, Cat3 -> m3
1. TOP, Cat4 -> m2 m1
21. Cat27 -> m14 m15 m16
15. Cat19 -> m11
11. Cat11 -> m10 m9
9. Cat3 -> m7 m8
```

The goal is to produce an utterance expressing

```
(m17 m16 m15 m14 m8 m7 m2 m1)
```

In a first phase, the agent tries to cover the given meaning set by successive application of his grammatical associations.

```
Produce agnt-1: (m17 m16 m15 m14 m8 m7 m2 m1)
Next task: EXPAND
 =+= Meaning Match with assoc21
  Added form: (f19 f18 f17) Covered meaning: (m16 m15 m14)
 =+= Meaning Match with assoc9
  Added form: (f9) Covered meaning: (m8 m7)
 =+= Dual Partial Match with assoc5
  Covered Form: ((slot1 (assoc9 f9)))
  Covered Meaning: ((slot3 (assoc9 m7 m8)))
  Missing: ((slot2 Cat2)) ((slot4 Cat4))
 =+= Meaning Match with assoc1
  Added form: (f3 f2 f1) Covered meaning: (m1 m2)
  Insert slots in assoc5:
     ((slot2 (assoc1 f3 f2 f1)))
     ((slot4 (assoc1 m2 m1)))
 =+= Dual Partial Match with assoc22
  Covered Form: ((slot25 (assoc21 f19 f18 f17)))
  Covered Meaning: ((slot27 (assoc21 m14 m15 m16)))
  Missing: ((slot26 Cat26)) ((slot28 Cat28))
 =+= Dual Match with assoc12
  Covered form:
    ((slot10 (assoc5 (slot2 (assoc1 f3 f2 f1))
                     (slot1 (assoc9 f9))))
     (slot9 (assoc22 (slot25 (assoc21 f19 f18 f17)))))
  Covered meaning:
    ((slot12 (assoc5 (slot4 (assoc1 m2 m1))
                     (slot3 (assoc9 m7 m8))))
     (slot11 (assoc22 (slot27 (assoc21 m14 m15 m16)))))
 =+= Dual Partial Match with assoc16
  Covered Form:
  ((slot18
    (assoc12 (slot10 (assoc5 (slot2 (assoc1 f3 f2 f1))
                             (slot1 (assoc9 f9))))
             (slot9 (assoc22 (slot25 (assoc21 f19 f18 f17)))))))
  Covered Meaning:
  ((slot20
    (assoc12
     (slot12 (assoc5 (slot4 (assoc1 m2 m1))
                     (slot3 (assoc9 m7 m8))))
     (slot11 (assoc22 (slot27 (assoc21 m14 m15 m16)))))))
```

```
Missing: ((slot17 Cat17)) ((slot19 Cat19))
```

Note how the promissory note resulting from a partial match of $assoc5$ has later been resolved by a match with $asssoc1$, which yielded a structure capable to fill the missing slots. After all possible associations have been applied (which might imply several runs through the grammar), the semantic structure still contains a meaning element $m17$ which is not covered by any schema:

```
(TOP
 (slot20
  (Cat20 (slot12 (Cat12 (slot4 (Cat4 m2 m1))
                        (slot3 (Cat3 m7 m8))))
         (slot11 (Cat11 (slot27 (Cat27 m14 m15 m16)))))))
 m17
```

There are two incomplete substructures: the top based on assoc16, which has a filler for $slot20$ but not for $slot19$, and the filler of $slot11$ ($cat11$) which has a filler for $slot27$ but not for $slot28$.

Repair starts by lexicalising the uncovered meaning (using the lexicalisation operation) and by categorising the result so as to fill the lowest incomplete node in the hierarchy ($cat9$ in the syntactic structure and $cat11$ in the semantic structure):

```
<= New association assoc25 in agnt-1: [f21 f20] / [m17]
<= Categorise assoc25: [f21 f20] as Cat26 - [m17] as Cat28
```

This repairs the problem with the $cat11$ structure, but not with the top level node which has an unfilled slot. However since the filler of $slot18$ can itself be a top-node (it is based on $assoc12$), the repair consists in chipping off the top level. The final structures are then:

