The Role of Construction Grammar in Fluid Language Grounding

Luc Steels[1,2]

(1) Vrije Universiteit Brussel Artificial Intelligence Laboratory
(2) SONY Computer Science Laboratory - Paris.

Abstract

The paper examines in how far Construction Grammar is a useful foundation by which artificial agents can self-organise communication systems that are grounded in the real world through a sensori-motor embodiment and use grammar to express certain aspects of meaning. It proposes a particular computational formalism, Fluid Construction Grammar, and mechanisms by which constructions can be progressively built up and shared by agents as they engage in verbal interactions about real world scenes.

Keywords: Symbol grounding, Fluid Construction Grammar, multi-agent learning, semiotic dynamics, Embodied AI, abductive learning.
[paper in press for AI Journal, vol 164 - to appear end 2005]

1 Introduction

The symbol grounding problem is concerned with the question how autonomous agents might be able to relate symbols to the world through sensing and categorisation, either for external use, for example for communication, or for internal use, for example as the basis of episodic memory or rational problem solving. It is not only crucial that the symbols are systematically related to the world by way of perceptually grounded categories but also that the agents themselves (and not human designers or teachers) establish the categories and their symbolisation through learning and mutual coordination. The symbol grounding problem is known to be a key problem in artificial intelligence [15],[20] and requires the integration of many competences, from machine vision to syntactic processing and machine learning.

In this paper, we take (natural) language as the prototypical example of a symbol system, not only because it is very rich, but also because a lot is already known about the structure of natural language and how it is processed and learned. On the other hand, our goal is not to acquire an existing natural language but to understand how artificial agents could invent their own communication system with natural language like features, including grammar.

Although most research in computational linguistics and AI has focused on ungrounded language processing, there have been a few earlier attempts to link language to the real world. The earliest one involves a robot Shakey built at SRI in the early seventies [25]. Shakey was able to accept English-like commands and execute them in its real world environment consisting of a room with objects of different shapes. Shakey was one of the first robots with symbol grounding, in the sense that its actions and internal world models were anchored through sensors and actuators to the real world, and natural language input and output as well as the planning of actions made use of these internal world models, but the ontology, perceptual grounding, and language protocols were entirely hand coded.

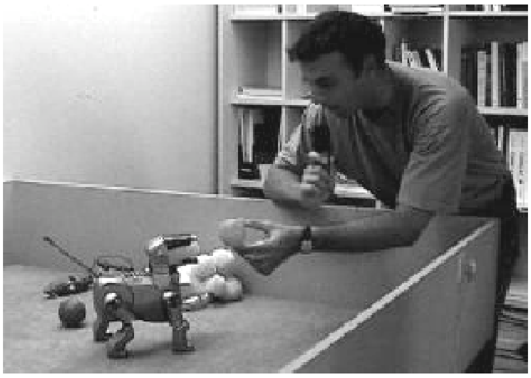

Fig. 1. Experiments in language grounding, where an autonomous robot (the Sony AIBO) learns meanings of words and the categories underlying them by playing language games with a human.

More recently, there has been a growing number of efforts to achieve grounded language systems where the robot acquires perceptually grounded categories and words and sentence constructions to symbolise aspects of the world. We reported one example [37] in which a human interacting with an autonomously moving AIBO robot acquired words and their perceptually grounded meanings by playing situated language games (see figure 1 from [37]). Another example, reported by Roy [28], presents a robot equiped with a camera that can inspect objects from different points of view. A human experimenter puts an object in front of the robot and then gives a description in spoken English. The robot has to learn both the perceptual categories (which is done using a memory-based approach to vision) and the relation between the categories and language. A similar approach has been discussed by Dominey [12], which focused on the acquisition of descriptions of actions. These experiments show that learning, as opposed to hand coding, is becoming a viable approach to the symbol grounding problem, but the kinds of grammars being considered are rather ad hoc and limited, and only single agents are considered, which learn an already existing language from a human.

In earlier papers [31], we have shown already how large populations of agents can autonomously develop shared lexicons and ontologies. The agents play language games in which one agent (the speaker) draws the attention of the hearer to a chosen object in the world (the topic). Agents develop new categorisations to be successful in discriminating the topic, invent new names for these categories, and align the associations between names and categories based on success or failure in the game. The positive feedback inherent in the agents' behavior causes the communication system and its underlying ontology to be progressively shared among all agents in the population, and new agents entering the population adapt to the system in place. This kind of progressive alignment has also been observed in human communications [14]. Other similar experiments with broader and more open-ended perceptual sources have been reported in [36], but they all remain at the level of lexical symbol systems, i.e. symbol systems in which where grammar does not play a role.

The present paper does not focus on the grounding of the world model, although the examples discussed in the paper are all based on an event recognition system that achieves this [34], nor on the question how a shared lexicon that names the categories used in a world model can self-organise (a problem which we consider to be solved [31]), but rather on the question in what way grounding can play a role in grammar construction and what kind of grammatical framework is most suited for this purpose. We will argue that Construction Grammar [11] and its associated learning framework, known as Constructivist Learning [39], look promising candidates. The major contribution of the paper is to report progress in testing this hypothesis.

2 Construction Grammars

2.1 Linguistic Perspective

Research in cognitive linguistics [21],[38] has emphasised that grammar strongly interacts with meaning and communicative function, and recent work in Construction Grammar [11] has produced an increasing number of examples of how this interaction shows up in natural language. Construction grammar puts bidirectional mappings between abstract semantic frames such as CAUSE-MOVE+cause+goal+theme and syntactic patterns like Subject+Verb+Object+Oblique at the core of the grammar (see figure 5), instead of defining syntax independently of semantics with derivational rules. Within the same cognitive linguistics tradition, human embodiment has been argued to play a signficant role in the fundamental grammatical structure of language [22], and recent work on Embodied Construction Grammar [2] and the 'mental simulation' of actions expressed in verb frames [13] give concrete ideas how an embodied cognition perspective applies to language. The perspective adopted in this paper is along the lines of this cognitive linguistics tradition by seeking a tight interaction between semantic grounding and the structure and processing of language and by using the sensori-motor grounding as a source for constraining the invention and learning of semantic categorisations.

The primary motivation why we want to investigate Construction Grammar for language grounding, lies in its ability to capture significant generalisations about the syntax-semantics interaction which are more difficult to express in other approaches [19],[23]. A second motivation comes from recent observations of children's language acquisition made by Tomasello and colleagues [39]. In contrast to the Chomskyan principles and parameters framework in which most of the structure of grammar is already given and only some parameters need to be set [26], Tomasello, et.al. have argued that language acquisition proceeds in a 'constructivist' manner: "Children acquire linguistic competence (...) only gradually, beginning with more concrete linguistic structures based on particular words and morphemes, and then building up to more abstract and productive structures based on various types of linguistic categories, schemas, and constructions." [39], p. 161.

This constructivist approach assumes that language development is (i) grounded in cognition because prior to (or in a co-development with language) there is an understanding and conceptualisation of scenes in terms of events, objects, roles that objects play in events, and perspectives on the event, and (ii) grounded in communication because language learning is intimately embedded in interactions with specific communicative goals. Chang and colleagues [6] have recently presented computer simulations of such learning processes based on empirical data of child language acquisition and Bayesian learning mechanisms and the perspective adopted here is along similar lines although we are exploring an abductive learning approach.

Our third reason for exploring construction grammar builds on Hopper's notion of Emergent Grammar [17]), which argues that language learning is never finished because language itself is constantly on the move as new words and constructions propagate in the population and partners in dialog align their modes of expression [8]. Seen over a longer time scale, this leads to the grammaticalisation phenomena observed by diachronic linguists [16]. The notion of fluidity is intended to capture this. Grammar (and consequently language) is fluid when some aspects of how meaning is being expressed are not yet conventionalised and so comprehension heavily depends on situated interaction and context. The issue of grammatical correctness is secondary as agents try above all to make sense of each other's utterances whether they are grammatical or not. At the same time, unconventionalised meaning expression can become conventionalised or conventions may erode to be replaced by others. So language is forever 'in the making'.

2.2 The Description Game

Since the early work on language understanding and production, a distinction has been made between an agent's world model, which is linked to the real world through sensing and categorisation, and the meanings that are symbolised in a particular language sentence (figure 2). Meanings are a subset of the facts found in the world model with additional information on how these facts are to be used in communication. Conceptualisation is the process of selecting the meanings that will be expressed to achieve a communicative goal (which may require additional perception and categorisation), and interpretation is the process of mapping these meanings back into the world model and thus to the real world itself (which may again require additional inference or active perception to extend the world model). In the case of a population of agents, we assume that each agent has his own world model, but there must be sufficient 'common ground' to allow communication, for example because both partners have been looking at the same world scene.

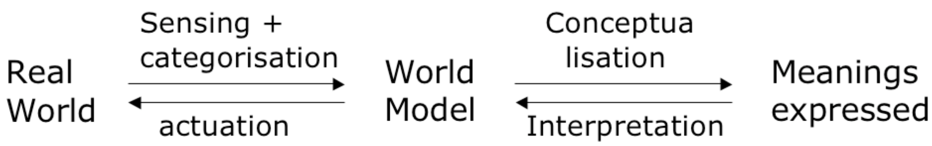

Fig. 2. There is a consensus that the grounding of language in the real world goes through the intermediary of a world model from which the meanings to be expressed are selected.

In order to make the discussion concrete and allow computational and robotic experimentation, this paper focuses on one specific type of verbal interaction: One agent (the speaker) describes an event that happened recently in the shared world environment, and another agent (the listener) attempts to see whether this event actually occurred. Such a 'description game' fails if the listener cannot make sense of the description of the speaker or if there is no event or more than one that satisfies the description. So it is not only important that the description is true for the event the speaker has chosen but also that it distinguishes this event from other events that happened in roughly the same time period. Both agents have their own internal world model, anchored through perception, categorisation, and action in the world, and there is no telepathy, so they can only coordinate their categories and modes of expression through verbal and non-verbal interaction.

