Book Review

Design Patterns in Fluid Construction Grammar

Luc Steels (editor)
Universitat Pompeu Fabra and Sony Computer Science Laboratory, Paris

Reviewed by
Nathan Schneider, Carnegie Mellon University
and Reut Tsarfaty, Uppsala University

In computational modeling of natural language phenomena, there are at least three modes of research. The currently dominant statistical paradigm typically prioritizes *instance coverage*: Data-driven methods seek to use as much information observed in data as possible in order to generalize linguistic analyses to unseen instances. A second approach prioritizes detailed *description* of grammatical phenomena, that is, forming and defending theories with a focus on a small number of instances. A third approach might be called *integrative*: Rather than addressing phenomena in isolation, different approaches are brought together to address multiple challenges in a unified framework, and the behavior of the system is demonstrated with a small number of instances. *Design Patterns in Fluid Construction Grammar* (*DPFCG*) exemplifies the third approach, introducing a linguistic formalism called Fluid Construction Grammar (FCG) that addresses parsing, production, and learning in a single computational framework.

The book emphasizes grammar-engineering, following broad-coverage descriptive paradigms that can be traced back to Generalized Phrase Structure Grammar (GPSG) (Gazdar et al. 1985), Lexical Functional Grammar (LFG) (Bresnan 2000), Head-Driven Phrase-Structure Grammar (HPSG) (Sag and Wasow 1999), and Combinatory Categorial Grammar (CCG) (Steedman 1996). In all of these cases, a formal meta-framework allows computational linguists to formalize their hypotheses and intuitions about a language's grammatical behavior and then explore how these representational choices affect the processing of natural language utterances. Many of the aforementioned approaches have engendered large-scale platforms that can be used and reused to provide formal description of grammars for different languages, such as Par-Gram for LFG (Butt et al. 2002) and the LinGO Grammar Matrix for HPSG (Bender, Flickinger, and Oepen 2002).

FCG offers a similar grammar engineering framework that follows the principles of Construction Grammar (CxG) (Goldberg 2003; Hoffmann and Trousdale 2013). CxG treats *constructions* as the basic units of grammatical organization in language. The constructions are viewed as learned associations between *form* (e.g., sounds, morphemes, syntactic phrases) and *function* (semantics, pragmatics, discourse meaning, etc.). CxG does not impose a strict separation between lexicon and grammar—indeed, it is perhaps best known as treating semi-productive idioms like "the X-er, the Y-er" and "X let alone Y" on equal footing with lexemes and "core" syntactic patterns (Fillmore,

---

Kay, and O'Connor 1988; Kay and Fillmore 1999). FCG, like other CxG formalisms—namely, Embodied Construction Grammar (Bergen and Chang 2005; Feldman, Dodge, and Bryant 2009) and Sign-Based Construction Grammar (Boas and Sag 2012)—is unification-based.[1] The studies in this book describe constructions and how they can be combined in order to model natural language interpretation or generation as feature structure unification in a general search procedure.

The book has five parts, covering the groundwork, basic linguistic applications, processing matters, advanced case studies, and, finally, features of FCG that make it fluid and robust. Each chapter identifies general strategies (*design patterns*) that might merit reuse in new FCG grammars, or perhaps in other computational frameworks.

Part I: Introduction lays the groundwork for the rest of the book. "Introducing Fluid Construction Grammar" (by Luc Steels) presents the aims of the FCG formalism. FCG was designed as a framework for describing linguistic units (constructions—their form and meaning), with an emphasis on language variation and evolution ("fluidity"). The constructionist approach to language is described and the argument for applying it to study language variation and change is defended. Psychological validity is explicitly ruled out as a modeling goal ("The emphasis is on getting working systems, and this is difficult enough"; page 4). The architects of FCG set out to include both sides of the processing coin, however—parsing (interpretation) and production (generation). The concept of search in processing is emphasized, though some of the explanations of processing steps are too abstract for the reader to comprehend at this point. A further desideratum—robustness to noisy input containing disfluencies, fragments, and errors—is given as motivating a constructionist approach.

