Notice

This paper is the author's draft and has now been published officially as:

Gerasymova, Kateryna (2012). Expressing Grammatical Meaning with Morphology: A Case Study for Russian Aspect. In Luc Steels (Ed.), *Computational Issues in Fluid Construction Grammar*, 91–122. Berlin: Springer.

BibTeX:

```
@incollection{gerasymova2012expressing,
   Author = {Gerasymova, Kateryna},
   Title = {Expressing Grammatical Meaning with Morphology:
            A Case Study for Russian Aspect},
   Pages = {91--122},
   Editor = {Steels, Luc},
   Booktitle = {Computational Issues in {Fluid Construction Grammar}},
   Publisher = {Springer},
   Series = {Lecture Notes in Computer Science},
   Volume = {7249},
   Address = {Berlin},
   Year = {2012}}
```

Expressing Grammatical Meaning with Morphology: A Case Study for Russian Aspect

Kateryna Gerasymova

Sony Computer Science Laboratory Paris, France

Abstract. Phrasal structures form the backbone of any sentence, and they provide key information about, respectively, the constituent structure of a sentence and how its meanings are to be used in achieving a communicative purpose. In addition, languages typically feature several other systems that express meaning through grammatical rather than lexical means. Examples are a tense-aspect system, which expresses information about the timing and temporal structure of events, a mood-modality system, which concerns the epistemic status and opinion of the facts reported in a sentence and a determination system, which provides information about the access status of the referent of nominal phrases. This chapter shows how such grammatical meanings are approached within the framework of Fluid Construction Grammar through a concrete example of the Russian aspect system.

1 Introduction

Certainly a large part of the meaning of a sentence is expressed through lexical items, but languages also package a lot more information into the sentence by expressing additional meanings through grammatical devices like morphology and syntax, resulting in the need for a discussion of *grammatical meaning* instead of simply *lexical meaning*. Languages of the world differ widely in terms of which meanings they express grammatically. One language may grammatically express a very specific nuance in meaning that may be completely ignored in the grammar of another language. Here are some examples:

1. To describe events in time, some languages employ tenses to locate situations in the past, present or future from a particular moment in time. In French, a speaker can say “il pleut" (it rains), “il pleuvait" (it rained) and “il pleuvera" (it will rain) so as to relate the event of rain to the moment of speech. In Bamileke-Dschang (spoken in Cameroon), a finer-grained distinction holds that includes past and future tenses of five different degrees of remoteness, such as immediate past and the past within one day from today, whereas in English only the distinction of past/non-past is expressed morphologically [4].
2. Other languages focus more upon how events unfold in time using the grammatical category of aspect. Aspect does not annotate the passage of time in a situation according to an external clock, but reveals the internal timing of

an event by describing its temporal structure, such as, “it rained" versus “it was raining" [3].
3. By means of modality, the status of the proposition that describes the situation can be indicated, as in the German utterance “Ich glaubte, er wäre krank" where the subjunctive mood of“wäre" marks the belief of the speaker.
4. In Japanese, a sophisticated system of honorifics may be used to convey information about the social distance or disparity in rank between speaker and hearer, or to emphasize social intimacy or similarity in rank.
5. In Romance languages such as French, the ubiquitous articles serve as an expression of definiteness, as in “Les fruits que j’ai achetés" (The fruit that I bought), which, in contrast, are completely absent from Japanese or Slavic languages.

This chapter presents a case study that demonstrates how grammatical meanings can be expressed through grammar and how this expression can be represented in constructions for Fluid Construction Grammar [15, 17]. Examples are drawn from Russian aspect. Although the Russian aspectual system is notorious for its complexity, it is possible to crystallize a regular subsystem out of it and to address the issues of grammatical expression of this subsystem, serving the didactic purposes of this chapter. Thus, no attempt is made here to give a comprehensive description of the total verbal system of Russian with its numerous exceptions and grammaticalization processes. Rather, we only address the principle or idea of Russian aspect as a grammatical category.

The ultimate goal of this study is to be able to process dialogues in FCG that appeared in the comprehension experiment of [19], who investigated how Russian children develop their understanding of aspectual forms. Preschool children were interviewed after watching pairs of short movies, each illustrating what would be described by a different aspectual form of the same verb stem. The comprehension of those dialogues was the test condition for the ability to manage aspect. For example, the question Кто нарисовал лицо? (*Kto narisoval lico?*, ‘Who has drawn the face?’) tested whether a child understood the concept of completion of the event of drawing expressed by the perfective aspect. The grammar presented in this chapter constitutes part of a larger study on aspect acquisition [8], which consequently motivated the choice of the test examples.

This chapter assumes that the reader has had a first encounter with FCG, for example by reading [16, 18] and is also acquainted with the templates for implementing phrasal constructions [5, 14]. Section 2 begins with a sketch of the linguistic background in order to build a foundation for the grammar developed later. Section 3 introduces some general design principles on how to organize complex grammars and how to divide labor between constructions. In the next stage, the full grammar is implemented with the help of templates (Section 4), raising new questions in the process, such as how to deal with an unmarked case of imperfective (Section 5). Section 6 briefly outlines the language processing, and, finally, the Appendix provides an insight into what actually happens behind the scenes by offering a tutorial on how to write fully-fledged constructions and how to develop templates for them.

2 Linguistic Insights into Russian Aspect

When modeling a nontrivial linguistic phenomenon, it is crucial to find a linguistic theory capable of providing the necessary grip for its in-depth computational treatment. The analysis in this chapter is based on the view that aspect is a *grammatical* category, manifested in Russian through the contrast between perfective and imperfective.[1] As formulated by [6], perfective aspect expresses the action as a *total event* summed up with reference to a single *juncture*, and imperfective is characterized by the absence of that notion. In Russian, in contrast to many other languages such as Turkish, English or the broader family of Romance languages, it is the perfective rather than the imperfective that is morphologically marked in verbs.

2.1 Dimensions of Aspect

However, the story does not simply end with this basic opposition of perfective versus imperfective. In order to comprehensively describe the Russian aspectual system, another distinction needs to be introduced – the *semantic* category of Aktionsart. Aktionsart expresses additional, often temporal, properties of the event introduced by a verb. For instance, *telic* Aktionsart conveys that the event has an inherent goal or result; *ingressive* Aktionsart profiles the beginning of the event; *delimitative* Aktionsart conveys that the event has a limited time span and so on. The categories of aspect and Aktionsart are linked by the fact that perfective aspect is defined as a means to highlight the boundaries of the event. It is not important which boundaries are profiled (initial, final or both), as long as at least one of them is actually profiled. While the notion of perfectivity does not discriminate between the different possible positions of the boundary, the boundary's position is fundamental for the Aktionsart of the verb [2, 19]. For example, in the verb нарисовать (*narisovat'*, 'draw.PFV') perfective highlights the inherent notion of completeness of the event by focusing on the final boundary, and in the verb заплакать (*zaplakat'*, 'start-crying.PFV'), perfective signals the notion of beginning (the initial boundary), viewing the beginning of *crying* as a single indivisible whole. Imperfective is often connected with the durative Aktionsart, but due to its unmarked nature it is also compatible with a wide range of contexts, even those where most languages would use perfective [20].

Overall, aspect is omnipresent in Russian grammar, and every verb in all forms and tenses is either perfective or imperfective. For instance:

(1) Нечего было^{i} делать^{i}, мы приютились^{p} у
nothing be.PT.IPFV do.INF.IPFV we harbor.PT.PFV.REFL near
огня, закурили^{p} трубки, и скоро чайник
fire smoke.PT.PFV pipe and soon tea kettle
зашипел^{p} приветливо. *Nečego bylo^{i} delat^{i}, my prijutilis^{p} u*
hiss.PT.PFV friendly

[1] Perfective is hereafter indicated as PFV and imperfective as IPFV.

ognja, zakurili^{p} *trubki, i skoro čajnik zašipel*^{p} *privetlivo.*

'There was nothing to do but to make ourselves comfortable by the fire, we lighted up our pipes, and soon the teakettle began to hiss happily.'
[M. Y. Lermontov. Герой нашего времени ('A hero of our time')]

The above sentence exhibits examples of verbs in different aspects and Aktionsarten. The two verbs at the beginning of the sentence are imperfective (indicated by the superscript ^{i}). All the rest are perfectives (indicated by the superscript ^{p}) of various Aktionsarten. For instance, both perfectives закурили^{p} (*zakurili*, 'began to smoke.PFV.PT.1PS.PL') [2] and зашипел^{p} (*zašipel*, 'began to hiss.PFV.PT.3PS.SG') portray the beginning of events of smoking and hissing, respectively, and are of the ingressive Aktionsart.

