Towards Bidirectional Processing Models of Sign Language: A Constructional Approach in Fluid Construction Grammar

Remi van Trijp (remi@csl.sony.fr)
Sony Computer Science Laboratory Paris
6, Rue Amyot, 75005 Paris France

Abstract

Sign languages (SL) require a fundamental rethinking of many basic assumptions about human language processing because instead of using linear speech, sign languages coarticulate facial expressions, shoulder and hand movements, eye gaze and usage of a three-dimensional space. SL researchers have therefore advocated SL-specific approaches that do not start from the biases of models that were originally developed for vocal languages. Unfortunately, there are currently no processing models that adequately achieve both language comprehension and formulation, and the SL-specific developments run the risk of becoming alienated from other linguistic research. This paper explores the hypothesis that a construction grammar architecture offers a solution to these problems because constructions are able to simultaneously access and manipulate information coming from many different sources. This claim is illustrated by a proof-of-concept implementation of a basic grammar for French Sign Language in Fluid Construction Grammar.

Keywords: sign language; language processing; construction grammar; computational modeling

Introduction

Sign languages (SL) require a fundamental reassessment of many of the basic assumptions about human language processing because instead of using linear streams of speech, sign languages *coarticulate* facial expressions, shoulder and hand movements, and eye gaze (Emmorey, 2002). Adequate models of sign languages must also be able to cope with the use of a three-dimensional *signing space* and with "the omnipresence of *iconicity* at all levels of the language" (Braffort & Filhol, 2011, p. 192). The study of sign languages is therefore of utmost importance for our understanding of human cognition. Moreover, the *World Health Organization* estimates that over 5% of the world population (360 million people including 32 million children) has disabling loss of hearing, so the development of intelligent systems for sign languages has an enormous potential for societietal benefits.

Unfortunately, while lexical signs can now reasonably be recognized and produced in their base forms, many hard problems remain unsolved when it comes to grammatical processing (Vogler & Goldenstein, 2008). One important issue that SL researchers have identified is that mainstream linguistic theories and contemporary language technologies are too strongly biased towards the properties of vocal languages (particularly Indo-European languages), and have therefore developed SL-specific language technologies (Huenerfauth, 2006; Filhol, 2012). Currently, however, there are no models available that adequately handle both language formulation and comprehension, and there is a risk that these important research efforts become alienated from the rest of linguistics.

This paper proposes that a *construction grammar* architecture (Fillmore, 1988; Goldberg, 1995; Croft, 2001; Östman & Fried, 2004; Steels, 2011) offers a solution to these problems. The most salient property of construction grammar is that it eliminates the need for separate layers of linguistic organization, such as the sharp distinction between lexicon and syntax. Instead, *constructions* (i.e. conventionalized mappings between meaning/function and form) are posited as the sole data structure for representing linguistic knowledge. Constructions are thus able to simultaneously access and manipulate multimodal sources of information, which offers unique opportunities for modeling sign languages.

The paper illustrates its claims through a proof-of-concept implementation of a basic grammar for French Sign Language in Fluid Construction Grammar (FCG; `www.fcg-net.org`; Steels, 2011), an open-source grammar formalism for exploring constructional analyses in language formulation, comprehension and learning. The goal of this implementation is not to replace or compete with existing models in sign language research, but rather to demonstrate how linguistically sound and bidirectional processing models could work in tandem with the recent SL-specific developments. The FCG-implementation may thus also lead to tighter interactions between sign language research and the rest of the linguistic enterprise.

Sign Language Modeling

Sign languages are so-called *under-resourced* languages, which means that there are only few reference grammars and small corpora available (Braffort & Filhol, 2011). Formal sign language modeling thus began in all earnestness using rule-based grammars that have strong ties to particular linguistic theories, such as the ViSiCast system based on Head-Driven Phrase Structure Grammar (Marshall & Safar, 2004).