The next chapter, "A First Encounter with Fluid Construction Grammar" (by Steels), describes the mechanisms of FCG in detail. In FCG, a working analysis hypothesized in processing is known as *transient structure*; the transduction of form to meaning (and vice versa) selects a sequence of constructions that apply to the transient structure to gradually expand it until reaching a final analysis. Identifying constructions that may apply to a transient structure presents a non-trivial search problem, also addressed by the architects of FCG. The sheer number of technical details make this chapter somewhat overwhelming. Most of the chapter is devoted to the low-level feature structures and the operations manipulating them. Templates—a practical means of avoiding boilerplate code when defining constructions—are then introduced, and do most of the heavy lifting in the rest of the book.

In *Part II: Grammatical Structures*, we begin to see how constructions are defined in practice. "A Design Pattern for Phrasal Constructions" (by Steels) illustrates how constructions are used to describe the combination of multiple units into higher-level, typed phrases. Skeletal constructions compose with smaller units to form hierarchical structures (essentially similar to the Immediate Constituents Analysis of Bloomfield [1933] and follow-up work in structuralist linguistics [Harris 1946]), and a range of additional constructions impose form (e.g., ordering) constraints and add new meaning to the newly created phrases. This chapter is of a tutorial nature, illustrating the step-by-step application of four kinds of noun phrase constructions to expand transient structures in processing. Over twenty templates are introduced in this chapter; they encapsulate design patterns dealing with hierarchical structure, agreement, and feature percolation.

---

1 Less formal members of the Construction Grammar family include cognitive approaches in the
   Berkeley tradition (Lakoff 1987; Goldberg 1995, 2006), Cognitive Grammar (Langacker 1987),
   and Radical Construction Grammar (Croft 2001).

An aspect of phrasal constructions that is not yet dealt with is the complex linking of se-mantic arguments and the morphosyntactic categorizations of the composed elements.

"A Design Pattern for Argument Structure Constructions" (by Remi van Trijp) then builds on the formal machinery presented in the previous chapter to explicitly address the complex mappings between semantic arguments (agent, patient, etc.) and syntactic arguments (subject, object, etc.). This mapping is a complex matter due to language-specific conceptualization of semantic arguments and different means of morphosyntactic realization used by different languages. In FCG, each lexical item introduces its linking potential in terms of the different types of semantic and syntactic arguments that it may take, with no particular mapping between them. Each argument structure construction imposes a partial mapping between the syntactic and semantic arguments to yield a particular argument structure instantiation, one of the multiple alternatives that may be available for a single lexical item. This account stands in sharp contrast to the lexicalist view of argument-structure (the view taken in LFG, HPSG, and CCG) whereby each lexical entry dictates all the necessary linking information. The construction-based approach is defended for its ability to deal with unknown words[2] and constructional coercion[3] (Goldberg 1995). The argument structure design pattern allows FCG to crudely recover a partial specification of the form-meaning mapping of these elements, which is important for robust processing (see subsequent discussion).

Part III: Managing Processing addresses how FCG transduces between a linguistic string and a meaning representation, where the two directions (parsing and production) share a common declarative representation of linguistic knowledge (the grammar). This entails assembling an analysis incrementally on the basis of the grammar, the input, and any partial analysis that has already been created. With FCG (and unification grammars more broadly) this search is nontrivial, and streamlining search (i.e., minimizing non-determinism and avoiding dead ends) is a key motivator of many of the grammar design patterns suggested in the book.

"Search in Linguistic Processing" (by Joris Bleys, Kevin Stadler, and Joachim De Beule) deals mainly with the problem of choosing which of multiple compatible con-structions to apply next. Whereas the default heuristic search in FCG is a greedy, depth-first search (which can backtrack if the user-defined end-goal has not yet been achieved) the FCG framework allows for a guided search through scores that reflect the relative tendency of a construction to apply next. The authors suggest that such scoring can be informed by general principles, for instance: (i) specific constructions are preferred to more general ones, and (ii) previously co-applied constructions are preferred. Choosing appropriate constructions to apply early on dramatically reduces the time needed for processing the utterance.