2.2 Morphology

The morphology of the Russian aspect mirrors the complexity of its semantics. Again, in contrast to other languages like English, there is no single morphological marker that marks either of the two aspects. In English, the progressive aspect is marked with the conjugated 'to be' + infinitive of the verb + '-ing', as in "it is raining" in contrast to "it rains", and thus the progressive is marked unambiguously. In contract to this, Russian verbs can be roughly divided into 'simple' verbs, consisting of a stem and a conjugated ending, such as читать (*čitat'*, 'read.IPFV'), щипать (*ščipat'*, 'pinch.IPFV'), and 'complex' verbs, which are derived from the 'simple' verbs by the addition of aspectual markers, such as by prefixation перечитать (*perečitat'*, 're-read.PFV') and выщипать (*vyščipat'*, 'pinch-out.PFV'). Simple verbs typically describe activities and are imperfective, such as резать (*rezat'*, 'cut.IPFV'). The addition of a prefix changes the aspect of simple verbs into perfective, such as нарезать (*narezat'*, 'cut.PFV'), порезать (*porezat'*, 'cut-for-a-while.PFV'), дорезать (*dorezat'*, 'cut-to-the-end.PFV') and so on, indicated schematically in Figure 1. Russian has nineteen verbal prefixes that productively form perfective [12]. There is also a perfectivizing suffix -ну- leading to such forms as резануть (*rezanut'*, 'cut-once.PFV').

Moreover, Russian verbs can undergo more than one aspectual derivation. After the prefix is added to the simple verb (e.g., думать, *dumat'*, 'think.IPFV') making it perfective (придумать, *pridumat'*, 'invent.PFV'), the so-called imperfectivizing suffixes can flip the verb's aspect to imperfective again, as in придумывать (*pridumyvat'*, 'invent.IPFV'). However, another prefix can be attached to this form, changing the aspect to perfective again – попридумывать (*popridumyvat'*, 'invent-for-a-while.PFV'). This chapter focuses on the first aspectual derivation: the addition of prefixes to simple verbs, which changes them from imperfective to perfective. These forms account for roughly 80 percent of

[2] The following abbreviations are used throughout the chapter: PFV – perfective, IPFV – imperfective, PR – present tense, PT – past tense, FT –future tense, REF – reflexive, PS – person, PL – plural, SG – singular.

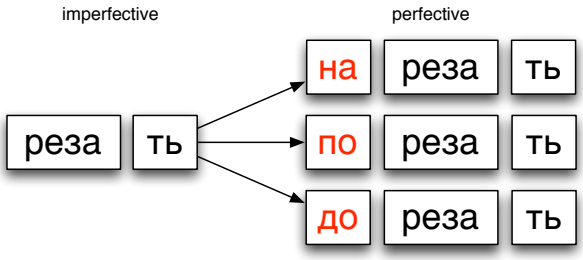

Fig. 1. Schema of the simple verb резать (*rezat'*, 'cut.IPFV') and its perfective derivatives нарезать (*narezat'*, 'cut.PFV'), порезать (*porezat'*, 'cut-for-a-while.PFV') and дорезать (*dorezat'*, 'cut-to-the-end.PFV'), which are formed by prefixation.

the total verb occurrences in written corpora, and, as stated by [7], awareness of the aspectual opposition is therefore likely to be focused precisely on the contrast between simple imperfective and prefixed perfective forms (and not on that of perfective and secondary imperfective).

3 Designing a Grammar

3.1 What Is a Suitable Grammar Architecture?

Rather than immediately start writing FCG constructions, it is important first to develop a general strategy as to how the grammar should be organized. The production procedure starts with a meaning that has to be communicated and tries to construct an utterance that would convey this meaning in a given context. In the following example, the speaker wishes to communicate the event of Michael completing the drawing of a face, which in English might be expressed as "Michael has drawn a face" and in Russian "Миша нарисовал лицо" (*Miša narisoval lico*). The meaning underlying the target utterance can be represented in the following way in the notation of the first-order predicate calculus:

(2)
```
(michael michael-indiv context-1)
(draw draw-ev context-1)
(drawer draw-ev michael-indiv)
(drawee draw-ev face-obj)
(event-type draw-ev complete)
(face face-obj context-1)
(context context-1)
```

The above notation essentially indicates four main components in the context: 1) there is an individual Michael and 2) a draw event (`draw-ev`) which has a

drawer and a drawee, 3) that event is complete, and 4) the object of that event, the drawing, is a face.

When dealing with a nontrivial language subsystem and trying to write constructions to handle it, it is easier to split up the whole constructicon in sets of constructions with similar functions, arriving thereby at a clearer design and division of labor between constructions. It is first of all useful to distinguish between the lexical and grammatical pathways. There are a lot of advantages to such an organization, which will become apparent in the course of the chapter. Let us momentarily shift the focus away from the grammar and toward the information that can already be expressed by purely lexical items. The two predicates `(michael michael-indiv context-1)` and `(face face-obj context-1)` could fall into the responsibility of the lexical constructions that have the corresponding meaning of `michael` and `face` on their semantic poles. (See the introductory chapter in this volume [18].) Similarly, the three predicates

```
(draw draw-ev context-1)
(drawer draw-ev michael-indiv)
(drawee draw-ev face-obj)
```

can be expressed by the lexical entry for the verb “to draw." The meaning left unprocessed is `(event-type draw-ev complete)`, which, in Russian, is grammaticalized and morphologically expressed in the verb. Thus, this predicate should be captured by grammatical constructions different from the lexical ones for the case of Russian, although, in some languages, this meaning can very well be expressed lexically. Now we turn to the question as to which grammatical constructions are needed in order to capture this meaning in a way consistent with the Russian grammar.

As shown in the previous section, the notion of completeness is an integral part of the semantics of some Russian verbs, which are said to belong to the telic Aktionsart. There are other Aktionsarten characterized by the notions of ingressivity, durativity, deliminativity, and so forth. In other words, Aktionsart describes the lexical temporal semantics of a verb and is therefore a semantic category. This principle can be formalized with the help of a special construction, which puts the semantic feature of *completeness* – the predicate `(event-type draw-ev complete)` – in relation to the semantic category of Aktionsart of the corresponding verb. Additionally, the semantic dimension of Aktionsart has to be translated into its grammatical counterpart of aspect by another grammatical construction. For example, for the telic Aktionsart this mapping construction should state that the notion of telicity is grammatically expressed by the perfective aspect. It is then the duty of another kind of construction – the morphological construction – to express the perfective aspect by the attachment of a prefix to a verb stem, with the particular string of a prefix depending on the semantic category of Aktionsart.

The notion of totality characteristic of all perfective verbs does not constitute a part of meaning (which is supposed to come from the world model), it is rather a

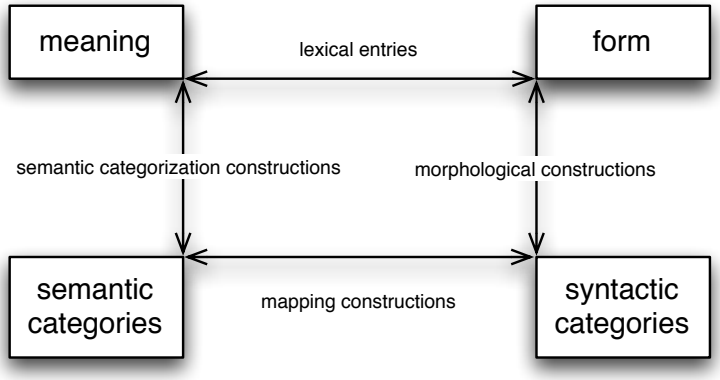

Fig. 2. The grammar square for aspect shows the different relations that grammars need to establish. They are done here by different construction types: lexical, semantic categorization, mapping and morphological. Lexical constructions map lexical stems to their meanings (top, horizontal arrow), semantic categorization constructions re-categorize meaning in terms of semantic categorizations (left arrow), mapping constructions map abstract semantic structures to abstract syntactic structures (bottom arrow), and morphological constructions express syntactic categories using morphology (right arrow).

semantic constraint captured in the semantic category of the corresponding verb. This design decision is motivated by the theory of genesis of aspect proposed by [7], underlining that the "perfectivity" (i.e. the notion of totality) of a prefixed verb is basically nothing more than a by-product of the word-building process out of which the forms with new semantic nuances are derived.

3.2 Division of Labor between Constructions

Writing effective operational constructions is complicated. One has to consider many aspects, such as unification and merging procedures, hierarchical organization, bidirectional applicability and so on. Tackling all of the issues simultaneously is possible only in simple cases. In more complicated ones, it is useful to first look at a construction as a "black box" and attempt to determine its exact behavior resulting from a specific body of input, especially in light of its interaction with other constructions.

Grammatical meaning works through the intermediary of semantic and syntactic categorizations as illustrated in the grammar square (see Figure 2), and for clarity of design, constructions are used that correspond to each of these steps, even though all of them can involve criteria from any level of the grammar. For example, morphological constructions (further called *morph-constructions*)

focus on mapping syntactic categorizations to surface forms, but they may take semantic as well as phonological criteria into account.