The scenes that have been used for the experiments underlying the present paper are similar to those used in child language research, such as: A puppet called Jill slides a box to another puppet called Jack (see figure 3), Jill enters a house, Jack moves towards Jill, Jill pushes Jack towards the house, etc. Two robots that are about to play a description game first look through their camera at such a scene which is played by a human experimenter. The scene typicallly contains a dozen or so hierarchically structured actions. The agents then process the image stream to obtain a set of event descriptions at several layers of detail. The event descriptions that make up the world model are represented as time-stamped facts in first-order predicate calculus notation, following standard AI practice.

Fig. 3. The domain of the experiments reported in this paper use scenes enacted with puppets so that typical interactions between humans involving agency can be perceived and described. The scene in this figure describes an event where 'Jill' slides a block to 'Jack'.

The vision system that our robots use to transform real world visual images into the event structures in the world model is in itself very complex and described in more detail in [34]. It first delineates objects based on colour histograms, then groups the pixels belonging to the same object together. Starting from basic primitive relations like, touch, movement, appearance, etc. more complex event descriptions are assembled by event detection algorithms that use hierarchical probabilistic state transition networks for the recognition of actions. These networks will play a role later in the invention and acquisition of semantic categories used for constraining grammatical constructions. An example of such a network is shown in figure 4. It recognises a cause-move event in terms of different states, starting when two objects are not touching, then one is approaching the other, and then both move at the same time while remaining in touch. A slide-event like the one in figure 3 contains about twenty different states and actions with specific constraints on them, for example, the object that is being moved must remain in contact with a surface. Event recognition is unreliable due to complexities of the real world and multiple hypotheses are generated with different degrees of certainty, but saliency, confidence, general fit, and prior expectation coming from language (particularly when the agent plays the role of hearer) are used to make better hypotheses about what micro- and macro-events took place.

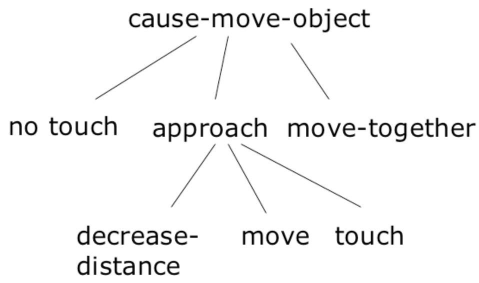

Fig. 4. A small part of the state transition networks that are used to recognise events in the scene.

Because the world consists of dynamically changing situations, classified as events, the object and event recognition system that maintains the world model is strongly related to other research on visual event classification [29], [18], temporal world modeling [1], and the conceptual analysis of event expression in natural language [38]. For the purposes of the present paper, the nature of the vision system is not critical and any other system could be used as long as it produces a stream of hierarchical event descriptions reflecting the events in the real world as experienced by the agent.

In our initial explanations, English-like words are used to make it easier to follow the examples. Simplifying to the extreme, let us assume that the world model contains (among thousands of others) the following facts as a result of perceiving and categorising the scene in figure 3:

> slide(ev1), slide-1(ev1,obj1), slide-2(ev1,obj2), slide-3(ev1,obj3), name(obj1,Jill), status(obj1,single-object), block(obj2), status(obj2,object-set), name(obj3,Jack), status(obj3,single-object)

There are three objects involved and one slide-event. The different roles of objects in the event are represented as separate predicates instead of arguments of a single slide-predicate: slide-1 (the one who is causing the sliding), slide-2 (the object involved), and slide-3 (the goal towards which the object is moved). We use English words for the names of predicates, instead of predicate-1, predicate-2, etc. just to make it easier to understand what is going on, but these names should of course not be confused with 'real' English words and the meaning of these predicates comes entirely from how they are grounded in the world and in language. A description in English of the scene captured in the facts above could be: “Jill slides blocks to Jack”. This example will be used in the rest of the paper to illustrate the nature of the grammar. Because we want to focus on grammar, many issues concerning the world model will be ignored. Thus, the examples given in this paper assume that speaker and hearer have the same world model, even though this is hardly ever the case for real world embodied agents and not assumed in our experiments.

2.3 Expressing equalities between variables

Because we want to understand how a population of agents invents a grounded grammar, we need to understand what triggers the invention of grammatical constructions, i.e. why grammar is needed. Let us assume that there is no grammar to start with. Each of the lexical items in the sentence “Jill slide(s) blocks (to) Jack” is making a contribution to the meaning that the listener is supposed to reconstruct. The word “Jill” conveys that there is an object, introduced with a variable ?obj1, predicated as having the name Jill, the object has the status of ‘single-object’ as opposed to a set of objects. The word “slide” conveys that there is an event ?ev1 with three roles. The word “blocks” conveys that there is a set of objects which is predicated as being a block. And the word “Jack” introduces yet another single object ?obj5. So just from decoding the words, the listener can reconstruct the following set of clauses:

> name(?obj1,Jill), status(?obj1,single-object), slide(?ev1), slide-1(?ev1,?obj6), slide-2(?ev1,?obj2), slide-3(?ev1,?obj3), block(?obj4), status(?obj4,single-object), name(?obj5,Jack), status(?obj5,single-object)

Following standard approaches to semantic interpretation, we can view these clauses as constraints on the objects and events that are being described, and the interpretation process can therefore be viewed as matching the clauses against the facts in the world model to obtain a series of bindings:

> ((?obj1 . obj1)(?ev1 . ev1)(?obj6 . obj1) (?obj2 . obj2) (?obj3 . obj3)(?obj4 . obj2)(?obj5 . obj3))

By definition, the description game is successful if the listener is able to find a unique set of bindings for all the variables.

Based on the lexicon only, the listener cannot know that ?obj6, to be bound to the entity who is causing the block to slide, is the same as ?obj1, predicated a having the name Jill, or that ?obj2, the object being moved in the slide-event, is the same as the block ?obj4. This only becomes clear when the clause is matched by the listener against the world model he constructed based on visual scene analysis. If two variables are constrained to be bound to the same object, we will say that there is an equality between these variables. In this example, we have the following equalities:

(?obj1 = ?obj6) (?obj2 = ?obj4) (?obj3 = ?obj5)

Knowing these equalities before attempting interpretation strongly reduces the computational complexity of matching the expression with variables against the world model. In fact, the maximum number of possible assignments for a given meaning M_h with m variables is $O(d^m)$, where d is the number of objects in the domain. Searching through this set to find the assignment(s) that are compatible with the hearer's world model W_h is exponential in the number of variables. In the present example there are 4 entities in W_h, and 7 variables in M_h, which makes the set of possible assignments equal to $4^7 = 16384$. Many language sentences feature a much larger set of words and involve situations that involve a lot more than 4 entities.

Two things can be done: (1) reducing the number of variables in M_h, or (2) shrinking the set of objects and facts in the world model, which reduces d. Human language users use quite a few devices (linguistic and extra-linguistic) to restrict the context and this reduces the domain of the variables and the number of facts that need to be considered, but we will focus here on (1), and particularly how grammar achieves this by signalling equalities between variables.

In "Jill slides the-block to Jack", the syntactic structure is specifying how the variables involved in the different predicates are interlinked. It therefore reduces the number of variables from 7 to 4. Specifically, the fact that "Jill" is the subject of the sentence, indicated by its syntactic category (noun), its position in front of the verb, and the agreement in number and person between the subject and the verb, conveys the information that Jill is the agent of the action communicated by the verb and thus that the variables ?obj1 and ?obj6 should be considered as equal. The linkage of a syntactic pattern (in this case a Subject+Verb+Object+Oblique pattern) to a semantic frame (in this case a CAUSE-MOVE+cause+goal+theme Frame) is called a (grammatical) construction in Construction Grammar and graphically represented as in figure 5. The 'slide' event with its various roles is a specific instance for which this construction can be used. Obviously the construction in figure 5 is more generally applicable to many other CAUSE-MOVE situations, as in: "Jill gives the block to Jack" or "Jane sent a letter to her mother".

In what follows we will treat "to" as a function word, partly to demonstrate how such function words are handled. In a more realistic treatment of the CAUSE-MOVE construction "to" would viewed as the expression of the path of motion, so that the construction also fits sentences like "Fred

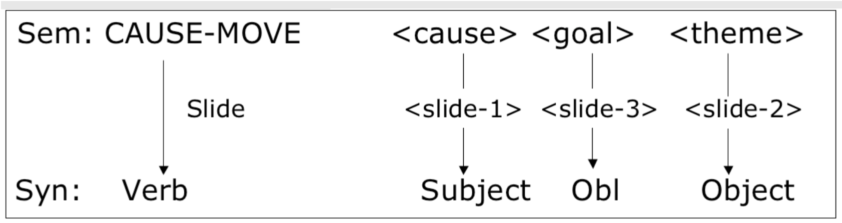

Fig. 5. A construction relates a syntactic pattern such as Subject+Verb+Object+Oblique with a semantic frame such as CAUSE-MOVE+cause+goal+theme

sprayed paint onto the wall", where the path is expressed with "onto". There are of course several other constructions in English applicable to describe the same slide-event. One is the CAUSE-RECEIVE construction which uses the Subject+Verb+Obj+Obj2 pattern to express a CAUSE-RECEIVE+agt+recipient+patient frame, as in "Jill slides Jack the block" or "Jane sent her mother a letter". The CAUSE-MOVE construction only implies motion whereas the CAUSE-RECEIVE construction implies that there is a recipient which effectively received the object as a result of the caused motion [11].