"Organizing Constructions in Networks" (by Pieter Wellens) takes this idea to the next level, and proposes to organize the different constructions in networks of conditional dependencies. A conditional dependency links two constructions where one provides necessary information for the application of the other. These dependency networks can be updated whenever an input is processed so that the system learns to search more efficiently when the same constructions are encountered in the future. Using these networks to guide the search thus significantly reduces the search for compatible constructions. An empirical effort to quantify this effect indeed shows a

---

2 For example, "He blicked the napkin off the table"—a reasonable inference is that the subject caused the napkin to leave the table.
3 "He sneezed the napkin off the table"—note that *sneeze* is not normally transitive.

sharp reduction in search time; unlike the held-out experimental paradigm accepted in statistical NLP, however, the parsed/produced sentence is assumed to have been seen already by the system.

Part IV: Case Studies addresses three challenging linguistic phenomena in FCG. "Feature Matrices and Agreement" (by van Trijp) on German case offers a new unification-based solution to the problem of feature indeterminacy. For instance, in the sentence *Er findet und hilft Frauen* 'He finds and helps women', the first verb requires an accusative object, whereas the second requires a dative object; the coordination is allowed only because *Frauen* can be either accusative or dative. *Kindern* 'children', which can only be dative, is not licensed here. Encoding case in a single feature on the *Frauen* construction wouldn't work because the feature would have to unify with contradictory values (from the verbs' inflectional features). Instead, case restrictions specified lexically for a noun or verb can be expressed with a distinctive feature *matrix*, with each matrix slot holding a variable or the value + or −. Unification then does the right thing—allowing *Frauen* and forbidding *Kindern*—without resorting to type hierarchies or disjunctive features.

"Construction Sets and Unmarked Forms" (by Katrien Beuls) on Hungarian verbal agreement models a phenomenon whereby morphosyntactic, semantic, and phonological factors affect the choice between poly- and mono-personal agreement—that is, the decision whether a Hungarian transitive verb should agree with its object or just with its subject. The case and definiteness of the object and the person hierarchy relationship between subject and object determine which kind of agreement obtains, and phonological constraints determine its form. To make the different levels of structure interact properly, constructions are grouped into sets (lexical, morphological, etc.) and those sets are considered in a fixed order during processing. Construction sets also allow for efficient handling of unmarked forms (null affixes)—they are considered only after the overt affixes have had the opportunity to apply, thereby functioning as defaults.

"Syntactic Indeterminacy and Semantic Ambiguity" (by Michael Spranger and Martin Loetzsch) on German spatial phrases models the German spatial terms for *front*, *back*, *left*, and *right*. To model spatial language in situated interaction with robots, two problems must be overcome. The first is *syntactic indeterminacy*: Any of these spatial relations may be realized as an adjective, an adverb, or a preposition. The second is *semantic ambiguity*, specifically when the perspective (e.g., whose 'left'?) is implicit. Both are forms of underspecification which could cause early splits in the search space if handled naïvely. Much in the spirit of the argument structure constructions (see above), the solutions (which are too technical to explain here) involve (a) disjunctive representations of potential values of a feature, and (b) deferring decisions until a more opportune stage.