Lexical entries provide the base material for further grammatical processing; therefore, they are applied first both in production and parsing. Given a particular stretch of meaning to be expressed, these lexical entries should grab that meaning in production, encapsulate it in a new unit, and associate a word string with that unit. In so doing, the lexical entry for the noun 'face' should trigger in the presence of the meaning `(face ?obj ?ctx)` and associate it with the string "lico", as schematically shown in the Example 3. A similar association applies for the verb 'to draw' with the difference that the meaning consists of the three predicates instead of one (Example 4).

(3) `meaning: (face ?obj ?ctx)` $\longleftrightarrow$ `form: (string "lico")`

(4)
```
meaning:                          form:
(draw ?ev ?ctx)          ⟷       (string "risova")
(drawer ?ev ?drawer)
(drawee ?ev ?drawee)
```

In order to prepare the resulting linguistic structure for the application of grammatical constructions, lexical entries should also introduce additional semantic and/or syntactic categorizations of the unit, thereby providing constraints for the latter to trigger on. For example, if a morphological marker should only be used with verbs, the lexical entry of the verb should supply the information that it is a verb.

We now look at the constructions that establish the relations in the grammar square, focusing first on production to make the construction types easier to understand.

1. After applying lexical constructions, the additional meaning of the event-type should be re-categorized in terms of the language-internal semantic category of Aktionsart and encapsulated in the unit of the verb introduced by the lexicon. This task is undertaken by semantic categorization constructions (called *sem-cat constructions*). A sem-cat construction is needed here, stating that if the event type of the event expressed in a unit is *complete*, then the semantic category 'telic Aktionsart' is added to this unit. We need these sem-cat constructions because this categorization is not always so straightforward and different meanings could map into the same Aktionsart depending on the context. The relation that the sem-cat construction has to put in effect is schematically captured in the following way:
   (5) `meaning: (event-type ?ev complete)` $\longleftrightarrow$
   `sem-cat: (aktionsart telic)`
   Sem-cat constructions could in principle take into account many other aspects of the linguistic context, such as syntactic criteria, but the examples treated here are sufficiently simple so that this is not necessary.

2. The next step in production is the translation of Aktionsarten into their grammatical counterparts of aspect. This is necessary because the same Aktionsart can be mapped to both perfective or imperfective aspect depending on the context. For instance, a distinction between перерисовать (*pererisovat'*, 'redraw.PFV') and перерисовывать (*pererisovyvat'*, 'redraw.IPFV'), which are both of the totalizing Aktionsart, is in aspect (the perfective in the former and imperfective in the latter). Moreover, the notion of *totality* characteristic of all perfective verbs is not yet captured in the transient structure. We achieve this effect through *mapping constructions*, which implement the bottom bi-directional relation of the grammar square. Our example must then have a construction triggering on the semantic category of telic Aktionsart. It should not only link this category to perfective aspect, but also add the feature *totality* characteristic of all perfective verbs. Here is the mapping:
   (6) `sem-cat: (aktionsart telic)`
   `sem-cat: (view ?ev totality)`
   $\longleftrightarrow$ `syn-cat: (aspect perfective)`
   Such mapping constructions implement the core of aspectual grammar by establishing the subtle interplay between Aktionsarten and aspect, thereby achieving a distinguished role for each category: Aktionsart is responsible for the semantics of telicity, whereas perfective aspect is responsible for the notion of totality.
3. As the last step in production, a construction is needed that implements the expression of perfective aspect by means of prefixation. Which one of nineteen prefixes is attached to the verb depends on the semantics of the perfective form, that is, on the Aktionsart. These kinds of constructions are called *morphological constructions* because they settle morphology, even if it may involve taking additional pragmatic, semantic or other linguistic contexts into account. Morph-constructions establish the following relation :
   (7) `sem-cat: (aktionsart telic)`
   $\longleftrightarrow$
   `syn-cat: (aspect perfective)`
   `prefix: (string "na-")`
   Morph-constructions thus provide the missing strings of grammatical markers (similar to the lexical constructions that supply strings for lexical entries) and thereby finalize the production process.

This organization, already proposed in [9], is schematically illustrated in Figure 2. Let us see how these different construction types now operate in parsing:

1. The lexical entries again provide the base material for further grammatical processing, so they apply first. Given a particular word in the utterance, the lexical entry encapsulates it in a new unit and associates the relevant meaning with that unit. Lexical entries also add semantic and syntactic categories to the unit that will be relevant in further grammatical processing.
2. Next, the morph-constructions are applied, because they detect additional form elements in the utterance and translate them into syntactic categories.

For example, they detect that a verb is prefixed and add the relevant syntactic categorizations to the unit that covers the base verb.
3. All syntactic and semantic categorizations are now available to apply the mapping constructions, which map some of the syntactic categories (e.g., perfective aspect) into a semantic categorization (telic Aktionsart), constrained by the syntactic and semantic context.
4. Finally, sem-cat constructions are applied that map the language-internal semantic categories into meaning, in our case Aktionsart, to the event type of the verb.

Hence, parsing uses the same construction sets, but they are ordered in the opposite direction from that in production: clockwise when looking at the grammar square for parsing and counter-clockwise for production (Figure 2).

4 Implementing a Grammar

This section uses templates to introduce an implementation of the grammar design outlined in the previous section. Templates allow the grammar engineers to abstract away from the technical details of the FCG formalism and instead concentrate on linguistic aspects. The Appendix at the end of the paper offers an explanation of how to develop such templates.

Our starting point in production is on the meaning that has to be expressed (as presented in Example 2):

(8)
```
(michael michael-indiv context-1)
(draw draw-ev context-1)
(drawer draw-ev michael-indiv)
(drawee draw-ev face-obj)
(event-type draw-ev complete)
(face face-obj context-1)
(context context-1)
```

4.1 Lexical Constructions

Face-construction. As discussed previously, the lexical construction for the noun 'face' produces the following bidirectional mapping:

(9) `meaning: (face ?obj ?ctx)` $\longleftrightarrow$ `form: (string "lico")`

However, a lexical construction usually has to do more in order to prepare the resulting linguistic structure for the application of grammatical constructions. It should also introduce some additional semantic and/or syntactic categories of the unit, thereby providing constraints for the grammatical constructions to trigger on. Instead of using only one template `def-lex-cxn` specifying everything that is needed to build a construction, the entire task can be split into different templates for handling different issues, as discussed by [14]. Thus, the template

for defining a lexical entry for the noun “face" is shown below, where, in addition to the simple meaning-to-form mapping with `def-lex-skeleton`, there is also a specification of semantic and syntactic categories of the construction with the help of the `def-lex-cat` template. [3]

(10)
```
(def-lex-cxn face-cxn
     (def-lex-skeleton face-cxn
          :meaning (== (face ?obj ?ctx))
          :args (?obj ?ctx)
          :string "lico")

     (def-lex-cat face-cxn
          :sem-cat (==1 (class indiv))
          :syn-cat (==1 (lex-cat noun)
                        (gender neuter)
                        (case ?case))))
```

Draw-construction. The lexical entry construction for the verb рисовать (*risovat'*, ‘draw’) (more precisely, for the stem рисова- because endings of verbs in Russian are subject to conjugation) has to establish the following mapping:

(11)
```
meaning:                          form:
(draw ?ev ?ctx)        ⟷         (string "risova")
(drawer ?ev ?drawer)
(drawee ?ev ?drawee)
```

The corresponding template for defining a draw-construction with additional semantic and syntactic categories appears as the following:

(12)
```
(def-lex-cxn draw-cxn
     (def-lex-skeleton draw-cxn
          :meaning (== (draw ?ev ?ctx)
                       (drawer ?ev ?drawer)
                       (drawee ?ev ?drawee))
          :args (? ev ?ctx)
          :string "risova")
     (def-lex-cat draw-cxn
          :sem-cat
           (==1 (class event)
                (sem-val
           (==1 (agent ?ev ?drawer)
                (patient ?ev ?drawee))))
          :syn-cat (==1 (lex-cat verb)
                        (gender ? gender))))
```

[3] All templates (`def-lex-cxn, def-lex-skeleton and def-lex-cat`) are discussed in detail in [14].

Of particular interest is the verb-specific semantic category of semantic valency (`sem-val`), which contains information regarding who the agent and patient of the event described by the verb are. This valency is used later to establish a grammatical agreement between the subject and the verb in a sentence. [4]

Scaling up the lexicon. Once the language-specific slots for a `def-lex-cxn` template have been worked out, it is very simple to scale up the lexicon. New nouns can be defined as in Examples 13 and 14 and other verbs as in Example 15.