Most traditional theories invite SL-researchers to analyze SL utterances as a sequence of lexical gestures that are combined with each other in a phrase-structural analysis. However, as argued by Braffort and Filhol (2011), all other features must then be aligned with the boundaries of these lexical gestures, which makes it hard to capture coarticulation in sign language syntax. Moreover, sign languages are much more complex than sequences of signs in their *citation form* (i.e. signs as they appear in isolation): in continuous signing, several parameters of a sign can be modified with infinite possibilities, as illustrated for the sign BALL in French Sign Language in Figure 1 (from Braffort & Filhol, 2011, Fig. 9.1). This Figure shows the citation form of the sign on the

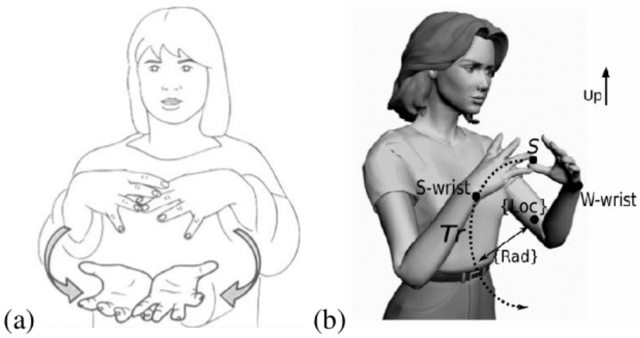

Figure 1: This Figure from Braffort and Filhol (2011, Fig. 9.1) shows the citation form of the sign BALL on the left, and its parametric variation on the right.

left, which consists of two hands making a circle movement. On the right, the parametric variation of the sign is illustrated. For instance, the speaker can express a big ball by increasing the radius of the circle (`Rad` in the Figure), or identify the spatial location of the ball with respect to a previously signed referent by changing the location of the circle movement (`Loc` in the Figure).

Besides rule-based grammars, the increasing availability of annotated corpora (e.g. DEGELS1; Braffort & Boutora, 2012) has now made it possible to also include techniques from machine learning and probability theory in the development of sign language models. Most of these methods, however, have been developed for speech and have only limited success when applied to sign languages, particularly when it comes to the spatial properties of SL (Dalle, 2006) or the aforementioned fact that signs rarely occur in their citation form (Braffort & Boutora, 2012).

Sign language researchers have therefore started to advocate and implement SL-specific methods and technologies. One example, discussed in more detail by Filhol (2012) and Braffort and Filhol (2011), is the AZee language for SL generation. The AZee language is fundamentally different from static constraint systems in that it takes a more procedural approach: constraints in the AZee system can best be viewed as a set of instructions that build XML specifications, which can then be visualized as sign language utterances using a software avatar. These XML specifications are built using a set of types and operators. For instance, the type `cstr` allows to impose constraints at a particular point in time, such as eye gaze direction. An example of an operator is `morph`, which controls non-skeletal articulators such as facial muscles. The AZee approach is thus able to modify generic rules on the fly, such as taking a lexical gesture's citation form and changing its parameters depending on the context.

Computational Construction Grammar

Construction grammar is a family of cognitive-functional approaches to language that, amongst other reasons, emerged as a reaction against the assumption in mainstream linguistics that grammatical phenomena can be divided into a *core* (which can be described as a set of rules) and a *periphery* (which is a list of exceptions). Construction grammarians also reject the assumption that language is divided into different, largely independent modules such as phonology, syntax and semantics. Instead, they argue that all linguistic knowledge can be described using *constructions* as the central representation unit. Charles J. Fillmore, widely recognized as the father of construction grammar, defined a construction as follows:

> By **grammatical construction** we mean *any* syntactic pattern which is assigned to one or more conventional functions in a language, together with *whatever* is linguistically conventionalized about its contribution to the meaning or the use of structures containing it. (Fillmore, 1988, p. 36, italics added)

I italicized the words *any* and *whatever* in this quote to emphasize the fact that constructions are able to represent any kind of mapping between meaning/function and form. For instance, as Fillmore (1988, p. 35) explains, constructions are not limited to the immediate-dominance constraints of phrase-structure rules (i.e. every phrase-structure rule is a relation between a parent and its immediate children in a local tree configuration) but can make direct reference to the linguistic information they require, wherever this information may be located.

Constructional approaches are currently thriving in all areas of linguistics (Goldberg, 1995; Croft, 2001; Östman & Fried, 2004) and have also attracted the attention of formal and computational linguists (e.g. Steels, 2004; Bergen & Chang, 2005; Boas & Sag, 2012). The most advanced line of work in computational construction grammar was instigated by Steels (2004), who proposed a computational data structure in which constructions can be implemented as mappings between two feature structures (typically representing meaning/function on the one hand, and form on the other). This in turn led to the development of Fluid Construction Grammar (FCG; Steels, 2011), which is currently the only construction grammar formalism to handle both language formulation and comprehension.[1]

In recent work, Steels (to appear) has offered a formal specification of a construction that is independent from how constructions are implemented in the FCG-system. I will briefly summarize this specification in this section in order to demonstrate its relevance for the challenges of sign languages. Let us first start from the well-known representation of a phrase structure grammar rule, as shown in example (1).