2.4 Expressing Additional Meaning

So far we have seen that a first role of grammar is to convey information about equalities between variables and thus drastically reduce the computational complexity of semantic interpretation. But there is an additional role. Grammatical constructions can introduce additional predicates or changes in the predicates supplied by the lexicon, and thus contribute additional meaning. Consequently the meaning of the whole sentence becomes more than the meaning of the individual parts and the grammar can explicitly specify how this is done. Constructions are therefore not just 'a taxonomic artefact' [7]. This was already illustrated with the subtle distinction in conceptualisation induced by the CAUSE-RECEIVE and the CAUSE-MOVE construction, both based on the same verb. It is further illustrated by the following series of sentences from French which all involve the same verb 'rendre'. Because the verb is each time embedded in a different construction, the meanings become quite different as well.

1a. Il me *rend* malade [Subject+Object+*rendre*+Predicate]
1b. Lit: He me makes sick (He makes me sick)
2a. Je me *rends* a la maison. [Subject+Object+*rendre*+*a*+directional]
2b. I me take to the house (I go home)
3a. Il me *rend* mon livre [Subject+Ind-Object+*rendre*+Object]
3b He me give my book (he gives me my book)

These examples (and there are many more constructions in which *rendre* can participate) show that putting all meaning-contributing aspects in the lexicon (as advocated in lexical-functional grammars such as Head-driven Phrase Structure Grammar [27]) is not always desirable.

As argued more extensively elsewhere (e.g. [23]), grammatical constructions have two important advantages, which are both relevant for the grounding problem:

- First, having an inventory of constructions brings additional expressive power to a language because the same word (like *rendre*) can productively be used in many different constructional contexts and listeners will be able to infer the meaning based on the construction.
- Second, it allows listeners to infer the meaning or usage of unknown words by using the semantic frames that the construction provides. For example, in the sentence "Jill frooples a box to Jack" we know that Jill somehow transfers the box to Jack, even though we do not know what "frooples" means. If a real world scene is available showing a physical transfer between Jill and Jack with specific characteristics (perhaps Jill throws the box to Jack), then semantic interpretation becomes possible and a good guess can be made of the meaning of "frooples". Such bootstrapping strategies have been demonstrated in children for the acquisition of new verb meanings [39] and is one of the main reasons why a construction grammar is interesting from the viewpoint of learning grounded language.

Constructions clearly have different degrees of specifity (i.e. idiomaticity), ranging from very idiomatic constructions built around a particular noun or verb, to very general constructions with wide applicability, such as Subject+Predicate+Object (as in "John reads a book"). Constructions thus form networks where more specific constructions inherit from more general ones and combine with each other to achieve high expressive power. Moreover empirical observations of actual language use show that the inventory of constructions used by an individual (including adults) is constantly changing. Constructions capture conventionalised patterns of usage, but new patterns develop all the time and others may go out of fashion.

2.5 Syntactic and Semantic Categories

A construction such as shown in figure 5 links some semantic aspects of a sentence structure with some of its syntactic aspects by using various semantic and syntactic categories. The slide-event is here conceptualised as a CAUSE-MOVE frame with a cause, a theme and a goal, and there are various selection restrictions on the objects that can fill these roles. There is a long tradition in

AI and cognitive linguistics, dating back to the seventies, for studying such semantic frames and their selection restrictions and it is now known that there is considerable semantic variation among languages [38]. It follows that semantic frames and their selection restrictions will have to be learned as part of learning the language and it will be argued later that sensory-motor grounding, in particular the state transition networks needed for event recognition, can play a significant role in this.

On the syntactic side, a grammatical pattern is specified in terms of grammatical relations, like subject, direct-object, predicate, indirect-object, determiner, modifier, etc. These relations only hold when a particular set of constraints on the form of the sentence is valid, and these constraints in turn make use of the parts of speech (noun, verb, adjective, etc.) and various syntactic features (like gender, number, person, tense, aspect, etc.), which may be involved in concord phenomena and the determination of morphological variations on the word stem.

Grammatical relations, parts of speech, syntactic features are all examples of syntactic categories. There is a very large number of syntactic and semantic categories (surely in the thousands) involved in the grammar of any specific natural language and there is known to be considerable variation among languages concerning the specific inventory of syntactic categories they employ (for example, Japanese has no syntactic gender, Chinese has no syntactic tense nor makes a tight distinction between adjectives and nouns, etc.). Moreover syntactic categories exhibit prototype behavior with clear uses and boundary cases (e.g. many nouns can function as verbs in English, objects or events can often be coerced into new roles that violate their 'natural' selection restrictions [23]).

Although this is a minor point but necessary to handle the examples in this paper, a distinction will be made between natural and grammatical syntactic features. The number of a noun (phrase) is typically telling us something about the status of the entity it refers to, namely whether it involves a single-object (singular) or a set of objects (plural). Number is in this case a natural syntactic category in the sense that it contributes to meaning. On the other hand, the number of a verb is not related to the event described by the verb. Number is in this case a grammatical syntactic feature used for creating specific syntactic phenomena (agreement between subject and predicate). Grammatical syntactic features often start by being meaningful and then adopt a purely grammatical function. Thus the gender of nouns is meaningful in the case of animate beings (compare actor versus actress) but becomes arbitrary when it is generalised to other kinds of objects which have no natural gender. For example, moon is neuter in English, feminine in French or Italian, and masculine in German.

Logic is generally accepted in AI for representing both the facts in a world model and the meaning that is expressed by a sentence, and we follow that tradition as mentioned earlier. But we will also use a logic-based representation for the syntactic and semantic categorisations and the constraints on semantic or syntactic structures that characterise a particular construction. Syntactic categories are predicated over units such as:

string(unit-1,"john"), precedes(unit-1,unit-2), number(singular,natural), direct-object(unit-1,unit-2), tense(unit2, past), ...

The final form of a sentence will be described in a declarative fashion. For example, "John walks home" is represented as

string(unit-1, "john"), string(unit-2,"walks"), string(unit-3,"home") precedes(unit-1, unit-2), precedes(unit-2,unit-3)

The explicit representation of ordering relations makes it much easier to deal with relatively free word order or parse sentences even if the word order constraints are partially violated.

Semantic categories are predicates over the entities in the domain of discourse which are (re-)conceptualisations of the predicates directly derived from the sensory datastreams, such as

time-period(?ev1, before-now), agent(ev1, obj1), cause(ev1, obj3), etc.

3 Fluid Construction Grammars

Research in formal and computational models of Construction Grammar is still in its infancy. Some proposals have been made using feature structures for defining syntactic and semantic structures, and unification-style grammar for specifying constraints on paired syntactic and semantic structures [19], [2]. Recently a parsing system has been reported that uses constructions to build semantic structures that can be fed into a mental simulation module [4].

The Fluid Construction Grammar formalism that will be introduced in this section, is strongly compatible with these developments, but differs in certain technical details which give quite different computational behaviors. Two additional requirements, which are not met in earlier implementations, have guided our design: bi-directionality and fluidity.

(1) All rules regulating the form-meaning mappings must be applicable both in producing (i.e. constructing an utterance that expresses specific meanings derived through a conceptualisation process from a grounded world

model) and in parsing (reconstructing the meaning of an utterance and mapping it back into reality by way of the grounded world model). This is a tough technical requirement which will be achieved by viewing constructions as constraints and language processing as constraint propagation (see [3] for a similar approach).

(2) We argued already that symbol grounding requires agents to constantly align their ways of categorising and conceptualising the world and their ways of expressing these conceptualisations [31]. There are two implications: First of all, there is no guarantee that different agents have the same language inventories and so each agent must keep track of competing hypotheses about what conventions are floating around in the population and choose the one that is likely to have the most communicative success. This will be handled by giving all rules in the lexicon and grammar a particular strength which reflects expected communicative success. Second, after every interaction, agents must be prepared to update their language inventory, either because they had to invent new forms to express meanings that they had not yet expressed before, or because they acquired new form-meaning mappings, or because they had to change the strength of their rules to become more aligned to the verbal behavior of others in the population. This will be achieved by updating the strength of every rule after usage based on success or failure in the communication.

Because fluidity is a defining characteristic of the computational formalism proposed in this paper, we call it 'Fluid Construction Grammar' (FCG). The remainder of this section briefly introduces our formalism, using natural language-like examples of constructions and categories. We do not make any claims however that this formalism is adequate with respect to the full power of natural language, but it is already possible to let communication systems emerge with interesting properties.

3.1 Representing Syntactic and Semantic Structure

Language processing consists in building up the semantic and syntactic aspects of a sentence structure. In FCG, one is not done before the other (as in strictly modular approaches to language processing) but both are built up at the same time. The sentence structure will consist of units with slots containing information about the unit. There is no ordering among the slots and a slot contains always a set of elements which are themselves also unordered. The elements are either atomic, or expressions in predicate calculus notation, or variables (which are atomic symbols preceded with a question mark as in ?*event*). In contrast to other feature structure formalisms [27] feature structures are not used hierarchically as fillers of slots. All units are located at the same level but are labeled (unit-1, unit-2, etc.) and hierarchical structure is explicitly represented with the slots syn-subunits (for syntactic subunits) and sem-subunits (for semantic subunits). The labels allow reference to a unit, predication over units, for example to express ordering constraints, and the use of variables that become bound to specific units. Basically there is a unit for every word in the sentence and for every group of words that forms a larger unit. But there may be units which have only a semantic role (i.e. they have no direct analog in the form) and units which have only a syntactic role (e.g. for grammatical function words like “by” which have no direct analog in meaning).

The main slots of a unit for the semantic aspects are:

(1) Referent: which is the entity in the world-model the unit is about (or a variable that will become bound to an entity)
(2) Meaning: the clauses that will be used to identify the referent.
(3) Sem-cat: the semantic categorisations in which the entity referred to by the unit participates.
(4) Sem-subunits: the set of subunits of this unit from the viewpoint of semantic structure.

The main slots of a unit on the syntactic side are:

(1) Utterance: this is the string itself that is covered by the unit (possibly a multi-word phrase). It is not explicitly listed in the examples that follow.
(2) Form: which is a declarative description of the form in terms of strings, stems, affixes, and precedence or sequencing relations between units
(3) Syn-cat: the syntactic categorisations that are associated with the unit, such as number or gender.
(4) Syn-subunits: the set of subunits of this unit from the viewpoint of syntax.