(13)
```
(def-lex-cxn masha-cxn
     (def-lex-skeleton masha-cxn
          :meaning (== (masha ?obj ?ctx))
          :args (?obj ?ctx)
          :string "Masha")
     (def-lex-cat masha-cxn
          :sem-cat (==1 (class indiv))
          :syn-cat (==1 (lex-cat noun)
                        (gender feminine)
                        (case ?case))))
```

(14)
```
(def-lex-cxn letter-cxn
     (def-lex-skeleton letter-cxn
          :meaning (== (letter ?obj ?ctx))
          :args (?obj ?ctx)
          :string "pis'mo")
     (def-lex-cat letter-cxn
          :sem-cat (==1 (class indiv))
          :syn-cat (==1 (lex-cat noun)
                        (gender neuter)
                        (case ?case))))
```

(15)
```
(def-lex-cxn read-cxn
     (def-lex-skeleton read-cxn
          :meaning (== (read ?ev ?ctx)
                       (reader ?ev ?reader)
                       (readee ?ev ?readee))
          :args (?ev ?ctx)
          :string "cita")
     (def-lex-cat read-cxn
          :sem-cat
           (==1 (class event)
                (sem-val
                 (==1 (agent ?ev ?reader)
                      (patient ?ev ?readee))))
          :syn-cat (==1 (lex-cat verb)
                        (gender ?gender))))
```

[4] More about the verb agreement can be found in [21].

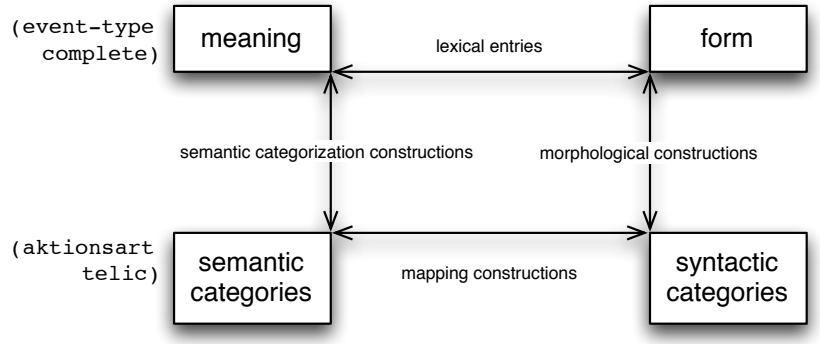

Fig. 3. The semantic construction translates the meaning of the complete event type into the telic Aktionsart and back.

4.2 Sem-cat Constructions

The next step in production is the application of the sem-cat constructions, which translate those parts of the meaning not directly expressed by lexical items into semantic categories that are later mapped onto syntactic features of the utterance, such as morphological markers and word order.

Telic-construction. In the case at hand, the sem-cat construction has to trigger on the meaning `(event-type ?ev complete)` and re-categorize it into the semantic category of telic Aktionsart, as depicted on the grammar square in Figure 3. (If we look at the meaning in Example 8, this predicate is precisely what remained unprocessed after the lexical constructions applied.)

(16)
```
meaning: (event-type ?ev complete) ⟷
  sem-cat: (aktionsart telic)
```

It is very important to note that the meaning of the event type is allocated to the already existing unit of the verb. The sem-cat construction does not create any new units to host this meaning; it rather enhances the verb with the event type and Aktionsart information that is expressed later by means of morphology.

The template `def-sem-cat-cxn` is used to define such constructions. It has a subtemplate called `def-sem-cat-skeleton` which defines the basic relation between meaning and semantic categorization. It also needs the `args` feature to provide a link between the event variable `?ev` of the construction and the one used with the unit in the transient structure by the lexicon based on the meaning of the verb. Thus, the completed form of a template looks as follows:

(17)
```
(def-sem-cat-cxn telicity-sem-cxn
  (def-sem-cat-skeleton telicity-sem-cxn
```

```
:meaning (== (event-type ?ev complete))
:args (?ev ?ctx)
:sem-cat (==1 (aktionsart telic))))
```

Scaling up. The sem-cat constructions for other Aktionsarten can be defined in a way similar to the `telicity-sem-cxn`. The corresponding event-types are represented with analogous predicates, such as `(event-type ?ev begin)` standing for ingressive, `(event-type ?ev finish)` for terminative, `(event-type ?ev for-a-while)` represents delimitative, `(event-type ?ev ongoing)` durative Aktionsarten and so on.

(18)
```
(def-sem-cat-cxn terminative-sem-cxn
  (def-sem-cat-skeleton terminative-sem-cxn
    :meaning (== (event-type ?ev finish))
    :args (?ev ?ctx)
    :sem-cat (==1 (aktionsart terminative))))
```

(19)
```
(def-sem-cat-cxn durative-sem-cxn
  (def-sem-cat-skeleton durative-sem-cxn
    :meaning (== (event-type ?ev ongoing))
    :args (?ev ?ctx)
    :sem-cat (==1 (aktionsart durative))))
```

4.3 Mapping Constructions

The next step in production is the transformation of the abstract semantic categories, which re-conceptualize meaning, into the abstract syntactic categories that are expressed through morphology. For Russian aspect, this transformation is the place where the interplay between semantic and grammatical categories of aspect is captured.

Telic-perfective-construction. For the case of telic Aktionsart, the semantic dimension of telicity has to be translated into its grammatical counterpart of perfective aspect with all the consequences involved. Here is the mapping discussed earlier:

(20)
```
sem-cat: (aktionsart telic)
  sem-cat: (view ?ev totality)
  ⟷ syn-cat: (aspect perfective)
```

As it was the case with the previous construction types, a template called `def-map-cxn` is used to create a mapping construction, which realizes this schematic translation. Instead of defining everything within the body of a single template, the different facets of the mapping construction are captured in several other templates grouped together with `def-map-cxn`. The `def-map-skeleton` is used to realize the basic transformations of categories; the addition of any supplementary categories is delegated to another template. In this case, the basic mapping is the translation of telic Aktionsart into perfective aspect (as long as the unit is

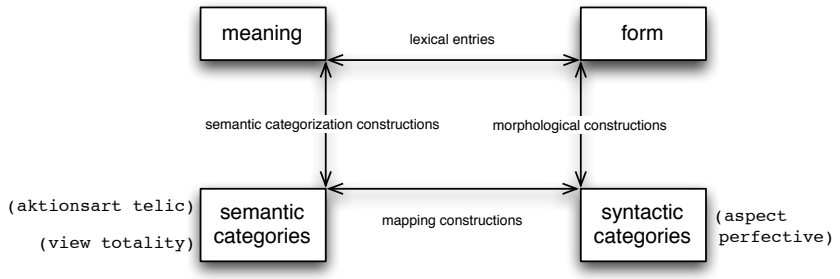

Fig. 4. The mapping telic-perfective-construction couples the semantic category of telic Aktionsart to its syntactic counterpart of perfective aspect, thereby capturing the semantic flavor of totality that is characteristic of all perfective verbs in an additional semantic category – totality view of event.

a verb) and vice versa. Basic here means that one of the categories is triggered during the unification phase and is translated into the other during merging.

In production, the construction is triggered by the presence of the telic verb in the transient structure (assigned by the semantic categorization construction). Whereas in parsing, the construction is triggered by the syntactic category of the perfective aspect (assigned by a morphological construction due to the presence of a prefix). In contrast, the supplementary category of the *event view* is never present in the transient structure and is added by the construction both in parsing and production. The special template `def-map-cxn` is used for this purpose. Summing up thus far, the schematic mapping from Example 20 is equivalent to the following template:

(21)
```
(def-map-cxn telicity-map-cxn
     (def-map-skeleton telicity-map-cxn
       :sem-cat (==1 (aktionsart telic))
       :syn-cat (==1 (lex-cat verb)
                     (aspect perfective)))
     (def-map-impose telicity-map-cxn
       :cxn-sem-cat (==1 (view totality))))
```

Figure 4 summarizes the semantic and syntactic dimensions of aspect, upon which the construction operates, with the help of the grammar square.

Scaling up. Other Aktionsarten correspond to analogous constructions, for example, the delimitative, ingressive and terminative Aktionsarten are also signaled through the perfective aspect. Thus, their mapping constructions differ only in the name of `aktionsart`, as in Example 22. However, the construction for the durative links durative Aktionsart to the imperfective aspect, which lacks

any notion of totality and is an unmarked case, so the structure of the construction will differ as well. Section 5 is dedicated to the discussion of dealing with the unmarked case of imperfective.

(22)
```
(def-map-cxn terminative-map-cxn
     (def-map-skeleton terminative-map-cxn
        :sem-cat (==1 (aktionsart terminative))
        :syn-cat (==1 (lex-cat verb)
                      (aspect perfective)))
     (def-map-impose terminative-map-cxn
        :cxn-sem-cat (==1 (view totality))))
```

Argument structure construction. Because we are considering a complete transitive sentence, there is also a need for a construction that actualizes the argument structure, that is, a construction that 1) equates the referent of the subject with that of the verb's agent, 2) equates the referent of the direct object with the verb's object, as well as 3) settles case assignments of subject and direct object, and, finally, 4) makes the predicate agree with the subject.