[1]It should be noted that Sign-Based Construction Grammar (SBCG; Boas & Sag, 2012), which is a variant of HPSG, can also be implemented for both formulation and comprehension using tools that have been developed for HPSG such as the TRALE system (Richter, 2006). However, SBCG redefines constructions as local tree configurations similar to the immediate-dominance rules of a phrase structure grammar, hence it does not formalize the notion of a construction as envisioned by Fillmore (1988), which is the one adopted in this paper.

$$S \quad \rightarrow \quad NP \quad VP \tag{1}$$

A phrase structure rule is a declarative rule that specifies an immediate-dominance constraint between its left-hand side and its right-hand side (Chomsky, 1956). Computational and formal linguists have proposed various important extensions and modifications to such traditional rules. For example, in Generalized Phrase Structure Grammar (GPSG; Gazdar, Klein, Pullum, & Sag, 1985), the symbols of these rules are no longer atomic, but complex categories that can be described using feature-value pairs (as pioneered by Kay, 1979). Secondly, GPSG separates word order constraints from immediate-dominance relations, so the rule of example (1) can be used both for languages in which the NP precedes the VP, or for languages in which the order is reversed.

Example (2) shows how example (1) can be reformulated with feature-value pairs and explicit word ordering constraints using the formal notation that I will apply throughout this paper. The symbol < specifies a linear precedence constraint between the NP and VP constituents, and symbols that start with a question mark are *variables* whose values are underspecified. For instance, the variables `?p` and `?n` indicate that the rule underspecifies which values should be assigned to the person and number features of the NP and VP, but the fact that the same variable symbols are used in both constituents imposes an equality constraint between them.

$$\left[\text{ category: S }\right] \rightarrow \left[\begin{array}{l}\text{category: NP}\\ \text{agreement:}\\ \quad\text{person: ?p}\\ \quad\text{number: ?n}\end{array}\right] < \left[\begin{array}{l}\text{category: VP}\\ \text{agreement:}\\ \quad\text{person: ?p}\\ \quad\text{number: ?n}\end{array}\right] \tag{2}$$

Constructions, as operationalized by Steels (2004, to appear), are different from such phrase structure rules in two important ways. First, the right-hand side of a construction is divided between a *formulation lock* and a *comprehension lock*. These *locks* specify the conditions under which a construction can be applied in processing. The formalization thus changes the direction of the arrow to the left in order to indicate that information on the left-hand side is only made available if the conditions on the right-hand side are satisfied. Example (3) shows a lexical construction for the word *ball* (corresponding to the rule *Noun* → *"ball")* with the formulation lock above the single line in the right-hand side of the construction, and the comprehension lock under the line. The formulation lock specifies that the construction may only apply in formulation if the meaning *ball(?x)* needs to be verbalized (here we use a first-order logic calculus for representing meaning), and the comprehension lock specifies that the construction may only apply in comprehension if the string "ball" has been observed in the input.

$$\left[\begin{array}{l}\textbf{?ball-unit}\\ \hline\hline \text{referent: ?x}\\ \text{sem-cat: identifier}\\ \text{syn-cat: noun}\end{array}\right] \leftarrow \left[\begin{array}{l}\textbf{?ball-unit}\\ \hline\hline \text{\# meaning:}\\ \quad\text{predicates: }\{\text{ball(?x)}\}\\ \hline \text{\# form:}\\ \quad\text{string: ``ball''}\end{array}\right] \tag{3}$$

The second major difference is that the arrow symbol in a construction no longer implies an immediate-dominance relation,[2] but should be read as "the constraints on the left-hand side can be imposed on the constraints on the right-hand side." In other words, constructions can directly access information that falls beyond the scope of local tree configurations without resorting to additional formal machinery such as transformations or feature passing.

This second difference is technically achieved by reifying feature-value pairs as *units* that have a unique name, as can be seen in example (3) above the double line (`?ball-unit`). Traditional formalisms typically lack such an explicit notion of units so the only way of accessing information is through path descriptions of nested feature structures. In a construction, feature structures can be retrieved directly by either using a unit's name, or by finding a unit whose features match with either the formulation or comprehension lock of a construction.