3.2 Rule sets

The rules of the lexicon and the grammar impose constraints on the syntactic and semantic aspects of a sentence structure. They naturally fall into a set of classes (figure 6) based on the role they play in the overall process, but formally speaking the rules all have the same structure (defined shortly) and they could all be put into one big rule-set in which rules become active whenever they are applicable.

First there are two classes of rules which realise a two-way mapping from syntactic to semantic structure:

(1) Lexical rules (lex-rules) map a lexical item into a set of clauses that specify the meaning of that item. They can be further subdivided into:

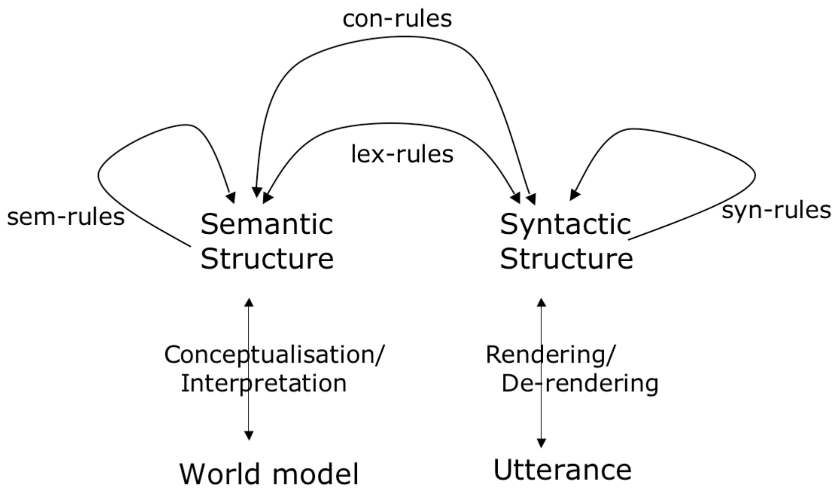

Fig. 6. The different types of rules that are employed for constraining the semantic and syntactic aspects of a sentence structure.

   (a) Lexical stem rules (lex-stem-rules) which map the stem of words to a set of predicates.
   (b) Lexical category rules (lex-cat-rules) which map 'natural' syntactic categories (like the number of a noun, gender in the case of animate entities, tense in the case of the main verb, etc.) to additional meanings.
(2) Construction rules (con-rules) map parts of syntactic structure (i.e. units with particular syntactic categories) into parts of semantic structure (i.e. semantic categories of some units or possibly additional clauses to be added to the meaning). These rules therefore implement constructions of the type shown in figure 5.

Next there are two classes of rules which expand parts of the syntactic or semantic structure by adding syntactic or semantic categorisations:

(1) Syntactic categorisation rules add syntactic categorisations to the syntactic structure. They come in two types:
   (a) Morphological rules (morph-rules) decompose words into a stem with affixes and add syntactic categorisations, such as gender or part of speech.
   (b) Phrase structure rules (ps-rules) relate syntactic properties of a sentence (for example word order) to additional syntactic categorisations, such as subject or direct-object.
(2) Semantic categorisation rules (sem-rules) add semantic categorisations to the semantic structure. For example, they map the arguments of a specific predicate, like 'slide' into the semantic roles of the more abstract CAUSE-MOVE frame.

In line with unification grammar formalisms, language processing will be viewed as a kind of inference process and mechanisms pioneered in logic programming and rule-based systems can therefore be employed to implement parsing and production. The sentence structure (syntactic and semantic) can be viewed as the 'fact base' over which rules operate. Examples of rules are shown in figure 7, 8 and following. The filler of a slot in a rule specifies which elements have to be present in the structure being matched. Curly brackets {, } are used when the contents of a slot have to contain exactly the same elements as in the target structure. In contrast to other unification-based formalisms, the rules do not use numerical indices for referring back to other parts of a structure, but variables, which can be bound not only to specific items, such as the value for the syntactic feature person, but also to units themselves (as ?unit in figure 7).

The bi-directional application of a rule uses two subfunctions: Match and Merge. Match is equivalent to the standard unification operation familiar from logic programming. Merge is equivalent to the standard unification operation in unification grammars, which not only binds variables but also adds all parts from the merging structure to the merged structure.

(1) PRODUCING (i.e. go from meaning to form)
    (a) Match the left pole of a rule against the structure already built. If the rule matches, this yields a set of bindings for the variables or a set of equalities between variables (because the sentence structure being built may also contain variables).
    (b) Merge the right pole of the rule with the sentence structure. Merging means that the instantiated right pole is first matched against the sentence structure, possibly yielding additional bindings, and then the union is taken of the further instantiated right pole and the sentence structure.
(2) PARSING (i.e. go from form to meaning)
    (a) Match the right pole of a rule against the sentence structure. If the rule matches, this yields a set of bindings for the variables or a set of equalities between variables (because the setence structure being built may also contain variables).
    (b) Merge the left pole of the rule with the sentence structure. Merging means that the instantiated left pole is first matched against the sentence structure, possibly yielding additional bindings, and then the union is taken of the further instantiated left pole and the sentence structure.

So producing and parsing are totally analogous, the only thing which changes is the direction of rule application. The type of the rule determines which part of the sentence structure will be used in matching and merging. For example, in the case of a ps-rule (phrase structure rule) both the left and the right pole are syntactic, whereas in the case of a con-rule (a construction) the left pole is semantic and the right pole syntactic.

3.4 An Example of Parsing

Let us look at some examples of rules, relevant for the parsing and production of the example sentence "Jill slides blocks to Jack" [1] . We first look at how the rules are used for parsing and then show later that exactly the same rules work for production. A morphological rule that decomposes "slides" into a stem and a set of syntactic categories is shown in figure 7. Number and person are 'grammatical' as opposed to 'natural', because they do not contribute to meaning. The rule is applicable in both directions. If the stem and the syntactic categories are known (in production) it adds the right pole, namely the string "slides". If the string is seen during parsing, the left pole is merged with the syntactic structure providing information on the stem and the syntactic categorisation of this unit. The use of the affix "-s" is of course productive in English and therefore the rule in figure 7 should just split up "slides" into its two morphemes with another rule associating singular 3d person with "-s", but we simplify here because of space limitations.

```
morph-rule slides
   ?unit
      syn-cat:
         verb,person(3d,syntactic),
         number(singular,syntactic)
      form: stem(?unit,slide)
<==>
   ?unit
      form: string(?unit,"slides")
```

Fig. 7. Example of a morphological rule that decomposes the word "slides" into a stem and a number of syntactic categories.

Figure 8 is an example of a lex-stem rule that constrains the semantic structure based on the presence of the stem "slide" and figure 9 is an example of a lex-cat rule that relates 'natural' syntactic features namely singular number with an additional aspect of meaning, namely that there is a single object. Another syntactic feature that will be used is 'person' which refers to the discourse

[1] An interactive webservice where one can explore such rules has been set up at http://arti.vub.ac.be/FCG/

role of the object (either participant as speaker (1st person) or hearer (2nd person) or external (3d person)).

```
lex-stem-rule slide-entry
   ?unit
      referent: ?ev
      meaning:
         slide(?ev,?truth), slide-1(?ev,?obj1),
         slide-2(?ev,?obj2), slide-3(?ev,?obj3)
⟺
   ?unit
      form: stem(?unit,slide)
```

Fig. 8. Example of a lexical stem rule that associates the word stem "slide" with predicates to be added to the meaning of a semantic structure.

```
lex-cat-rule singular-cat
   ?unit
      referent: ?x
      meaning: status(?x,single-object)
⟺
   ?unit
      syn-cat: number(singular,natural)
```

Fig. 9. Example of a lex-cat rule that associates a natural syntactic feature, in this case number, with a predicate added to the meaning in the semantic structure.

An example of a semantic categorisation rule is shown in figure 10. It maps a specific predicate (in this case 'slide') into a semantic categorisation (in this case 'CAUSE-MOVE') that will be useful later as a selection restriction for the application of the CAUSE-MOVE construction. Notice that the arrow only goes in one direction. Semantic categorisation rules are indeed the only rules that are uni-directional, simply because a slide event is a CAUSE-MOVE event but not every CAUSE-MOVE is a slide event.

An example of a phrase structure rule is shown in figure 11. The rule recognises or specifies the Subject+Verb+Object+Oblique sentence pattern. The left pole of the phrase structure rule contains this abstract pattern and the right pole the syntactic constraints for the realisation of this pattern. These include specific parts of speech, ordering constraints, and agreement between subject and predicate for number and person. Agreement is expressed by using the same variables.

Figure 12 shows the syntactic structure built up by the kinds of rules discussed so far and figure 13 the semantic structure. Note in figure 13 that the variables introducing the different objects in units unit4, unit6, unit2, etc., and those

```
sem-rule slide-cause-move
   ?event-unit
      referent: ?event
      meaning:
         slide(?event,?truth), slide-1(?event,?obj1),
         slide-2(?event,?obj2), slide-3(?event,?obj3)
==>
   ?event-unit
      sem-cat:
         cause-move(?event), cause(?event,?obj1),
         theme(?event,?obj2), goal(?event,?obj3)
```

Fig. 10. Example of semantic rule that reconceptualises the slide event in terms of a CAUSE-MOVE frame.

introducing the roles in the slide event (filling the meaning slot of unit3) are not yet equal. Note also how unit3 contains semantic categories due to the application of the sem-rule shown earlier.

Figure 14 shows an example of a construction. It implements the one shown schematically in figure 5. The main function of this construction is to ensure that the variables introducing the various objects and the roles in the event become equal, in other words that it is known what the roles are of the objects introduced by the various noun phrases in the event introduced by the verb. The work is done by the unification of the left pole of the rule with the semantic structure already built (see figure 13).