Since the implementation of the argument structure is not the focus of this chapter, we only show the phrasal construction used in this example and refer the reader to the earlier work on phrasal constructions [13, 14]. More sophisticated argument structures are discussed in [1, 21].

```
(def-phrasal-cxn transitive-phrase-cxn
    (def-phrasal-skeleton transitive-phrase-cxn
         :phrase
         (?phrase-unit)
         :constituents
         ((?subject-unit)
          (?predicate-unit)
          (?object-unit)))
    (def-phrasal-agreement transitive-phrase-cxn
         (?subject-unit
          :sem-cat (==1 (class indiv))
          :syn-cat (==1 (lex-cat noun)
                        (gender ?agent-gender)
                        (case nominative)))
         (?predicate-unit
          :sem-cat
           (==1 (class event)
                (sem-val ((agent ?ev ?agent)
                          (patient ?ev ?patient))))
          :syn-cat (==1 (lex-cat verb)
                        (gender ?agent-gender)))
         (?object-unit
          :sem-cat (==1 (class indiv))
          :syn-cat (==1 (lex-cat noun)
                        (case accusative))))
    (def-phrasal-linking transitive-phrase-cxn
         (?subject-unit
          :args (?agent ?ctx))
         (?predicate-unit
          :args (?ev ?ctx))
         (?object-unit
          :args (?patient ?ctx))))
```

4.4 Morphological Constructions

The last processing step in production is the application of morphological constructions. Such constructions operate mostly only on the syntactic pole of linguistic structures and specify the surface form of abstract syntactic categories.

Prefix-construction. The morphological constructions expressing the perfective aspect have to determine which prefix out of the possible nineteen prefixes should apply. This decision has also to take semantics into account, illustrating the non-modular nature of grammar. Here is the proposed schematic mapping:

(23)
```
sem-cat: (aktionsart telic)
```
$\longleftrightarrow$
```
syn-cat: (aspect perfective)
prefix: (string "na-")
```

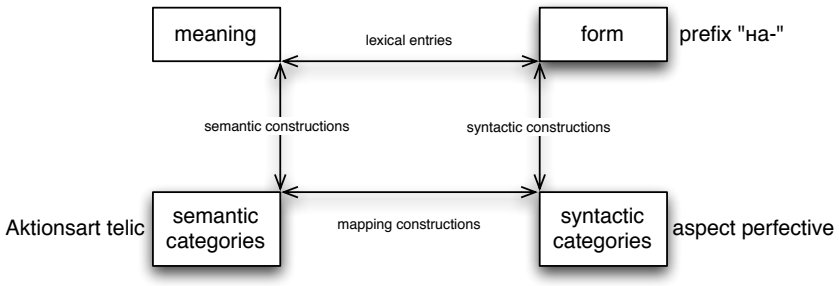

Fig. 5. The prefix-construction specifies the surface expression of telic Aktionsart and perfective aspect through the prefix на- (*na-*).

The template for defining syntactic constructions is called `def-morph-cxn` and is able to take constituents such as prefix, stem and suffix, some of which are optional. It starts by defining a basic skeleton using the template `def-morph-skeleton`. The mapping (23) corresponds to the following template, which specifies the prefix на- (*na-*) and a stem as constituents of a `telic-prefix-cxn` construction, with some constraints put on the latter:

(24)
```
(def-morph-cxn telic-prefix-cxn
  (def-morph-skeleton telic-prefix-cxn
     :prefix "na-"
     :stem
     (?stem-unit
      :sem-cat (==1 (aktionsart telic))
      :syn-cat (==1 (aspect perfective)))))
```

The prefix-construction states that the prefix на- (*na-*) can serve as an expression of the telic Aktionsart, such as in написать (*napisat'*, 'write'), нарвать (*narvat'*, 'cut/pluck/pick'), налгать (*nalgat'*, 'lie') and so on. Other prefixes have similar morphological constructions. What the prefix-construction does in terms of aspectual dimensions is summarized in the grammar square in Figure 5.

Scaling up. Other prefixes can be defined in a similar way, as shown:

(25)
```
(def-morph-cxn terminative-prefix-cxn
  (def-morph-skeleton terminative-prefix-cxn
     :prefix "do-"
     :stem
     (?stem-unit
      :sem-cat (==1 (aktionsart terminative))
      :syn-cat (==1 (aspect perfective)))))
```

Ending-construction. Additionally, all Russian verbs require a conjugated ending to complete their form. The ending is implemented here with the help of a similar template, specifying the ending -л (*-l*), which has to be attached to all masculine verbs in the past tense:[5]

```
(def-morph-cxn masculine-ending-morph-cxn
  (def-morph-skeleton masculine-ending-morph-cxn
     :suffix "-l"
     :stem
     (?stem-unit
      :syn-cat (==1 (lex-cat verb)
                    (gender masculine)))))
```

The feminine ending -ла (*-la*) is defined analogously.

5 Dealing with the Unmarked Case

Imperfective aspect is the unmarked case in Russian grammar. This raises the question of how to treat the unmarked forms in FCG, as pointed out in the previous section. The difficulty, namely, lies in parsing: if in the course of the application of the morphological constructions there is no marker indicating perfectivity, the syntactic category of aspect remains unassigned in the transient structure. The question arises, then, as to how to fill something in based on the *absence* of information.

One can imagine several possible ways of dealing with unmarked cases. A rather unfortunate solution is to have an explicit representation of an absent marker, such as the empty prefix (NIL-prefix), and then to have a morphological construction that triggers on it, assigning the imperfectivity to a verb. Although it is widely used, this solution creates unnecessary search because such null-forms would have to be assumed all the time. The second possible way is to assume the default (unmarked) feature value from the very beginning, so, in the case at hand, the default imperfective aspect would already appear in the definition of the verb stem. To be realizable, this solution requires that linguistic processing is able to override these defaults when a specific grammatical marking is encountered. At this point, FCG does not allow overriding values of features and with good reason, one of which is that although overriding significantly extends the representational power, it also increases the risk in grammar design. More importantly, all the decisions that were made based on the default (such as all constructions applying under the assumption of imperfective) have to be reverted, which creates the need for complex backtracking mechanisms. Such mechanisms are not currently part of the FCG-interpretation process due to the great costs connected to them. What is the alternative solution for handling unmarked cases?

[5] Expression of tense falls out of the scope of the present chapter.

It is important to postpone the decision on a feature value until it is definitely sure that no counter evidence of marking can be expected anymore; and only then is the default case assumed. One way to actualize this approach is by organizing constructions into sets and ordering their application (as discussed in [22], and used in a case study on Hungarian by [1]). Specifically, after the application of the morphological constructions, the system runs special constructions for unmarked cases. The drawback of this solution is that this special construction set is only needed in parsing. Thus, one ends up with the application of different constructions in different processing directions, which is counter to the general design philosophy of FCG. Another solution explored here requires the constructions that need to know about the possibly unmarked feature values to add this information, which is explained in detail below.

Durative-imperfective construction. The mapping construction needed here has to relate the durative Aktionsart to the imperfective aspect. This construction is the first one to notice the lack of grammatical marking in parsing. After all morphological constructions have been tried out and none of them have detected an expression of perfective, the aspect feature on the syntactic pole of the transient structure remains empty due to the unmarked nature of imperfective. At this point the decision concerning the unmarked case can be made with certainty, and the language processing requires the mapping of the *newly assumed* imperfective into the durative Aktionsart. Being the first one to apply at this point, our construction, besides the mapping realization, has to also fill in the default case of the imperfective aspect. Since imperfective is never present in unification, it cannot be specified in the skeleton of the construction and has to be added afterwards by using the `def-map-impose` template which adds the information about the imperfective aspect as illustrated in Example 26:

(26)
```
(def-map-cxn durativity-map-cxn
     (def-map-skeleton durativity-map-cxn
          :sem-cat (==1 (aktionsart durative))
          :syn-cat (==1 (lex-cat verb)))
     (def-map-impose durativity-map-cxn
          :cxn-syn-cat
             (==1 (aspect imperfective))))
```

In case there are many constructions that require the information supplied by such an unmarked case, this solution implies that all of them make the decision separately upon application. This redundancy is not problematic here, since in the case at hand, only the durative has to be translated to imperfective. In other cases, however, this way of solving the default case may become less elegant.

6 Language Processing

The goal of this section is to examine, in more detail, how all the constructions introduced in earlier sections apply in production and in parsing.

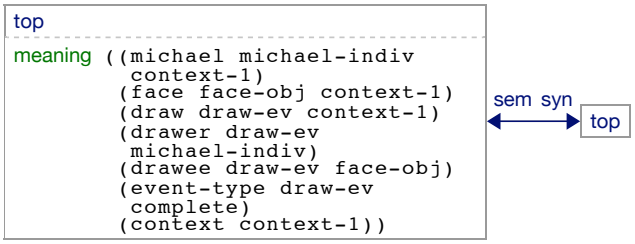

Fig. 6. Initial transient structure in production. Its semantic pole contains the meaning that has to be expressed; the syntactic pole is empty.

6.1 Production

At the beginning of production, the FCG engine creates an initial linguistic structure, which is a meaning-form mapping similar to constructions, as shown in Figure 6: the semantic pole contains the meaning that has to be expressed, and the syntactic pole is empty so far. In the process of production this linguistic structure gradually becomes enhanced with other linguistic information, especially on the syntactic side, finally creating an utterance as an outcome of production.