Crucially, there are no restrictions on which feature-value pairs can form a unit. For example, the Subject-Verb construction in example (4) imposes a linear precedence constraint between a clause's subject (i.e. accessing the utterance's *functional structure*) and its main verb phrase (i.e. accessing the utterance's *constituent structure*). Example (5) shows the Topic construction, which imposes a clause's topic to take up clause-initial position, thereby accessing the utterance's *information structure*.

$$\begin{array}{c}\left[\begin{array}{l}\textbf{?clause-unit}\\ \hline\hline \text{subunits: }\{\text{?subject-unit, ?vp-unit}\}\\ \text{meaning:}\\ \quad\text{referent: ?ev}\\ \quad\text{predicates: }\{\text{vantage-point(?ev, x)}\}\\ \text{sem-cat: proposition}\\ \text{syn-cat: clause}\end{array}\right] \leftarrow \\ \left[\begin{array}{l}\textbf{?subject-unit}\\ \hline\hline \text{meaning:}\\ \quad\text{referent: ?x}\\ \text{sem-cat:}\\ \quad\text{ref-expression}\\ \hline \text{syn-function:}\\ \quad\text{subject}\\ \text{syn-cat: NP}\end{array}\right] < \left[\begin{array}{l}\textbf{?vp-unit}\\ \hline\hline \text{meaning:}\\ \quad\text{referent: ?ev}\\ \text{sem-cat:}\\ \quad\text{pred-expression}\\ \hline \text{valence:}\\ \quad\{\text{subject(?subject-unit)}\}\\ \text{syn-cat: VP}\end{array}\right]\end{array} \tag{4}$$

$$\left[\begin{array}{l}\textbf{?topic-unit}\\ \hline\hline \text{pragm-function:}\\ \quad\text{clause-topic}\end{array}\right] \leftarrow \left[\begin{array}{l}\textbf{?topic-unit}\\ \hline\hline \text{meaning:}\\ \quad\text{referent: ?x}\\ \quad\text{\# predicates:}\\ \qquad\{\text{clause-topic(?ev, ?x)}\}\\ \hline \text{\# form:}\\ \quad\text{position: clause-initial}\\ \text{syn-cat: NP}\end{array}\right] \tag{5}$$

[2] In other words, a construction does not impose any restrictions on its left-hand and right-hand sides and therefore has the expressive power of an *unrestricted grammar* (Chomsky, 1959). Such expressive power is not uncommon in grammar formalisms (see e.g. Carpenter, 1991, on the power of lexical rules in the style of HPSG and categorial grammars), but needs to be managed carefully when scaling a grammar to broad coverage.

As demonstrated by van Trijp (2014), by simultaneously accessing these different kinds of linguistic information, such constructions may freely combine with each other to form different utterance types such as *the boy kicked the ball*, *the ball the boy kicked* (topicalization) and *what did the boy kick?* (WH-questions).

Since there are no constraints on which sets of feature-value pairs can be reified as units, constructions can easily specify units that pertain to SL-specific properties such as eye gaze, hand and shoulder movements, and the signing space. Moreover, form constraints in the FCG-system make reference to units rather than to actual words, so there is no need to align for instance gestural units with lexical boundaries, which was one of the main problems of traditional formalisms (Braffort & Filhol, 2011).

Towards a Construction Grammar Model for Bidirectional Sign Language Processing

Parametric Variation. One important obstacle in sign language modeling is parametric variation in the expression of gestural units. That is, gestural units do not simply appear in their citation form when used in continuous signing, as illustrated before in Figure 1. Instead of "reading off" lexical units from a linear representation device such as a phrase structure tree, SL-specific models such as AZee therefore take a more procedural approach in which language formulation yields a set of form specifications that can be visualized using avatar software (Filhol, 2012). The FCG-system takes the same approach by producing form specifications that can be *rendered* into an utterance.[3]

Example (6) shows a construction that captures the citation form of the sign for [BALL]. Let us first look at the right-hand side of the construction. The formulation lock contains the same meaning representation as used for the vocal word *ball* in example (1). If the speaker wishes to express this meaning, the remainder of the construction is thus unlocked. Since the sign for *ball* is a highly lexicalized sign, the comprehension lock contains a *pointer* to the sign's citation form. That is, in language comprehension the information of this construction is unlocked if the sign for [BALL] is recognized.