Part V: Robustness and Fluidity (by Steels and van Trijp) surveys the different features of the system that ensure robustness in the face of variation, disfluencies, and noise. Natural language is fluid and open-ended. There is variation between speakers, there are disfluencies and speech errors, and noise may corrupt the speech signal. All of these may jeopardize the interpretability of the signal, but human listeners are adept at processing such input. In the spirit of usage-based grammar (Tomasello 2003), FCG emphasizes the attainment of a *communicative goal*, rather than ensuring grammaticality of parsed/produced utterances. This is accomplished with a diagnostic-repair process that runs in parallel to parsing/production. Diagnostics can test for unknown words, unfamiliar meanings, missing constructions, and so on. Diagnostic tests are implemented by *reversing* the direction of the transduction process: a speaker may assume the hearer's point of view to analyze what she has produced

in order to see whether communicative success has been attained. Likewise, a hearer may produce a phrase according to his own interpretation of the speaker's form, and check for a match. If a test fails, repair strategies such as proposing new constructions, relaxing the matching process for construction application, and coercing constructions to adapt to novel language use are considered.

Fluidity and robustness are the hallmarks of FCG, and the computational framework has been used in experiments that assume embedded communication in robotic agents. This research program is developed at length by Steels (2012b).

Discussion. Like the legacy of the GPSG book (Gazdar et al. 1985), this book's main merit is not necessarily in its technical details or computational choices, but in demonstrating the feasibility of implementing the constructional approach in a full-fledged computational framework. We suggest that the CxG perspective presents a formidable challenge to the computational linguistics/natural language processing community. It posits a different notion of modularity than is observed by most NLP systems: Rather than treat different levels of linguistic structure independently, CxG recognizes that multiple formal components (phonological, lexical, morphological, syntactic) may be tied by convention to a specific meaning or function. Systematically describing these "cross-cutting" constructions and their processing, especially in a way that scales to large data encompassing both form and meaning and accommodates both parsing and generation, would in our view make for a more comprehensive account of language processing than our field is able to offer today. Thus, we hope this book will be provocative even outside of the grammar engineering community.

This book is not without its weaknesses. In parts the writing is quite technical and terse, which can be daunting for readers new to FCG. Contextualization with respect to other strands of computational linguistics and AI research is, for the most part, lacking, though a second FCG book (Steels 2012a) picks up some of the slack on this front.[4] *DPFCG* does not address the feasibility of learning constructions directly from data,[5] nor does it discuss the expressive power of the formalism in relation to learnability results (such as that of Gold [1967]). As admitted by the authors, much more work would be needed to build life-size grammars. Still, we hope that readers of *DPFCG* will appreciate the authors' vision for a model of linguistic form and function that is at once formal, computational, fluid, and robust.

References

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei.

Bergen, Benjamin K. and Nancy Chang. 2005. Embodied Construction Grammar in simulation-based language understanding.

---

4 In particular, the Embodied Construction Grammar formalism noted earlier offers a similar computational framework, though it reflects different research goals. The most important difference is that FCG was designed to study language evolution, and ECG to study language acquisition and use from a cognitive perspective. FCG is largely about processing (support for production as well as parsing is considered essential); ECG places a greater premium on the meaning representation, formalizing a number of theoretical constructs from cognitive semantics. Chang, De Beule, and Micelli (2012) compare and contrast the two approaches in detail.

5 Ideas that may be of relevance here are discussed in the Data-Oriented Parsing literature (Scha 1990; Bod 2003) and the statistical learning of Stochastic Tree Substitution Grammars (O'Donnell 2011).

In Jan-Ola Östman and Mirjam Fried, editors, *Construction Grammars: Cognitive Grounding and Theoretical Extensions*. John Benjamins, Amsterdam, pages 147–190.

Bloomfield, Leonard. 1933. *Language*. Holt, Rinehart and Winston Inc.

Boas, Hans C. and Ivan A. Sag, editors. 2012. *Sign-Based Construction Grammar*. Number 193 in CSLI Lecture Notes. CSLI Publications, Stanford, CA.

Bod, Rens. 2003. An efficient implementation of a new DOP model. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 19–26, Budapest.

Bresnan, Joan. 2000. *Lexical-Functional Syntax*. Blackwell.

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel grammar project. In *Proceedings of COLING 2002, Workshop on Grammar Engineering and Evaluation*, Taipei.