The first construction set to apply in production is the lexical entries set. Upon its application, each lexical construction creates a new unit in both poles of the transient structure and relocates there all the relevant information of the found word.[6] After all the lexical entries that could be applied – in our example, these are the face-, michael- and draw-constructions – have been applied, the current transient structure contains three units hanging from the top-unit, each corresponding to one lexical stem, as shown in Figure 7.

This way, starting from the initial structure in Figure 6 with an empty syntactic pole, the FCG engine gradually enhances the transient structure by trying out different lexical, semantic, mapping and finally morphological constructions until no more constructions can be applied. Figure 8 shows the syntactic pole of the resulting structure. The head of the hierarchy builds the `top-unit`, followed by the `phrase-unit` that was created by the argument structure construction to capture all the constituents of the transitive sentence under one parent. The units for each of the constituents were established by the lexical constructions; in the course of production other grammatical constructions have gradually filled them with linguistic information. In later stages, the morphological constructions have attached two units to the verb with information about the prefix and ending. The final linguistic structure is rendered into the utterance *Misha na-*

[6] The detailed application of lexical constructions is covered in [14, 16].

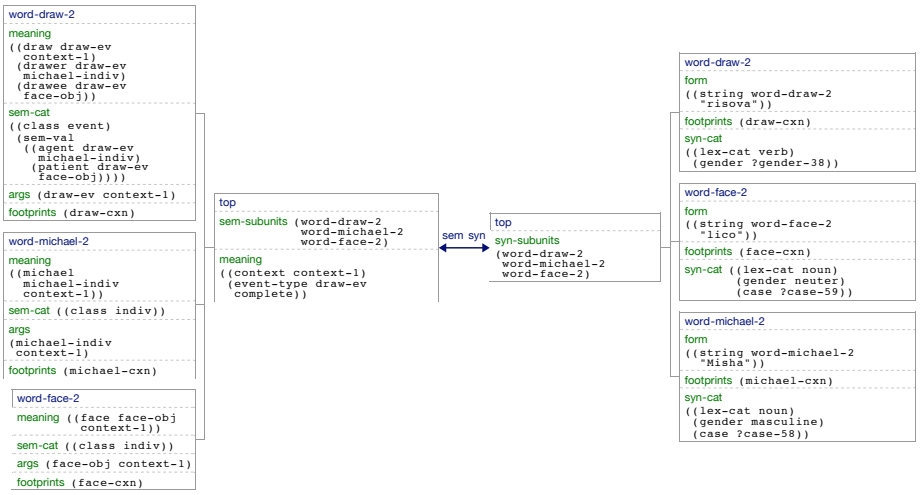

Fig. 7. Current transient structure after the application of all lexical entries. Each lexical entry creates a new unit in both poles of the linguistic structure and relocates all the relevant information to it.

risova -l lico ('Misha has drawn a face'), which was the target of the production process.

6.2 Parsing

The great advantage of FCG is that in parsing the exact same events occur as in production except for the direction of the construction application. The parsing process is initiated by an agent's perceiving an utterance: *Misha na-risova -l lico*. This information is captured by the FCG system in the initial coupled feature structure (Figure 9): on the syntactic side the top unit consists of parsed strings and ordering constraints explicating which string meets which, while the semantic pole remains empty. Note the mirroring of poles when compared to initial structure in production in Figure 6. From this stage, the system constructs the meaning of the observed utterance by simply reversing the order of the construction application: the unification takes place on the syntactic pole followed by the merging of the semantic pole (and the syntactic pole). This difference leads also to the reversed order of application of the various types of constructions: lexical constructions still come first but are immediately followed by the morphological constructions. This order is necessary because these two construction types provide the syntactic information that is required by mapping constructions for determining a verb's Aktionsart and aspect. Finally, once the

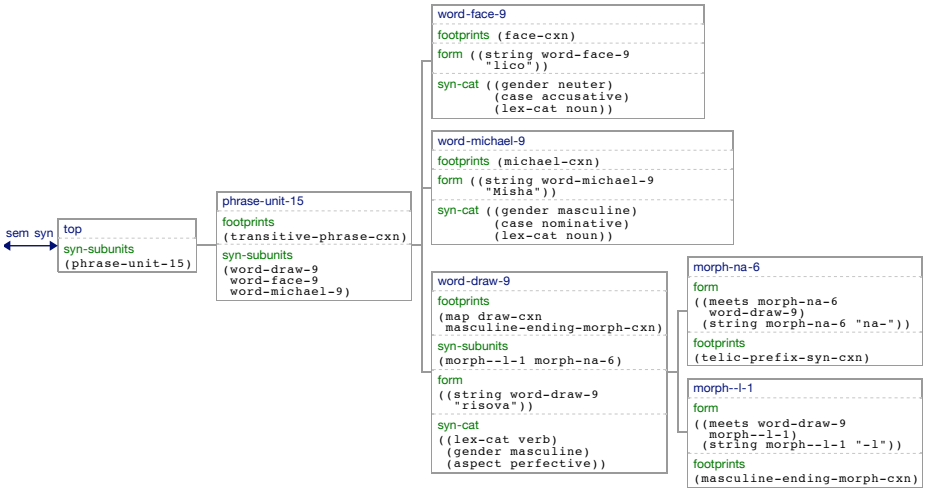

Fig. 8. The syntactic pole of the transient structure at the end of production. It is rendered into the utterance *Misha na- risova -l lico* ('Misha has drawn a face').

mapping constructions have been applied, the semantic constructions are able to reveal the meaning encoded in the semantic categories. Thus, in production, the movement along the grammar square (Figure 2) is clockwise.

Analogous to production, the application of available constructions enhances the initial transient structure with an empty semantic pole to the final transient structure in Figure 10, which contains elaborate semantics. This transient structure codes for the underlying meaning of the perceived utterance *Misha narisova -l lico* (Миша нарисовал лицо), which is the combination of all meaning features of its units:

(27) `(michael ?drawer ?ctx)`
`(draw ?ev ?ctx)`
`(drawer ?ev ?drawer)`
`(drawee ?ev ?drawee)`
`(event-type ?ev complete)`
`(face ?drawee ?ctx)`
`(context ?ctx)`

It is important to note that the notion of totality is not directly represented in the meaning, but is instead captured as a semantic constraint of an event view `(sem-cat (==1 (view totality)))`.

The final transient structure of parsing is structurally identical to the final transient structure of production with the only difference being that it contains

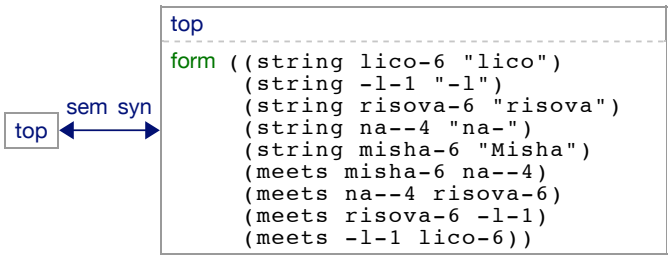

Fig. 9. Initial feature structure by parsing of *Misha na- risova -l lico*.

some variables. The reason for this difference is that in production the meaning comes out of the world model, and everything is already instantiated with concrete entities from the context, whereas in parsing the resulting meaning is anchored to the world only during the interpretation process which starts after parsing.

7 Discussion

The presented organization of the grammar into different sets of constructions provides not only a mechanism for setting decision points, necessary, for example, for dealing with the unmarked case, but it also has important implications on flexibility. In case of uncertainty and communicative problems, the inability to process parts of the utterance does not inhibit the processing of the utterance as a whole. For instance, when encountering unfamiliar or missing grammatical markers, the ability to process the lexicon already gives the possibility to be partially understood.

The division into different types of constructions is also helpful for organizing the learning process that is the target of the current research. Significant is that the presented construction sets exhibit different levels of abstraction. That is, they can be subsumed in different sets not only in terms of their functionality but also in terms of abstractness, with the lexical entries being much less abstract than the most abstract mapping constructions. During the acquisition process, lexical constructions can be learned independently of the complex aspect system; aspect markers can be learned first in an ad hoc way, and then the more abstract and more difficult to learn categories can be acquired [8].