If a sign is conventionally associated with a number of parameters, these can be incorporated as feature-value pairs in the left-hand side of the construction, as shown for the parameters `loc` (location) and `rad` (radius). Note, however, that the values of these features are variables, which means that their actual values are underspecified. The construction also specifies a *type* for the sign, which I will explain in more detail in a few paragraphs.

[3] There is also technical congruence between FCG and state-of-the-art SL systems: FCG has been developed in Common Lisp, which uses S-expressions that can be translated quite straightforwardly into the XML specifications of other systems.

```
[ ?ball-unit               ]      [ ?ball-unit                     ]
[ referent: ?x             ]      [ # meaning:                     ]
[ form:                    ]  ←   [     predicates: {ball(?x)}     ]   (6)
[   loc: ?loc              ]      [ # form:                        ]
[   rad: ?rad              ]      [     citation-form: "ball"      ]
[   type: circle           ]
[ sem-cat: identifier      ]
[ syn-cat: noun            ]
```

As mentioned before, French Sign Language users can express the concept of a big ball by increasing the radius of the circular movement made with both hands. In order to capture this parametric variation, we can implement a construction that does not have its own citation form, but instead applies to the citation form of another sign and thereby modifies it. Example (7) shows how such a construction might look like. In its comprehension lock, the construction specifies that if a sign is recognized with *up* as a value for the form-feature `rad` (that is, a radius was detected that is significantly larger than the sign's prototypical radius), then this can be mapped onto the concept of bigness. The left-hand side of the construction is empty because no additional information needs to be imposed on the sign.

```
      [ ?unit-x                    ]
      [ meaning:                   ]
      [   referent: ?x             ]
∅ ←   [   # predicates: {big(?x)}  ]   (7)
      [ sem-cat: identifier        ]
      [ form:                      ]
      [   rad: up                  ]
      [ syn-cat: noun              ]
```

When applied in formulation, the construction specifies that the concept of bigness can be expressed by increasing the radius of a sign. Obviously, this construction is not general enough because the radius feature is not appropriate for all signs. The concept of [BIG] can thus be better implemented as an operation whose outcome depends on the type of the sign that it modifies, similar to method specialization in object-oriented programming. For example, in Lisp we can implement a generic function for the modifier `modify-size` that takes a *type* and a *value* as arguments:

```
(defgeneric modify-size (type value)
   (:documentation "Controls the size
                    parameters of a sign."))
```

Sup++pose that the sign for ball is typed as `circle`, we can now write a specialized method that has to return an appropriate feature structure:

```
(defmethod modify-size ((type (eql 'circle))
                        (value t))
   `(rad: ,value))
```

The FCG-system includes special operators for calling such methods at processing time. In the formal notation, a call to a method is indicated by the string ++ followed by the method name and its arguments. For example, the notation ++*modify-size(circle, up)* will call the method that specializes on the type `circle` and return the feature-value pair *rad: up*. The new definition of the construction for bigness is shown in example (8). Note that the call to the operator *modify-size* passes the variable `?type` as the operator's first argument. This variable name is also used as the value of the feature `type` just above, so its value will be bound to the actual type of the sign that is being modified.

$\emptyset \leftarrow$ [**?unit-x** | meaning: referent: ?x; # predicates: {big(?x)}; sem-cat: identifier | form: type: ?type; ++modify-size(?type, up); syn-cat: noun] (8)

Multilinearity. A second important challenge is to handle "the multilinearity and the complex synchronisation patterns involving all (manual and non-manual) articulators of SL" (Braffort & Filhol, 2011, p. 199). For example, the utterance *the child approaches the car* in French Sign Language consists of four manual units (shown in Figure 2) and two distinct eye gaze directions (ibid., at p. 200–201).

Following an OSV pattern, the first part of the utterance involves the two-handed sign CAR followed by its *placement* in the signing space. Braffort and Dalle (2008) found that placement in French Sign Language always follows the same pattern: after the object has been signed, a classifier (arm, finger or hand) is established at a target location with a small downward movement. An eye gaze directed at the same location immediately precedes the classifier. Example (9) shows a construction that captures these constraints.