```
Syntactic structure:
 (TOP
  (slot10 (Cat10 (slot2 (Cat2 f3 f2 f1))
                 (slot1 (Cat1 f9))))
  (slot9  (Cat9 (slot26 (Cat26 f21 f20))
                (slot25 (Cat25 f19 f18 f17)))))
Semantic structure:
 (TOP
  (slot12 (Cat12 (slot4 (Cat4 m2 m1))
                 (slot3 (Cat3 m7 m8))))
  (slot11 (Cat11 (slot28 (Cat28 m17))
                 (slot27 (Cat27 m14 m15 m16)))))
```

There are several other operators needed. For example, to combine two schemas with a common element into a single one, to break up a schema when only part is matching, etc.

# 4 Systemic Growth

As more and more meanings are to be expressed, the construction and acquisition operators gradually generate a more complex grammar, in the sense of an increasing number of categories, and a growing set of rules covering trees of increasing depth and width. But so far, these mechanisms do not yet exploit the added value of the categorisation. This can only happen if the agents develop and exploit constraints on what can be a member of a category. In the case of syntactic categories, these are constraints on form, called syntactic features in linguistic theory. In the case of semantic categories, these are constraints on the meaning characteristics of its members. These semantic features act as selection restrictions on the applicability of a meaning schema.

The constraints feedback on the grammar formation process in two ways. When new members of a category are constructed by a speaker they are made to satisfy the syntactic category constraints so that the hearer can make a reasonable guess what the category of a new form might be. In the parsing and production process, semantic and syntactic features are used to tighten and thus optimise the matching process. As the constraints on a slot-filler become tighter, the possible situations matching with an association become more restricted, possibly leading to the formation of new schemata, thus creating new hierarchy and hence a spiral of increased complexity. Due to space limitations, this paper does not elaborate on this second feature.

## 4.1 Category Profiles

It is well known that linguistic categories (like natural kinds in general) cannot be defined in terms of strict necessary and sufficient conditions [12],[6]. For example, many adverbs in English end in '-ly' (slowly, quikly, steadily, etc.) but there is no general rule that says that they all have to do so. Many nouns are based on classifying objects (table, man, mouse) but they need not be. In addition, there is enormous flexibility to recategorise temporarily elements of certain categories in order to fit other categories. Constraints can always be overridden.

To represent the prototypical characteristics of a category, agents build up a *category profile*. It represents the frequency of each descriptive element in the examples seen so far. Thus, if the members of a syntactic category are:

```
[f1 f2 f3 f4]
[f5 f3 f1 f6]
[f1 f2 f5]
```

Then the frequency of $f1$ is 1.0 because it occurs in all cases. The frequency of f2 is 0.66 because it occurs in 2 out of 3 cases. The profile of a category is easy to maintain. Every time a new member is seen, the counters for each of

the possible descriptive elements are updated and the frequency of an element is equal to the number of occurences divided by the total number of cases seen.

A category profile is used in two ways. First it is used to construct new members by considering the frequencies as probabilities that the element occurs in the new example being constructed, with some additional random creation to ensure that new cases are different. Some members of a category are not constructed by the agent himself and therefore may deviate substantially from the agent's category profile, even though they satisfy the profile of the other agent. There is a hidden positive feedback process which gradually leads to increased order and coherence in the population: When an agent detects regularity (even though the regularity may be completely unintentional) he uses this regularity to create new examples which cause others to see more regularity, leading to even more regularity, etc.

Second, a category profile is used to classify a case as belonging to a category by calculating the match between the case and the profile. The frequencies are then interpreted as indicating the weight with which a description should be part of the case. The match is decomposed into gain and loss. When a description in the profile is part of the case, the gain goes up with an amount equal to the frequency of the element in the profile. When it is not, the loss goes up with the same amount. A case is classified as belonging to the category to which it has the smallest distance (highest gain, least loss).

## 4.2 Integration in the Language Game

Here is now an example how the category profile mechanism and the systematicity it generates are used. A new form needs to be created for covering ($m20$ $m11$ $m5$ $m5$) such that this form fits within the missing slot $slot19$ of $assoc25$:

```
Produce agnt-1 (m5 m5 m11 m20 m20 m4 m2 m3 m8)
Next task: EXPAND
 =+= Meaning Match with assoc1
  Added form: (f10 f9 f7)
  Covered meaning: (m8 m3 m2 m4 m20)
 =+= Dual Partial Match with assoc25
  Covered Form: ((slot21 (assoc1 f10 f9 f7)))
  Covered Meaning: ((slot41 (assoc1 m20 m4 m2 m3 m8)))
  Missing: ((slot19 Cat21)) ((slot31 Cat31))
```

The following forms already exist as members of $cat21$, listed in the order of creation. Each form was created based on the profile of the category so far. The numbers before each form indicate the match (gain and loss) of the new example.