The units in the right pole match with units in the sentence structure resulting in a series of bindings (for example ?event-unit gets bound to unit3). Then the left pole, after instantiating these variables, was matched against the semantic structure yielding additional variable bindings, for example, ?cause is bound to ?x-106 and ?x-95, ?goal to ?x-108 and ?x-98, etc. After instantiating the left pole with these new bindings (which includes the introduction of a single variable for all variables that are equal) it is merged with the semantic structure in figure 13 to yield the one shown in figure 15. Note that it was necessary to extract additional bindings by matching the (instantiated) left pole against the sentence structure before merging it in order to equalise the variables.

The meaning of the phrase can be extracted from this semantic structure by taking the union of the meanings of all the units:

Jill(?obj-20), status(?obj-20, single-object), discourse-role(?obj-20,external), slide(?ev-9) slide-1(?ev-9,?obj-20), slide-2(?ev-9,?obj2-14) slide-3(?ev-9,?obj-17), block(?obj-14), status(?obj-14,object-set), discourse-role(?obj-14,external) Jack(?obj-17), status(?obj-17, single-

```
ps-rule svoobl-sentence
  ?top-unit
    syn-cat: svoobl-sentence
    syn-subunits:
      ?verb-unit,?subject-unit, ?obj-unit,?obl-unit
  ?verb-unit
    syn-cat: verb(?top-unit,?verb-unit)
  ?subject-unit
    syn-cat: subject(?top-unit,?subject-unit)
  ?obj-unit
    syn-cat: object(?top-unit,?obj-unit)
  ?obl-unit
    syn-cat: oblique(?top-unit,?obl-unit)
⟺
  ?top-unit
    syn-cat: sentence
    syn-subunits:
      ?verb-unit,?subject-unit, ?obj-unit,?prep-unit,?obl-unit
    form:
      pattern(?subject-unit,?verb-unit,?obj-unit,?prep-unit,?obl-unit)
  ?verb-unit
    syn-cat:
      verb, number(?number,syntactic), person(?person,syntactic)
  ?subject-unit
    syn-cat:
      noun,number(?number,natural), person(?person,natural)
  ?obj-unit
    syn-cat: noun
  ?prep-unit
    form: {string(?prep-unit,to)}
  ?obl-unit
    syn-cat: noun
```

Fig. 11. Example of phrase structure rule specifying the syntactic constraints on the Subject+Verb+Object+Oblique pattern

object), discourse-role(?obj-17,external)

When this expression is matched against the world model, bindings for the variables are found. If there is a unique set in which all variables are bound, the listener has found a unique interpretation for the sentence.

3.5 An Example of Language Production

We now examine how the same set of rules can be used in production. The process starts with a single unit1 with a meaning slot containing the following expression:

```
unit1
   syn-subunits: {unit2,unit3,unit4,unit5,unit6}
   syn-cat: {svoobl-sentence,sentence}
   form:
      {pattern(unit2,unit3,unit4,unit5,unit6)}
unit6
   syn-cat:
      {oblique(unit1,unit6), noun,person(3d,natural),
       number(singular,natural)}
   form:
      {stem(unit6,jill), string(unit6,jill)}
unit4
   syn-cat:
      {object(unit1,unit4), noun,person(3d,natural),
       number(plural,natural)}
   form:
      {stem(unit4,block), string(unit4,blocks)}
unit2
   syn-cat:
      {subject(unit1,unit2), noun,person(3d,natural),
       number(singular,natural)}
   form:
      {stem(unit2,jack), string(unit2,jack)}
unit3
   syn-cat:
      {verb(unit1,unit3), verb,  person(3d,syntactic),
       number(singular,syntactic)}
   form:
      {stem(unit3,slide), string(unit3,slides)}
unit5
   form: {string(unit5,to)}
```

Fig. 12. Syntactic structure after application of all rule sets

.

slide(ev1,true), slide-1(ev1,obj1), slide-2(ev1,obj2), slide-3(ev1,obj3), block(obj2), status(obj2,object-set), discourse-role(obj2,external), Jill(obj3), status(obj3,single-object), discourse-role(obj3,external), jack(obj1), status(obj1,single-object), discourse-role(obj1,external)

The first step is the application of the rules of the lexicon in order to cover this meaning. Each time a lex-stem rule is applied, a new unit is created both in the syntactic and semantic structure and the covered clauses are removed from unit1. The syntactic structure of the new unit contains the stem and its semantic structure that part of the meaning which is covered by that stem. Once units for the individual stems exist, the lex-cat rules can be applied, which add 'natural' syntactic features to them. For example, the lex-cat rule in figure 9 adds singular if the status is 'single-object'. Next the sem-rules add

```
unit3
   sem-cat:
      {cause-move(?x-104), cause(?x-104,?x-106),
       theme(?x-104,?x-107), goal(?x-104,?x-108)}
   referent: ?x-104
   meaning:
      {slide(?x-104,?x-105), slide-1(?x-104,?x-106),
       slide-2(?x-104,?x-107), slide-3(?x-104,?x-108)}
unit4
   referent: ?x-101
   meaning:
      {block(?x-101), status(?x-101,object-set),
       discourse-role(?x-101,external)}
unit6
   referent: ?x-98
   meaning:
      {jill(?x-98),  status(?x-98,single-object),
       discourse-role(?x-98,external)}
unit2
   referent: ?x-95
   meaning:
      {jack(?x-95),  status(?x-95,single-object),
       discourse-role(?x-95,external)}
unit5
unit1
   sem-subunits: {unit2,unit3,unit4,unit5,unit6}
```

Fig. 13. Semantic structure built after application of lexical rules and semantic categorisation rules

additional semantic categorisations.

The resulting syntactic structure is shown in figure 16. There are now units for each of the words that have covered some part of the total meaning. Note that there are no parts of speech assigned yet and that the verb has not yet received features for number or person. There are no grammatical relations (like subject, verb, object, etc.) either. The corresponding semantic structure is shown in figure 17.

The next step is the application of the CAUSE-MOVE construction (figures 5 and 14). The left-pole of the construction matches with the semantic structure in figure 17 so that the right pole can be merged with the syntactic structure. The unit variables of the left pole were bound in the matching (e.g. ?event-unit was bound to unit3) and the right pole instantiated with these variable bindings, matched against the syntactic structure to find additional variable bindings and then fully instantiated and merged with the syntactic structure. The new syntactic structure now contains a specification of the grammatical

```
con-rule cause-move-construction
  ?top-unit
    sem-subunits:
      ?event-unit,?cause-unit, ?goal-unit,?theme-unit
  ?event-unit
    referent: ?event
    sem-cat:
      cause-move-goal(?event), cause(?event,?cause),
      theme(?event,?theme), goal(?event,?recipient)
  ?cause-unit
    referent: ?cause
  ?theme-unit
    referent: ?theme
  ?goal-unit
    referent: ?recipient
⟺
  ?top-unit
    syn-cat: svoobl-sentence
    syn-subunits:
      ?event-unit,?cause-unit, ?theme-unit,?goal-unit
  ?event-unit
    syn-cat: verb(?top-unit,?event-unit)
  ?cause-unit
    syn-cat: subject(?top-unit,?cause-unit)
  ?theme-unit
    syn-cat: object(?top-unit,?theme-unit)
  ?goal-unit
    syn-cat: oblique(?top-unit,?goal-unit)
```

Fig. 14. Example of a construction which relates a CAUSE-MOVE frame to a Subject+Verb+Object+Oblique pattern

relations (subject, verb, etc.). Next the ps-rule shown in figure 11 can be applied, so that the Subject+Verb+Object+Oblique pattern gets translated into surface forms. The left-pole of the ps-rule matches with the syntactic structure built up so far. ?verb is bound to unit3, ?subject to unit2, etc. Then the instantiated right-pole is matched with the syntactic structure giving additional bindings (e.g. ?number is bound to singular and ?person to 3d) and the further instantiation is merged with the syntactic structure to yield, after application of the morph-rules, the one in figure 18. Note that a new unit has been created for the function word "to" and that the right word-order has been established.

The final utterance is extracted directly from this syntactic structure by taking the union of all the form constraints. Ignoring the stem specifications, which

```
unit3
   sem-cat:
      {cause-move(?x-104), cause(?x-104,?x-95),
       theme(?x-104,?x-101), goal(?x-104,?x-98)}
   referent: ?x-104
   meaning:
      {slide(?x-104,?x-105), slide-1(?x-104,?x-95),
       slide-2(?x-104,?x-101), slide-3(?x-104,?x-98)}
unit4
   referent: ?x-101
   meaning:
      {block(?x-101), status(?x-101,object-set),
       discourse-role(?x-101,external)}
unit6
   referent: ?x-98
   meaning:
      {jill(?x-98), status(?x-98,single-object),
       discourse-role(?x-98,external)}
unit2
   referent: ?x-95
   meaning:
      {jack(?x-95), status(?x-95,single-object),
       discourse-role(?x-95,external)}
unit5
unit1
   sem-subunits: {unit2,unit3,unit4,unit5,unit6 }
```

Fig. 15. The final semantic structure after application of the construction. The variables are now equalised.

have no influence on rendering, this yields:

> pattern(unit-jack-entry,unit-slide-entry,unit-block-entry,x-118,unit-jill-entry) string(unit-jill-entry,"Jill"), string(unit-slide-entry,"slides"), string(unit-block-entry,"blocks"), string(x-118,"to"), string(unit-jack-entry,"Jack")

which is rendered as "Jill slides blocks to Jack".

Obviously there are many more issues to be considered in realistic parsing and production, such as the problem that several rules may be applicable thus generating a search space of possibilities or that error-handling must be orchestrated in the case of missing rules. These issues are dealt with in the

```
unit-jack-entry
    syn-cat:
        {person(3d,natural), number(singular,natural)}
    form: {stem(unit-jack-entry,jack)}
unit-jill-entry
    syn-cat:
        {person(3d,natural), number(singular,natural)}
    form: {stem(unit-jill-entry,jill)}
unit-block-entry
    syn-cat:
        {person(3d,natural), number(plural,natural)}
    form: {stem(unit-block-entry,block)}
unit1
    syn-subunits:
        {unit-slide-entry, unit-block-entry, unit-jill-entry,unit-jack-entry}
unit-slide-entry
    form: {stem(unit-slide-entry,slide)}
```

Fig. 16. Syntactic structure after decomposition into different words and the application of lex-cat rules that turn aspects of meaning into syntactic features.

implementation but will not be discussed here further.