[**?placement-unit** | subunits: {?noun, ?eye-gaze, ?classifier}; referent: ?x; form: placement: ?target; orientation: ?orientation; sem-cat: ref-expression; syn-cat: NP] $\leftarrow$ [**?noun** | referent: ?x; sem-cat: identifier | syn-cat: noun] $<$

[**?eye-gaze** | $\emptyset$ | form: eye-gaze-direction: ?target] $\leq$ [**?classifier** | $\emptyset$ | form: articulator: ?articulator; handshape: ?handshape; orientation: ?orientation; placement: ?target; syn-cat: classifier]

(9)

Let us again first look at the right-hand side of the construction. The pattern consists of three units (which I called *?noun*, *?eye-gaze* and *?classifier*) that need to be synchronized with each other. First, *?noun* is a gestural unit that in our example matches with the sign CAR. The symbol $<$ indicates that it precedes the *?eye-gaze* unit, which specifies that the eye gaze should be directed at a particular location in the signing space

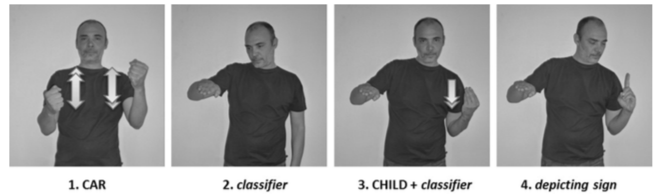

Figure 2: This Figure from Braffort and Filhol (2011, Fig. 9.3) shows four manual units for the utterance *The child approaches the car.*

(here: underspecified through the variable *?target*). The symbol $\leq$ indicates that this unit in turn must immediately precede the *?classifier* unit with potentially some temporal overlap between the units. The *?classifier* unit is bound to the same location in the signing space by repeating the variable *?target* as the value of its *placement* feature. This unit can also be parametrized in terms of which articulator was used (here: the signer's weak hand), which handshape is required (here: a specific classifier for vehicles), and which orientation the signed object takes.

The left-hand side of the construction specifies that these three units can be grouped together into a higher-level unit. For want of a more appropriate terminology, the construction states that together these units can be seen as a *referring expression* or a *noun phrase*. The target and orientation of the signed object are percolated upwards to this *?placement-unit* by repeating the variables *?target* and *?orientation* in the unit's form feature.

One important side-note is that grouping together the three units on the right-hand side as subunits of the *?placement-unit* does not prevent those units to be member of other groups as well. Indeed, instead of the single dominance relations of phrase structure trees, constructions allow units to be member of multiple higher-level units at the same time in order to facilitate information access and to make the formal synchronisation across units more flexible.

Conclusions and Outlook

Sign languages have unique properties that make them an indispensable object of study for our understanding of human cognition in all of its complexities. The differences with vocal languages are so vast that many assumptions in (computational) linguistics about linguistic representations and language processing must be reconsidered. State-of-the-art models in sign language research have therefore been developed from the ground up in order to eliminate the biases of traditional approaches based on speech. However, as a result, the field risks becoming alienated from the rest of linguistics, and there currently are no models available that adequately handle both language formulation and comprehension.

This paper proposed that construction grammar offers a solution to both problems because constructions are able to simultaneously access and manipulate various sources of linguistic information, which makes it possible to handle issues such as parametric variation and multilinearity that are at the heart of sign language grammar. The paper supported this argument through a proof-of-concept implementation in Fluid Construction Grammar that works for both formulation and comprehension. While it is clear that much fundamental work remains to be done in all areas of SL research (in terms of modeling, learning and corpus annotation), this paper has shown that a constructional approach holds much promise for achieving broad coverage grammars of sign languages.

Acknowledgments

The research reported in this paper was conducted at and funded by the Sony Computer Science Laboratory Paris. I wish to thank my colleagues for their constructive feedback and comments, particularly Luc Steels (ICREA Institut de Biologia Evolutiva, UPF-CSIC), Katrien Beuls (VUB AI-Lab), Miquel Cornudella Gaya and Paul Van Eecke (Sony CSL Paris). I would also like to thank Annelies Braffort (LIMSI-CNRS) for the interesting discussion and for sharing with me her expertise on the challenges and issues in Sign Language modeling.

References

Bergen, B. K., & Chang, N. (2005). Embodied Construction Grammar. In J.-O. Östman & M. Fried (Eds.), *Construction grammars: Cognitive grounding and theoretical extensions.* Amsterdam: John Benjamins.