Another approach for grammar organization is discussed in [22]. It shows how families of related constructions can be organized in a network-based relationship, how this organization is useful during linguistic processing and how it can be learned by the FCG-engine during the linguistic processing. The latter point,

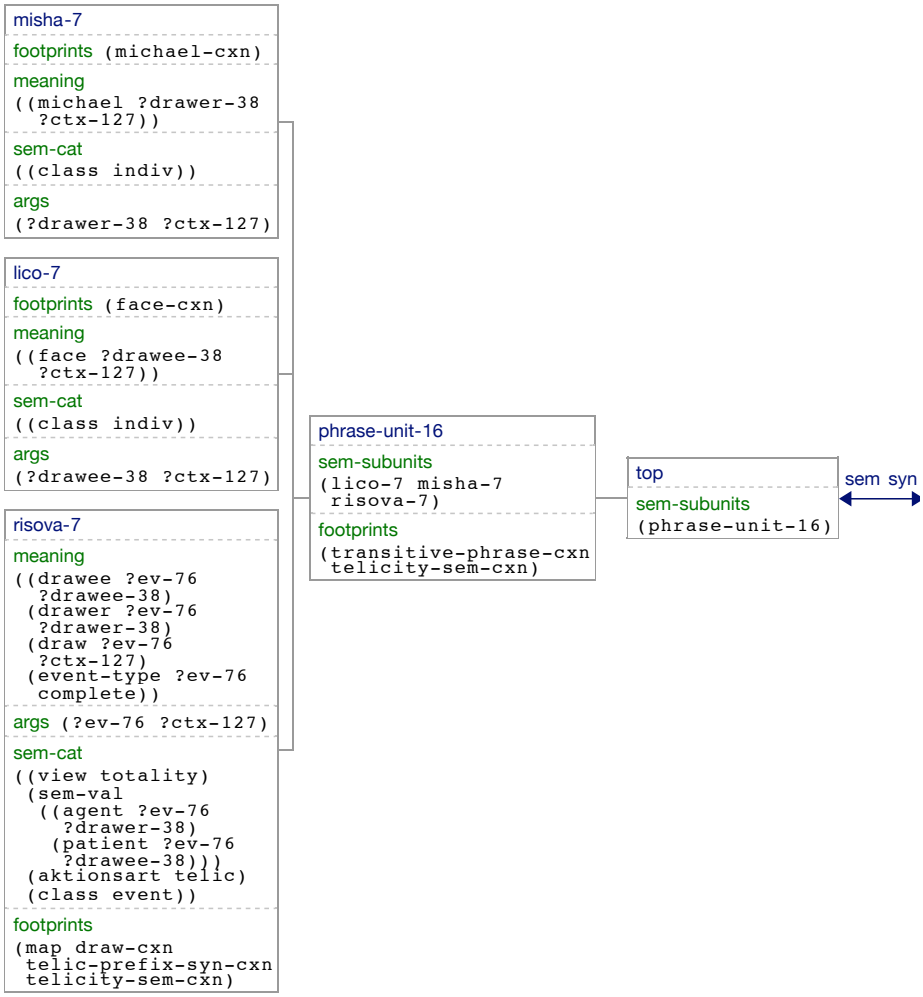

Fig. 10. The semantic pole of the final structure built in parsing of *Misha na- risova -l lico*. The underlying meaning is in Example (27).

that the constructicon organization can stem from the linguistic processing itself, demonstrates that the division of labor between constructions as shown in this chapter is not a pure artifact imposed by the grammar architect.

Overall, the presented chapter is relevant to some current questions in linguistics. One interesting issue of speculation for linguists is what in a language is learned in an individual instance and what is represented in a rule-based fashion.

The grammar presented in this chapter comprised prefixes that are likely to be learned as a rule, or in our terminology as semantic item-based constructions. However, there are many inconsistent cases in Russian, which is also the reason for the difficulties that aspect causes to language learners. There is essentially no one-to-one mapping between prefixes and Aktionsarten: one can say that each verb basically *decides* itself how to interpret a particular prefix, which in conjunction with 19 existing prefixes generates a terrifying number of cases that have to be memorized. To account for this complexity, the presented didactic example could be extended with respect to the intertwinement between prefixes and verbs. One possible way to enact this conjoining is to enhance constructions for verbs with the information about the prefixes they employ in order to build different Aktionsarten. Another alternative would be to create separate holophrastic constructions for each of the idiosyncratic perfective forms.

A further issue concerning Russian aspect that intrigues linguists is that not all verbs behave the same way with respect to aspectual derivation. Recent studies in cognitive linguistics suggested that it is the lexical meaning of a verb that constrains its possibilities for deriving different Aktionsarten [10, 11]. For instance, the verb пахнуть^{i} (*pachnut'*, 'smell') is inherently atelic and cannot derive the telic Aktionsart. To account for these constraints, the presented grammar could be extended to incorporate a new category – *potential* to derive a particular aspectual form – into the knowledge about a verb.

8 Conclusion

This chapter presented a case study of the Russian aspect as a didactic example demonstrating how to deal with grammatical meaning in FCG. The reader was introduced to a general methodology for designing complex grammars and dividing labor between constructions. The success of the approach was highlighted by demonstrating the grammar in operation using example dialogues that were produced and parsed. During the development of aspect grammar, the case of imperfective raised the issue of unmarked forms, which was plausibly solved on the basis of the current grammar organization. The key to the solution was the division of the constructicon into different construction sets providing potential decision points for influencing the language processing. When developing grammars for other grammatical categories, the design described in this study can aid grammarians in their decision process, especially for those domains that are expressed morphologically or feature unmarked forms.

Acknowledgements

Research reported in this chapter was partly funded by the EU project ALEAR and carried out at the Sony Computer Science Laboratory in Paris. I thank the whole team working on FCG at the University of Brussels (VUB AI lab) and at Sony CSL for their contributions in making FCG such a superb environment for doing sophisticated experiments in construction grammar.

Bibliography

[1] Beuls, K.: Construction sets and unmarked forms: A case study for Hungarian verbal agreement. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)
[2] Bickel, B.: Aspectual scope and the difference between logical and semantic representation. Lingua 102, 115–131 (1997)
[3] Comrie, B.: Aspect: An Introduction to the Study of Verbal Aspect and Related Problems. Cambridge: Cambridge University Press (1976)
[4] Comrie, B.: Tense. Cambridge: Cambridge University Press (1985)
[5] De Beule, J., Steels, L.: Hierarchy in fluid construction grammar. In: Furbach, U. (ed.) Proceedings of the 28th Annual German Conference on Artificial Intelligence. Lecture Notes in Artificial Intelligence, vol. 3698, pp. 1–15. Springer Verlag, Berlin, Germany (2005)
[6] Forsyth, J.: A Grammar of Aspect: Usage and Meaning in the Russian Verb. Cambridge: Cambridge University Press (1970)
[7] Forsyth, J.: The nature and development of the aspectual opposition in the Russian verb. The Slavonic and East European Review 50(121), 493–506 (1972)
[8] Gerasymova, K., Spranger, M.: Acquisition of Grammar in Autonomous Artificial Systems. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-2010) (2010)
[9] Gerasymova, K., Steels, L., van Trijp, R.: Aspectual morphology of Russian verbs in Fluid Construction Grammar. Proceedings of the 31th Annual Conference of the Cognitive Science Society (2009)
[10] Janda, L.A.: Aspectual clusters of Russian verbs. Studies in Language 31(3), 607–648 (2007)
[11] Janda, L.A.: Semantic motivations for aspectual clusters of Russian verbs. American Contributions to the XIV International Congress of Slavists p. 22 (2008)
[12] Krongauz, M.A.: Pristavki i glagoly v russkom jazyke: semantičeskaja grammatika. Moscow: Jazyki russkoj kul'tury (1998)
[13] Steels, L., De Beule, J., Neubauer, N.: Linking in Fluid Construction Grammars. In: Proceedings of BNAIC. pp. 11–18. Transactions of the Belgian Royal Society of Arts and Sciences, Brussels (2005)
[14] Steels, L.: A design pattern for phrasal constructions. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)
[15] Steels, L. (ed.): Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)
[16] Steels, L.: A first encounter with Fluid Construction Grammar. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)
[17] Steels, L. (ed.): Computational Issues in Fluid Construction Grammar. Springer Verlag, Berlin (2012)

[18] Steels, L.: Design methods for Fluid Construction Grammar. In: Steels, L. (ed.) Computational Issues in Fluid Construction Grammar. Springer Verlag, Berlin (2012)
[19] Stoll, S.: The role of aktionsart in the acquisition of Russian aspect. First Language 18(54), 351–376 (Oct 1998)
[20] Stoll, S.: Beginning and end in the acquisition of the perfective aspect in Russian. Journal of Child Language 32(4), 805–825 (2005)
[21] van Trijp, R.: A design pattern for argument structure constructions. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)
[22] Wellens, P.: Organizing constructions in networks. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)

Appendix: Defining a Grammar or How to Write Templates

Templates are useful abstractions; however, they hide what happens behind the scenes. This section is targeted to those readers who are interested in the technical details of the implementation. For an example of a semantic categorization construction, a general methodology of how to develop templates is illustrated in detail. The reader can get acquainted with how to write real FCG-constructions that serve a particular function and, thereafter, how to turn them into self-defined templates.

Methodology for writing templates. Templates can be very large or quite small, complicated or basic, highly specialized or multi-purpose, but the development of almost all of them goes through the following four steps:

1. The linguistic dimensions that play a key role are identified.
2. An example of a complete construction is developed into which a future template expands.
3. When scaling up, the changeable elements of a construction are identified.
4. A template emerges as a parametrized version of a construction.