4 Grammar Learning

We now turn to the question how construction grammars of the sort described in the previous paragraphs can be autonomously developed by agents through situated language games. As mentioned earlier, we are exploring a constructivist approach [39]. In contrast to the nativist position, defended, for example, by Pinker [26], this approach does not assume that the semantic and syntactic categories as well as the constructions (specifying for example that the agent of an action is linked to the subject of a sentence) are universal and innate. Rather, semantic and syntactic categories as well as the way they are used in constructions is built up in a gradual developmental process, starting from quite specific 'verb-island constructions'. The inventory of categorisations and constructions keeps changing as language users constantly expand and adapt their inventories to the communicative situations they have to deal with.

Earlier work on the acquisition of Construction Grammars [6] has explored observational learning based on Bayesian induction (also used in [28] for example). Here we explore a much more active stance from language users based on the Peircian notion of abduction [10]. The speaker first attempts to use constructions from his existing inventory to express whatever he wants to express. However when that fails or leads to uncertainties in interpretation (i.e.

```
unit-slide-entry
   sem-cat:
      {cause-move(ev1), cause(ev1,obj1),  theme(ev1,obj2), goal(ev1,obj3)}
   referent: ev1
   meaning:
      {slide(ev1,true), slide-1(ev1,obj1),
       slide-2(ev1,obj2), slide-3(ev1,obj3)}
unit1
   sem-subunits:
      {unit-slide-entry,unit-block-entry,unit-jill-entry,unit-jack-entry }
   referent: ev1
unit-block-entry
   referent: obj2
   meaning:
      {block(obj2), status(obj2,object-set), discourse-role(obj2,external)}
unit-jill-entry
   referent: obj3
   meaning:
      {jill(obj3), status(obj3,single-object), discourse-role(obj3,external)}
unit-jack-entry
   referent: obj1
   meaning:
      {jack(obj1), status(obj1,single-object), discourse-role(obj1,external)}
```

Fig. 17. Semantic structure after application of the lexical rules which have decomposed the total semantic structure into separate units, one for each word, and after application of the sem rules which added semantic categorisations.

equalities between variables), the speaker may extend his existing repertoire by inventing new constructions. These new constructions should be such that there is a high chance that the hearer may be able to guess their meaning, for example because they exploit analogies with existing constructions. The hearer also uses as much as possible the constructions in his own inventory to make sense of what is being said. But when there are unknown constructions, or the meanings do not fit with the situation being talked about, the hearer makes an educated guess about what the meaning of the unknown language constructions could be, and adds them as new hypotheses to his own inventory. Abductive constructivist learning relies crucially on settings where both agents have sufficient common ground, share the same situation, have established joint attention, and share communicative goals. Moreover language learners come with shared grounded world knowledge about actions and objects. Both speaker and hearer can use their own inventories as models of that of others in order to guess how the other one will interpret a sentence or why the speaker says things in a particular way. Moreover both speaker and hearer use the relevance principle [30] in order to come up with the most optimal description that is discriminating for the topic.

Because both speaker and hearer are taking risks making abductive leaps, a third activity is needed, namely testing whether the hypotheses are justified

```
unit-block-entry
    form:
        {string(unit-block-entry,blocks),
         stem(unit-block-entry,block)}
    syn-cat: {noun, object(unit1,unit-block-entry),
        person(3d,natural), number(plural,natural)}
unit-slide-entry
    form:
        {string(unit-slide-entry,slides),
         stem(unit-slide-entry,slide)}
    syn-cat:
        {person(3d,syntactic),
         number(singular,syntactic), verb,
         verb(unit1,unit-slide-entry)}
unit-jill-entry
    form:
        {string(unit-jill-entry,jill),
         stem(unit-jill-entry,jill)}
    syn-cat: { noun, oblique(unit1,unit-jill-entry),
        person(3d,natural), number(singular,natural)}
unit-jack-entry
    form:
        {string(unit-jack-entry,jack),
         stem(unit-jack-entry,jack)}
    syn-cat:
        {person(3d,natural),
         number(singular,natural), noun,
         subject(unit1,unit-jack-entry)}
x-118
    form: {string(x-118,to)}
unit1
    form:
        {pattern(unit-jack-entry,unit-slide-entry,
        unit-block-entry,x-118,unit-jill-entry)}
    syn-subunits:
        {unit-jill-entry, x-118,
        unit-block-entry, unit-jack-entry,unit-slide-entry}
    syn-cat: sentence(svoobl-sentence)
```

Fig. 18. Syntactic structure after application of the ps-rule that translates the SVOObl pattern into its surface form.

and aligning them to choices made by the rest of the group. When a construction leads to a successful interaction, there is evidence that this construction is (or is becoming) part of the set of conventions adopted by the group, and language users should therefore prefer it in the future. When the construction fails, the language user should avoid it if alternatives are available. By adapting their inventories after every game, agents progressively align their own language use to that of others and due to the positive feedback loop between use of a particular construction and success (the more success the more use, and the more use the more success), the population gradually settles in a shared inventory. The remainder of this section defines the set of macro-operators that we have developed so far as a first step to implement and hence experiment with this approach.

4.1 Lexicalisation

Lexicalisation means that a new word (lexical unit) is introduced in the lexical inventory of speaker or hearer. For the hearer, lexicalisation is triggered when there is a word for which there is no lexical entry yet. The hearer then reconstructs the potential meaning and associates it with the unknown word. For the speaker, lexicalisation is triggered when there are parts of the target meaning M which are not yet covered by any word and so the speaker introduces a new word to cover them.

We first focus on the hearer and assume there is only one unknown word. In a preparatory step, the hearer detects the unknown word and reconstructs its potential meaning:
Given a sentence S and a target meaning M.

(1) Parse S and extract the parsed meaning P_m.
(2) Compute the uncovered part of target meaning U_m. This is done by taking the difference between the clauses in the guessed meaning M and those in P_m: $U_m = M \backslash P_m$.
(3) Compute uncovered part of the sentence U_f. This is done by retrieving the unit in the syntactic structure that has no associated meaning specification in the semantic structure (which is the case because there was no lexical rule to contribute this meaning).

The hearer now hypothesises that U_f expresses U_m. Two new rules are constructed: (1) A new morph-rule associates the word string with a stem. Later on this rule may receive additional syntactic categories. (2) A new lex-stem-rule associates the stem with the uncovered meaning part.

Suppose the hearer gets the sentence "Jack gives Jill block" and has already lexical entries for jack, jill and block, so that the covered meaning is:

{block(?x-79), jill(?x-78), jack(?x-77)} and the uncovered target meaning U_m is:

$$U_m = \{\text{give-3(ev1,obj3), give-2(ev1,obj2), give-1(ev1,obj1), give(ev1,true)}\}$$

The uncovered part of the sentence U_f is the string "gives" in unit3 because the corresponding unit in the semantic structure does not have any associated meaning after lexicon lookup. This leads now to two new rules. A morph-rule for the word stem (which is initially hypothesised to be equal to the string itself) and a lex-stem-rule which associated the meaning with this stem as shown in figure 19 left. Note that the variables in the give-unit are **not** the same as those used in the other units.

```
morph-rule gives
   ?unit
      form: stem(?unit,"gives")
⟺
   ?unit
      form: string(?unit,"gives")
```

,

```
lex-stem-rule lex-give-3
   ?unit
      referent: ?ev1-82
      meaning:
         give-3(?ev1-82,?x-78), give-2(?ev1-82,?x-79),
         give-1(?ev1-82,?x-77), give(?ev1-82,?true-83)
⟺
   ?unit
      form: stem(?unit,gives)
```

Fig. 19. A newly created morph rule (left) and lex-stem-rule (right)

The lexicalisation operator used by the speaker is identical to that of the hearer, except that the speaker invents a new word when part of the target utterance is not covered, instead of associating a new word used by the speaker. The current implementation uses random strings for new words.

4.2 Grammaticalisation I. Invention

Creating a new construction involves the introduction of a new semantic frame and a new syntactic pattern as well as the rule linking the two. This is triggered when there are variable equalities detected when matching the parsed meaning against the target meaning. We first consider the case where speaker or hearer do not have a construction yet that could be adapted for the present purpose and so they make a totally new one. At present it is assumed that a single construction will cover all equalities to be resolved. There is first a preparatory phrase:

Given a sentence S and a target meaning M.

(1) Parse S and extract the parsed meaning P_m.
(2) Match P_m against the target meaning U_m, obtaining a set of bindings B, and a set of equalities E, equal to the sets of variables that are bound in B to the same object.

(3) If $E \neq \emptyset$ collect the set of all those units that include one of the variables in E as part of their meaning. The substructure of the semantic structure made up these units will act as the basis for the semantic frame, i.e. left pole, of the construction. The substructure of the syntactic structure made up of the same units will act as the basis for the syntactic pattern, i.e. the right pole, of the construction.

By way of example, we start from the same example sentence "Jack gives Jill blocks", assuming that there are lexical entries for each individual word but no construction yet. The semantic structure after application of these lexical rules is as in figure 20. and hence, the parsed meaning P_m is:

```
unit5
    referent: ?x-426
    meaning: {block(?x-426)}
unit4
    referent: ?x-425
    meaning: {jill(?x-425)}
unit2
    referent: ?x-424
    meaning: {jack(?x-424)}
unit1
    sem-subunits: {unit2,unit3,unit4,unit5}
unit3
    referent: ?ref-427
    meaning:
        {give(?ref-427,?x-428), give-1(?ref-427,?x-429),
         give-2(?ref-427,?x-430), give-3(?ref-427,?x-431)}
```

Fig. 20. Semantic structure after application of existing lexical rules, prior to grammar.