Boas, H. C., & Sag, I. A. (Eds.). (2012). *Sign-based Construction Grammar*. Chicago/Stanford: CSLI Publications.

Braffort, A., & Boutora, L. (2012). DEGELS1: A comparable corpus of French Sign Language and co-speech gestures. In *Proceedings of the eight international conference on language resources and evaluation (lrec 2012)* (pp. 2426–2429). Istanbul: ELRA.

Braffort, A., & Dalle, P. (2008). Sign language applications: Preliminary modeling. *Universal Access in the Information Society, Emerging Technologies for Deaf Accessibility in the Information Society*, *6*(4), 393–404.

Braffort, A., & Filhol, M. (2011). Constraint-based sign language processing. In P. Blache, H. Christiansen, V. Dahl, D. Duchier, & J. Villadsen (Eds.), *Constraints and language* (pp. 191–217). Cambridge: Cambridge Scholar Publishing.

Carpenter, B. (1991). The generative power of categorial grammars and head-driven phrase structure grammars with lexical rules. *Computational Linguistics*, *17*(3), 301–313.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, *2*, 113–124.

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control*, *2*(2), 137–167.

Croft, W. (2001). *Radical construction grammar: Syntactic theory in typological perspective*. Oxford: Oxford UP.

Dalle, P. (2006). High level models for sign language analysis by a vision system. In *Workshop on the representation and processing of sign language: Lexicographic matters and didactic scenarios (lrec 2006)* (pp. 17–20). Genes: ELRA.

Emmorey, K. (2002). *Language, cognition and the brain: Insights from sign language research*. Mahwah NJ: Lawrence Erlbaum.

Filhol, M. (2012). Combining two synchronisation methods in a linguistic model to describe sign language. In E. Efthimiou, G. Kouroupetroglou, & S.-E. Fotinea (Eds.), *Gesture and sign language in human-computer interaction and embodied communication* (pp. 194–203). Berlin/Heidelberg: Springer.

Fillmore, C. J. (1988). The mechanisms of "Construction Grammar". In *Proceedings of the fourteenth annual meeting of the Berkeley Linguistics Society* (pp. 35–55). Berkeley CA: Berkeley Linguistics Society.

Gazdar, G., Klein, G., Pullum, G., & Sag, I. (1985). *Generalized phrase structure grammar*. Oxford and Cambridge MA: Blackwell Publishing and Harvard University Press.

Goldberg, A. E. (1995). *A construction grammar approach to argument structure*. Chicago: Chicago UP.

Huenerfauth, M. (2006). *Generating American Sign Language classifier predicates for English-to-ASL machine translation*. Unpublished doctoral dissertation, University of Pennsylvania, Pennsylvania.

Kay, M. (1979). Functional grammar. In *Proceedings of the Berkeley Linguistics Society* (pp. 142–158). Berkeley, CA: Berkeley Linguistics Society.

Marshall, I., & Safar, E. (2004). Sign language generation in an ALE HPSG. In *The 11th international conference on Head-Driven Phrase Structure Grammar* (pp. 189–201). Leuven: CSLI Publications.

Östman, J.-O., & Fried, M. (2004). Historical and intellectual background of Construction Grammar. In M. Fried & J.-O. Östman (Eds.), *Construction grammar in a cross-language perspective* (pp. 1–10). Amsterdam: John Benjamins.

Richter, F. (2006). *A web-based course in grammar formalisms and parsing.* URL: http://milca.sfs.uni-tuebingen.de/A4/Course/PDF/gramandpars.pdf, Retrieved on 24 November 2011.

Steels, L. (2004). Constructivist development of grounded construction grammars. In *Proceedings 42nd annual meeting of the association for computational linguistics* (pp. 9–19). Barcelona: ACL.

Steels, L. (Ed.). (2011). *Design patterns in Fluid Construction Grammar*. Amsterdam: John Benjamins.

Steels, L. (to appear). The basics of Fluid Construction Grammar. *Constructions and Frames*.

van Trijp, R. (2014). Long-distance dependencies without filler-gaps: A cognitive-functional alternative in Fluid Construction Grammar. *Language and Cognition*, *6*(2), 242–270. doi: http://dx.doi.org/10.1017/langcog.2014.8

Vogler, C., & Goldenstein, S. (2008). Toward computational understanding of sign language. *Technology and Disability*, *20*, 109–119.