Chang, Nancy, Joachim De Beule, and Vanessa Micelli. 2012. Computational construction grammar: Comparing ECG and FCG. In Luc Steels, editor, *Computational Issues in Fluid Construction Grammar*. Springer Verlag, Berlin, pages 259–288.

Croft, William. 2001. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford University Press, Oxford.

Feldman, Jerome A., Ellen Dodge, and John Bryant. 2009. Embodied Construction Grammar. In Bernd Heine and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis*. Oxford University Press, Oxford, pages 111–138.

Fillmore, Charles J., Paul Kay, and Mary Catherine O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of 'let alone.' *Language*, 64(3):501–538.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag. 1985. *Generalised Phrase Structure Grammar*. Blackwell, Oxford, England.

Gold, E. Mark. 1967. Language identication in the limit. *Information and Control*, 10(5):447–474.

Goldberg, Adele E. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press, Chicago.

Goldberg, Adele E. 2003. Constructions: A new theoretical approach to language. *Trends in Cognitive Sciences*, 7(5):219–224.

Goldberg, Adele E. 2006. *Constructions at work: the nature of generalization in language*. Oxford University Press, Oxford.

Harris, Zellig S. 1946. From morpheme to utterance. *Language*, 22(3):161–183.

Hoffmann, Thomas and Graeme Trousdale, editors. 2013. *The Oxford Handbook of Construction Grammar*. Oxford University Press, Oxford.

Kay, Paul and Charles J. Fillmore. 1999. Grammatical constructions and linguistic generalizations: The What's X doing Y? construction. *Language*, 75(1):1–33.

Lakoff, George. 1987. *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. University of Chicago Press, Chicago.

Langacker, Ronald W. 1987. *Foundations of Cognitive Grammar*, volume 1. Stanford University Press, Stanford, CA.

O'Donnell, Timothy J. 2011. *Productivity and reuse in language*. Ph.D. dissertation, Harvard University, Cambridge, MA.

Sag, Ivan A. and Thomas Wasow. 1999. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, CA.

Scha, Remko. 1990. Language theory and language technology: Competence and performance. In R. de Kort and G. L. J. Leerdam, editors, *Computertoepassingen in de Neerlandistiek*. Almere, pages 7–22.

Schneider, Nathan. 2010. Computational cognitive morphosemantics: Modeling morphological compositionality in Hebrew verbs with Embodied Construction Grammar. In *Proceedings of the 36th Annual Meeting of the Berkeley Linguistics Society*, Berkeley, CA.

Steedman, Mark. 1996. *Surface Structures and Interpretation*. Number 30 in Linguistic Inquiry Monographs. The MIT Press, Cambridge, MA.

Steels, Luc, editor. 2012a. *Computational Issues in Fluid Construction Grammar*. Number 7249 in Lecture Notes in Computer Science. Springer, Berlin.

Steels, Luc, editor. 2012b. *Experiments in Cultural Language Evolution*. Number 3 in Advances in Interaction Studies. John Benjamins Publishing, Amsterdam.

Tomasello, Michael. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, Cambridge, MA.

Tsarfaty, Reut. 2010. *Relational-Realization Parsing*. PhD thesis, University of Amsterdam.

Nathan Schneider is a doctoral student at Carnegie Mellon University. His research has touched several areas of computational semantics, including an account of Semitic morphology in Embodied Construction Grammar (Schneider 2010). E-mail: `nschneid@cs.cmu.edu`. *Reut Tsarfaty* is a postdoctoral researcher at Uppsala University. In her research she developed the Relational-Realization framework for statistical modeling of flexible form-function correspondence patterns (Tsarfaty 2010). Tsarfaty is a guest editor of the recent *Computational Linguistics* special issue on Parsing Morphologically Rich Languages (PMRL) and she is the author of a book on the topic to be published by Morgan & Claypool Publishers. E-mail: `tsarfaty@stp.lingfil.uu.se`.