{block(?x-426), jill(?x-425), jack(?x-424), give(?ref-427,?x-428), give-1(?ref-427,?x-429), give-2(?ref-427,?x-430), give-3(?ref-427,?x-431)}

Matching the parsed meaning against the world model yields the set of bindings

B = ((?x-424 . obj1) (?x-425 . obj3) (?x-426 . obj2) (?x-431 . obj3) (?x-430 . obj2) (?x-429 . obj1) (?x-428 . true) (?ref-427 . ev1))

and hence the equalities:

E = {{?x-424, ?x-429}, {?x-426, ?x-430}, {?x-425,?x-431}}

These equalities need to be resolved by the creation of a new construction.

The left pole of the construction is now built as follows:

(1) Given the semantic substructure consisting of all the units from the semantic structure that should participate in the left pole of the construction.
(2) Standardise the variables that are equal so that they become a single variable.
(3) Introduce variables for the units
(4) Create an overarching unit which contains all participating units as sem-subunits. It is further called the top-unit.
(5) For each unit convert the meaning of that unit into an analogous set of semantic categories and construct new semantic categorisations linking each meaning to its corresponding semantic categories through a sem-rule.

Thus, the units involved in the equalities, i.e. unit3 (because of ?x-429, ?x-430 and ?x-431), unit2 (because of ?x-424), unit4 (because of ?x-425), and unit5 (because of ?x-426) are grouped into a new unit. Then the variables from the equalities are standardised and new variables are introduced for the units. And finally semantic categories are introduced for each, yielding the left pole shown in figure 21. Building the right pole of the construction (also shown in

```
?unit-432
    referent: ?ref-427
    sem-cat:
        sem-cat-268(?ref-427,?x-428), sem-cat-269(?ref-427,?x-424),
        sem-cat-270(?ref-427,?x-426), sem-cat-271(?ref-427,?x-425)
?unit-433
    referent: ?x-424
    sem-cat: sem-cat-272(?x-424)
?unit-434
    referent: ?x-425
    sem-cat: sem-cat-273(?x-425)
?unit-435
    referent: ?x-426
    sem-cat: sem-cat-274(?x-426)
?unit-436
    sem-subunits:
        ?unit-432,?unit-433,?unit-434,?unit-435
```

```
?unit-432
    syn-cat:
        syn-cat-312(?unit-436,?unit-432)
?unit-433
    syn-cat:
        syn-cat-314(?unit-436,?unit-433)
?unit-434
    syn-cat:
        syn-cat-316(?unit-436,?unit-434)
?unit-435
    syn-cat:
        syn-cat-318(?unit-436,?unit-435)
?unit-436
    syn-cat: syn-cat-310
    syn-subunits:
        ?unit-432,?unit-433,?unit-434,
        ?unit-435
```

Fig. 21. Left pole and right pole of the new grammatical construction.

figure 21) involves two steps:

(1) The equivalent of all the units on the semantic pole is used as the basis for the syntactic pole and a new syntactic category for the top unit is introduced.
(2) New grammatical relations are constructed for each participating unit, linking that unit with the top unit of the construction.

Now the required sem-rules have to be built, where the predicates in the meaning slot forming the left pole of the sem-rule are linked to their respective semantic categories as right pole of the sem-rule as shown in figure 22.

```
sem-rule sem-rule-76
   ?unit-432
      referent: ?ref-427
      meaning:
         give(?ref-427,?x-428), give-1(?ref-427,?x-424),
         give-2(?ref-427,?x-426), give-3(?ref-427,?x-425)
==>
   ?unit-432
      sem-cat:
         sem-cat-268(?ref-427,?x-428),
         sem-cat-269(?ref-427,?x-424),
         sem-cat-270(?ref-427,?x-426),
         sem-cat-271(?ref-427,?x-425)
```

Fig. 22. Example of semantic categorisation rules

The phrase structure rule can also be built. Its left pole is identical to the syntactic pattern of the construction rule. The right pole contains the constraints on the pattern. These include first of all syntactic categorisations (parts of speech) for the possible units but could include also word order, agreement, or any other syntactic feature. In the example shown in figure 23 only syntactic categories are used as constraints.

Finally, the morph-rules of the words involved must be expanded, specifying that the words now belong to the syntactic categories introduced in the phrase structure rule, as in the example shown in 24.

The syntactic and semantic structures in figures 25 and 26 result from the application of all rules built up so far. The extracted meaning P_m is now:

{give(?x-482,?x-483), give-1(?x-482,?x-481), give-2(?x-482,?x-479), give-3(?x-482,?x-480), jack(?x-481), jill(?x-480), block(?x-479)}

and when this is matched against the target meaning, we get the following set of bindings:

B = ((?x-480 . obj3) (?x-479 . obj2) (?x-481 . obj1) (?x-483 . true) (?x-482 . ev1))

There are no more equalities in this binding list due to the application of the construction. Note also how all the syntactic and semantic categorisations are present in the syntactic and semantic structures.

The way that the speaker adds a new construction is totally analogous to the way that the hearer does it. After first producing a sentence using his existing rules, the speaker *re-enters* the resulting sentence and construct a parse tree

```
ps-rule ps-syn-cat-310-rule
  ?unit-432
    syn-cat:
      syn-cat-312(?unit-436,?unit-432)
  ?unit-433
    syn-cat:
      syn-cat-314(?unit-436,?unit-433)
  ?unit-434
    syn-cat:
      syn-cat-316(?unit-436,?unit-434)
  ?unit-435
    syn-cat:
      syn-cat-318(?unit-436,?unit-435)
  ?unit-436
    syn-cat: syn-cat-310
    syn-subunits:
      ?unit-432,?unit-433,?unit-434, ?unit-435
⟺
  ?unit-436
    syn-subunits:
      ?unit-432,?unit-433,?unit-434, ?unit-435
  ?unit-432
    syn-cat: syn-cat-322
  ?unit-433
    syn-cat: syn-cat-321
  ?unit-434
    syn-cat: syn-cat-320
  ?unit-435
    syn-cat: syn-cat-319
```

Fig. 23. Example of built phrase structure rule.

```
morph-rule gives-entry
  ?unit
    form: stem(?unit,gives)
    syn-cat: syn-cat-326
⟺
  ?unit
    form: string(?unit,gives)
```

Fig. 24. Morphological rule is expanded with a new syntactic category.

from which the parsed meaning P_m can be extracted and matched against the meaning that the speaker originally wanted to express. The set of bindings and equalities can then easily be extracted and new rules can be built as shown in the previous paragraphs.

```
unit1
   syn-subunits: {unit2,unit3,unit4,unit5}
   syn-cat: syn-cat-327(sentence)
   form:
      {pattern(unit2,unit3,unit4,unit5)}
unit5
   syn-cat:
      {syn-cat-335(unit1,unit5), syn-cat-336}
   form:
      {stem(unit5,block), string(unit5,block)}
unit4
   syn-cat:
      {syn-cat-333(unit1,unit4), syn-cat-337}
   form:
      {stem(unit4,jill), string(unit4,jill)}
unit2
   syn-cat:
      {syn-cat-331(unit1,unit2), syn-cat-338}
   form:
      {stem(unit2,jack), string(unit2,jack)}
unit3
   syn-cat:
      {syn-cat-329(unit1,unit3), syn-cat-339}
   form:
      {string(unit3,gives), stem(unit3,gives)}
```

Fig. 25. Syntactic structure obtained by application of the newly created rules.

4.3 Grammaticalisation II. Coercion

The building of new constructions from scratch is rare in natural language. Usually a construction already exists and if a situation arises that requires additional grammar, speakers first try to re-use existing constructions by coercing certain predicates and words to fit in, which can be done by introducing additional syntactic and semantic categorisation rules. Agents could follow the same approach in the invention and negotiation of artificial languages. We consider here only the simplest example of this type of coercion, and much more research is needed to identify additional 'grammar-making operations'.

First we consider the case of the hearer. Assume that the hearer gets the utterance "Jack gives Frank blocks", and has lexical rules for all the words as well as the give-construction as explained in the previous section. Rather then creating a new construction, the hearer should try to find a construction which fits best with the incoming sentence, and coerce the element which does not yet fit (in this case "Frank"). Finding an appropriate structure goes in three steps:

```
unit3
   sem-cat:
      {sem-cat-275(?x-482,?x-483), sem-cat-276(?x-482,?x-481),
       sem-cat-277(?x-482,?x-479),
       sem-cat-278(?x-482,?x-480)}
   referent: ?x-482
   meaning:
      {give(?x-482,?x-483), give-1(?x-482,?x-481),
       give-2(?x-482,?x-479), give-3(?x-482,?x-480)}
unit2
   sem-cat: {sem-cat-279(?x-481)}
   referent: ?x-481
   meaning: {jack(?x-481)}
unit4
   sem-cat: {sem-cat-280(?x-480)}
   referent: ?x-480
   meaning: {jill(?x-480)}
unit5
   sem-cat: {sem-cat-281(?x-479)}
   referent: ?x-479
   meaning: {block(?x-479)}
unit1
   sem-subunits: {unit2,unit3,unit4,unit5}
```

Fig. 26. Semantic structure obtained by application of the newly created rules.

(1) The first step is similar to the creation of a new construction: Based on the equalities, all those units in the semantic structure are assembled that have variables participating in one of the equalities.
(2) Then existing constructions are examined to find whether there are any whose left pole matches partially with this substructure. The best matching construction is chosen.
(3) Suppose such a rule is found, then this rule tells the agent what the semantic category has to be of the unit that did not match and a new sem-rule can therefore be introduced to fix this problem.
(4) The missing grammatical relation can also be derived from the same unit and from there the phrase-structure rule that specifies which syntactic conditions (such as syntactic categorisation) have to be satisfied to establish this grammatical relation. This in turn leads to new morph-rules for the string that is to be coerced into this construction.