```
1.000,0.000, [f13 f19 f20 f11 f12]
```

```
[m20 m11 m5 m5] as Cat3
```

# 5 Conclusions

The paper focused on two issues in the emergence of grammar. The first issue is how hierarchy may emerge. A set of operators has been introduced by which agents progressively construct their grammar, either by extending it when they need to express something they cannot yet express or by learning from other agents how they express novel meanings or meaning combinations. Hierarchy emerges from reuse and syntactic and semantic categories arise as a side effect of hierarchy. A new category is created to capture what elements can fill a slot and elements are categorised or recategorised based on whether they are seen to occur in certain slots.

The second issue is how categorisation can be exploited to increase the systematicity of the grammar, something which is clearly seen in natural language. A prototype-based approach was presented in which an evolving profile is built up for each category. The profile is used to influence the construction of new members of a category and to tighten the matching between a schema and a new case.

The next step in this research is to see in how far additional constraints can be introduced to make the grammar closer to natural language grammars. This requires several steps. First of all a richer (first-order) description language needs to be adopted for form and meaning elements and consequently the parsing and production processes will have to be made more complex. Second, more sophisticated construction and acquisition operators will have to be introduced. Third, the meaning sets need to be constrained further to reflect the natural constraints on a typical human universe of discourse.

# References


[1] Batali. J. (1998) Computational Simulations of the Emergence of Grammar. In: [4].

[2] De Boer, B. (1997) Emergent Vowel Systems in a Population of Agents. In Harvey, I. et.al. (eds.) *Proceedings of ECAL 97*, Brighton UK, July 1997. The MIT Press, Cambridge Ma.

[3] Hashimoto, T. and T. Ikegami (1996) Emergence of net-grammar in communicating agents. BioSystems 38 (1996) 1-14.

[4] Hurford, J., C. Knight and M. Studdert-Kennedy (eds.) (1998) *Evolution of Human Language.* Edinburgh Univ. Press. Edinburgh, 1998.

[5] Kirby, S. (1998) Language Evolution without Natural Selection. From vocabulary to syntax in a population of learners. Evolution of Language Conference, London.

[6] Langacker, R. (1986) *Foundations of Cognitive Grammar.* Stanford University Press, Stanford, 1986.

[7] Lightfoot, D. (1979) *Principles of Diacrhonic Syntax.* Cambridge University Press, Cambridge.

[8] Steels. L. (1996) Self-organising vocabularies. In Langton, C. (ed.) *Proceedings of Artificial Life V.* Nara, 1996.

[9] Steels, L. (1997) The synthetic modeling of language origins. *Evolution of Communication*, 1(1):1-35.

[10] Steels, L. (1997b) Constructing and Sharing Perceptual Distinctions. In van Someren, M. and G. Widmer (eds.) *Proceedings of the European Conference on Machine Learning*, Prague, April 1997. Springer-Verlag, Berlin, 1997.

[11] Steels, L. (1997c) The origin of syntax in visually grounded robotic agents. In: Proceedings of IJCAI-97, Morgan Kauffman Pub. Los Angeles.

[12] Steels, L. and P. Vogt (1997) Grounding adaptive language games in robotic agents. In Harvey, I. et.al. (eds.) *Proceedings of ECAL 97* The MIT Press, Cambridge Ma., 1997.

[13] Taylor, J. (1995) Linguistic Categorization - Prototypes in Linguistic Theory. Oxford University Press, Oxford.

[14] Traugott, E. and B. Heine (1991) *Approaches to Grammaticalization. Volume I and II.* John Benjamins Publishing Company, Amsterdam, 1991.

[15] Vennemann, T. (1972)