Let us see how this works for the example sentence "Jack gives Frank blocks". We assume that the hearer has construction-3 and ps-syn-cat-310-rule listed earlier. The construction cannnot trigger because the Frank-unit (unit4) does not have the required syntactic category (syn-cat-37) nor the required semantic category (sem-cat-20). The problem is solved with the rules shown in figure 27.

The way that the speaker fits a new element into an existing construction is totally analogous to the way that the hearer does it, except for the preparatory

```
sem-rule sem-cat-13
  ?unit
    referent: ?ref-79
    meaning: frank(?ref-79)
==>
  ?unit
    sem-cat: sem-cat-20(?ref-79)
```

```
morph-rule frank-morph
  ?unit
    form: stem(?unit,"frank")
    syn-cat: syn-cat-37
<=>
  ?unit
    form: string(?unit,"frank")
```

Fig. 27. New semantic and syntactic categorisation rules to coerce "Frank" to be reusable.

work. The speaker first constructs the sentence using existing rules and then re-enters the result in the parsing process. Typically production does not reach strings but only stems if constructions cannot apply. The speaker then extracts the set of bindings and equalities by comparing the parsed meaning with the intended meaning. If there are any equalities, the speaker tries to find the construction that was the closest and uses that as a basis for creating an extension as described for the hearer.

This description of re-use is of course highly simplified. Most importantly, re-use should not just be taking place based on structuralist criteria, i.e. simply finding the closely matching construction. Rather, grounding should also play a major role. In the experiments implemented so far, this was done by using the definitions of events used by the event recognition algorithm as the source for computing analogies [33]

Another issue which we will not further develop is that both the grammatical constructions as well as the syntactic and semantic categorisation rules have a strength which is increased when the rules participated in a successful communication, or decreased in case of failure. The same semiotic dynamics that leads to a shared lexicon [36] progressively coordinates the individual grammars. The learning mechanism proposed here is indeed a 'constructivist' approach to category formation. Categories are not derived from statistical clustering but simple postulated by language users. At first these categories are ad hoc and have only a single entity as its member, but as constructions are re-used, more instances become member of the category and so they are getting a richer content. The exemplar-based learning of categorisation results in the prototype behavior also seen with the linguistic categories found in human natural languages.

5 Conclusions

Against the background of earlier work on the self-organisation of lexicons and ontologies in population of agents, this paper discussed the issue in what way grammar can aid in language grounding. We considered two questions: (1) What is the best grammatical framework for supporting language grounding, and (2) What is the best learning strategy for acquiring grammars according to this framework. The paper argued in favor of Construction Grammar, mainly because (1) this approach zooms in on the contribution that grammar can make to building up meaning, including resolving equalities between variables so that the complexity of semantic interpretation gets reduced, and (2) it insists that grammar is grounded and embodied in sensory-motor experiences and situated interactions.

We also argued in favor of a constructivist approach to grammar learning, which claims that grammar is built up in a gradual process using a number of general cognitive learning mechanisms (such as associative learning, constructivist categorisation, etc.) embedded in concrete embodied verbal interactions. The paper showed some directions in which this approach can be operationalised but obviously a lot more work needs to be done and is now possible due to the computational formalism that was introduced.

Research funded by Sony CSL with additional funding from the ESF-OMLL program, the EU FET-ECAgents project and the (French) CNRS OHLL project. The author is indebted to Joachim de Beule, Nicolas Neubauer, and Joris Van Looveren, who have made important contributions, particularly to the implementations underlying the present paper.

References

[1] Allen, J.F. and G. Ferguson (1994) Actions and events in interval temporal logic. Journal of Logic and Computation, 4(5):531–579, October 1994.

[2] Bergen, B.K. and N.C. Chang. 2003. *Embodied Construction Grammar in Simulation-Based Language Understanding.* In J-O Ostman and M. Fried (Eds.) Construction Grammar(s): Cognitive and Cross-Language Dimensions John Benjamin Publ Cy. Amsterdam.

[3] Blache, P. 2002 *Analyse des dpendances distance laide de graphes de contraintes Le cas des disloques* TAL. Volume 43, nr.3, pp 109-128.

[4] Bryant, J. (2004) Scalable Construction-Based Parsing and Semantic Analysis. In Proceedings of the Second International Workshop on Scalable Natural Language Understanding. Boston 2004

[5] Cangelosi, and D. Parisi 2003. *Simulating the Evolution of Language.* Springer-Verlag, Berlin.

[6] Chang, N.C. and T.V. Maia. 2003. *Learning Grammatical Constructions* In: Cohen, P. and T. Oates (2001) p. 10-16.

[7] Chomsky, N. 2003. *A minimalist program for linguistic theory* MIT Occasional Papers in Linguistics. Vol 1. MIT Linguistics Department, Cambridge Ma.

[8] Clark, H. H., and Brennan, S. A. (1991). Grounding in communication. In: Resnick, L.B., J.M. Levine, and S.D. Teasley (Eds.). Perspectives on socially shared cognition. Washington: APA Books.

[9] Cohen, P. and T. Oates, eds. (2001) Proceedings of the AAAI Spring Symposium on Grounding. AAAI Press, Anaheim, Ca.

[10] Fann, K.T. 1970. *Peirce's Theory of Abduction* Martinus Nijhoff, The Hague.

[11] Goldberg, A.E. 1995. *Constructions.: A Construction Grammar Approach to Argument Structure.* University of Chicago Press, Chicago

[12] Dominey, P. (2003) Learning Grammatical Constructions in a Miniature Language from Narrated Video Events. Proceedings 25th Annual meeting of the Cognitive Society, Boston.

[13] Feldman, J. and S. Narayanan (2003). Embodied Meaning in a neural theory of language . Brain and Language. to appear.

[14] Garrod, S., and Anderson, A. (1987) Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. Cognition 27(2), 181-218.

[15] Harnad, S. (1990) The Symbol Grounding Problem. Physica D 42: 335-346.

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

[17] Hopper, P. (1987) Emergent grammar. Berkeley Linguistics Conference (BLS), 13:139–157

[18] Inoue, H. and Y. Kuniyhoshi (1993) Qualitative recognition of ongoing human action sequences. Robotics and Vision, pp. 1600-1609.

[19] Kay, P. and C. Fillmore (1999) Grammatical Constructions and Linguistic Generalizations: the What's X doing Y? Construction Language 75, pp. 1-33.

[20] Kirsh, D. (1991) Foundations of artificial intelligence: The big issues. Artificial Intelligence 47 (1991) 3-30.

[21] Langacker, R. (1991) Foundations of Cognitive Grammar. 2 vols. Stanford University Press, Stanford.

[22] Lakoff, G. (1987) Women, fire, and dangerous things. University of Chicago Press, Chicago.

[23] Michaelis, L. 2004. *Entity and Event Coercion in a Symbolic Theory of Syntax* In J.-O. Oestman and M. Fried, (eds.),Construction Grammar(s): Cognitive Grounding and Theoretical Extensions. Constructional Approaches to Language series, volume 2. Amsterdam: Benjamins.

[24] Michaelis, Laura A. and Knud Lambrecht. 1996. Toward a Construction-based Theory of Language Function: The Case of Nominal Extraposition. Language 72: 215-247

[25] Nilsson, N.J. ed. (1984) Shakey the Robot, SRI Technical Note no. 323 (1984), SRI, Menlo Park, California.

[26] Pinker, S. 1998. *Learnability and Cognition: The acquisition of Argument Structure.* The MIT Press, Cambridge Ma.

[27] Pollard, C. and I. Sag 1994. *Head-driven phrase structure grammar.* Center for the Study of Language and Information of Stanford University. University of Chicago Press, Chicago.

[28] Roy, D. 2001. *Learning Visually Grounded Words and Syntax of Natural Spoken Language.* Evolution of communication 4(1).

[29] Siskind, J. (2000) Visual Event Classification Through Force Dynamics. AAAI Conference 2000. AAAI Press, Anaheim Ca. pp. 159-155.

[30] Sperber, D. and D. Wilson (1986), Relevance: Communication and Cognition. Harvard University Press, Cambridge Ma.

[31] Steels, L. (2001) Language Games for Autonomous Robots. IEEE Intelligent systems, September/October 2001, p. 16-22.

[32] Steels, L. (2003) Evolving grounded communication for robots. Trends in Cognitive Science. Volume 7, Issue 7, July 2003 , pp. 308-312.

[33] Steels, L. (2004) Constructivist Development of Grounded Construction Grammars. Proceedings Annual Meeting Association for Computational Linguistics Conference. Barcelona.

[34] Steels, L. and J-C. Baillie 2003. *Shared Grounding of Event Descriptions by Autonomous Robots.* Journal of Robotics and Autonomous Systems 43, 2003, pp. 163-173.

[35] Steels, L. and T. Belpaeme (2005) Coordinating Perceptually Grounded Categories through Language. A Case Study for Colour. BBS target article to appear 2005.

[36] Steels, L. and Kaplan, F. (2000) Collective learning and semiotic dynamics. In: Floreano, D. and Nicoud, J-D and Mondada, F., (ed.) (2000) Advances in Artificial Life (ECAL 99), Lecture Notes in Artificial Intelligence 1674, Springer-Verlag, Berlin. pp. 679-688.

[37] Steels, L. and F. Kaplan (2001) AIBO's first words. The social learning of Symbol and meaning. Evolution of Communication 4(1).

[38] Talmy, L. (2000) Toward a Cognitive Semantics: Concept Structuring Systems (Symbol, Speech, and Communication) The MIT Press, Cambridge Ma.

[39] Tomasello, M. and P.J. Brooks 1999. *Early syntactic development: A Construction Grammar approach* In: Barrett, M. (ed.) (1999) *The Development of Language* Psychology Press, London. pp. 161-190.