Taking an example of a semantic categorization construction for a case of the telic Aktionsart, let us develop the template `def-sem-cat-cxn` used in the previous section to define all semantic categorization constructions.

1. Key linguistic dimensions. Semantic categorization constructions have to re-categorize parts of the meaning into the language-internal semantic categories. Hence, they operate on the dimensions of meaning and sem-cats. For the case of the telic Aktionsart, the schematic mapping was already identified in Section 3.2 as the following:

(28)
```
meaning: (event-type ?ev complete) ⟷
   sem-cat: (aktionsart telic)
```

2. Writing a complete construction. By this time the reader is familiar with several FCG-constructions. Our telic-construction differs from all the previous ones in that it operates only on one pole of the linguistic structure to which it applies, namely, only on the semantic pole. This is the case because the transformation it realizes – translation of meaning into semantic categories and back – affects only semantics.

It is important to note that our telic-construction still contains two coupled poles; yet, both its left and its right poles refer to the semantic pole of a linguistic structure.

In building a construction, let us start by creating a skeleton and pinning down the meaning it should trigger on. After the application of the lexical constructions, the information about the event type is still located in the top of the linguistic structure (Figure 7), so the construction has to search for it there. Additionally, we tag this meaning to make it movable in order to allocate it later to an appropriate unit.

```
((?top-unit
  (tag ?meaning (meaning  (== (event-type ?ev complete))))))
<-->
((?top-unit))
```

There are some things to take into account before we relate this meaning to the telic Aktionsart. First, the preliminary construction shown above has to be constrained in a way so that it unifies with only specific kinds of structures. Namely, we want it to apply only in the presence of a verb because this is where the aspectual information about the internal structure of events is expressed in Russian. Thus, we extend the construction by the desired hierarchical structure, i.e., the top unit should contain a subunit (`?ev-unit`), which is responsible for an event. To assure that it is an event, we constrain the sem-category of this unit to the `(class event)`, under the assumption that the lexical entry for this verb has already applied creating this separate unit. The extended version looks as follows:

```
((?top-unit
  (tag ?meaning
       (meaning (== (event-type ?ev complete))))
  (sem-subunits (== ?ev-unit)))
 (?ev-unit
  (sem-cat  (==1 (class event)))))
<-->
((?top-unit
  (sem-subunits (== ?ev-unit))))
```

Now the time has come to translate the *completeness* of the event into the category of the telic Aktionsart of the corresponding verb. For this we have to add the sem-cat of Aktionsart to the event unit and move the corresponding meaning of event-type from the top-unit into it.

Important to note is that this construction actualizes merging of the new information into an existing structure. It accesses the substructure and alters it, augmenting the existing unit with some parts of meaning and a semantic category. To specify the semantic category of Aktionsart, we simply add it to the event unit on the *right* pole of the construction, because this is where the merging phase takes place in production:

```
((?top-unit
  (tag ?meaning
       (meaning (== (event-type ?ev complete))))
  (sem-subunits (== ?ev-unit)))
 (?ev-unit
  (sem-cat (==1 (class event)))))
<-->
((?top-unit
  (sem-subunits (== ?ev-unit)))
 (?ev-unit
  (sem-cat (==1 (aktionsart telic)))))
```

On the other hand, the addition of the tagged meaning on the *left* pole has to be done with the help of the J-operator. This addition cannot be done here by merging as well due to the nature of parsing. When writing FCG-constructions, one has to consider the flow of information in both application directions in order to ensure bi-directionality. With regard to production in the previous case, the information about the telic Aktionsart would already be present in the event unit after the application of the morphological construction: the construction for the prefix will already have translated the prefix into the telic Aktionsart of the prefixed verb. Thus, when the right pole of our semantic construction unifies with the linguistic structure, the presence of the sem-cat of the telic Aktionsart has to serve as a constraint in order to ensure that the translation only occurs by those verbs that are telic. Therefore, this condition is specified in the sem-cat of the event unit on the right pole. However, with respect to the current case of parsing, at this point in parsing there is no information about the meaning – it is specifically the job of the semantic construction to add meaning to the transient structure. The meaning cannot simply be put on the right pole of the event unit as a condition; it has to be added with the J-operator:

```
((?top-unit
  (tag ?meaning
       (meaning (== (event-type ?ev complete))))
  (sem-subunits (== ?ev-unit)))
 (?ev-unit
  (sem-cat (==1 (class event))))
  ((J ?ev-unit)
   ?meaning))
<-->
```

```
((?top-unit
  (sem-subunits (== ?ev-unit)))
 (?ev-unit
  (sem-cat (==1 (aktionsart telic)))))
```

The next step in building our telic-construction is the addition of footprints. Normally, the footprints are put in the same unit once with an excludes (==0) and once with an includes operator (==1) on each pole of the construction, respectively. The excludes first ensures that the construction applies for the very first time, and the includes leaves a mark (usually, the name) after the construction's application. In our case, however, the footprints have to be attached to separate units on the left and right poles because both refer to the semantic pole, and hence the added mark on one pole would cause conflicts by the merging of the other pole with an excludes operator and vice versa. The footprints are attached on the left pole to the event-unit and on the right pole to the top-unit:

```
((?top-unit
  (tag ?meaning
       (meaning (== (event-type ?ev complete))))
  (sem-subunits (== ?ev-unit)))
 (?ev-unit
  (sem-cat (==1 (class event)))
   (footprints (==0 telicity-sem-cxn)))
 ((J ?ev-unit)
  ?meaning
   (footprints (==1 telicity-sem-cxn))))
<-->
((?top-unit
  (sem-subunits (== ?ev-unit))
   (footprints (==0 telicity-sem-cxn)))
 (?ev-unit
  (sem-cat (==1 (aktionsart telic))))
  ((J ?top-unit)
   (footprints (==1 telicity-sem-cxn)))))
```

Our construction is already fully operational and behaves in the desired manner during production. However, in parsing a small detail is still missing. What the present construction lacks is that the variable `?ev` in the event-type predicate refers to the same event as the one in the event-unit `?ev-unit`. Up until now there was nothing linking the two, which means that nothing stated that *this particular event* is of the type complete, although the event-type predicate was moved into the event-unit, and the variable names happened to be the same locally in both feature structures. The gap can be closed with the help of the `args` feature. Recall that the lexical entry for event has specified its arguments in the list `(?ev ?ctx)`, the purpose of which has hitherto remained mysterious. It can now be made use of, in order to specify the variables' equality by means

of referring to both the event argument of the event unit and the event of the event-type predicate with the same name *within the same construction*. In this way, we arrive at the following telic-construction:

```
((?top-unit
  (tag ?meaning
       (meaning (== (event-type ?ev complete))))
  (sem-subunits (== ?ev-unit)))
 (?ev-unit
  (sem-cat (==1 (class event)))
  (args (?ev ?ctx))
  (footprints (==0  telicity-sem-cxn)))
 ((J ?ev-unit)
  ?meaning
  (footprints (==1  telicity-sem-cxn))))
<-->
((?top-unit
  (sem-subunits (== ?ev-unit))
  (footprints (==0  telicity-sem-cxn)))
 (?ev-unit
  (sem-cat (==1 (aktionsart telic))))
  ((J ?top-unit)
  (footprints (==1 telicity-sem-cxn))))
```

Now the construction is complete. Note that it is concerned neither with perfective aspect nor the notion of totality characteristic to all perfective verbs. This design decision underlines that Aktionsarten alone are not responsible for the emergence of the grammatical aspect. Another point to note is that this telic-construction is item-based; item being a specific kind of temporal semantics fixed for a construction, whereas the event that this temporal semantics refers to is unspecified and represented as a slot to fill in by a verb.

3. Identifying the pattern. To discover a pattern, one has to consider what the semantic categorization constructions for other Aktionsarten look like. As already mentioned, other semantic nuances are also represented with the predicate of event type, such as `(event-type begin)`. Respectively, they get mapped onto different Aktionsarten, such as the notion of beginning onto the `(aktionsart ingressive)`. Thus, these elements of a construction differ for other Aktionsarten. The rest of the construction's structure remains the same except for its name and the event variable present in the `:args` feature.

4. Template def-sem-cxn As soon as the changeable elements of a construction definition are known, a construction can be easily converted into a template, where concrete values are substituted by parameters that are supplied later by a template. In the case at hand, the feature value components for `meaning, sem-cat, args` as well as the name of the construction will be different for different constructions, thus they are turned into parameters in a

template definition (i.e., the underlined list). [7] When the template is called, these parameters are substituted inside the construction by the actual values supplied in the call. Shown below is the template definition with substituted parameters indicated in bold:

```
(defmacro def-sem-cat-cxn (name &key meaning args sem-cat)
  `(((?top-unit
      (tag ?meaning (meaning ,meaning))
      (sem-subunits (== ?unit-name)))
     (?unit-name
      (sem-cat (==1 (class event)))
      (args ,args)
      (footprints (==0 ,name)))
     ((J ?unit-name)
      ?meaning
      (footprints (==1 ,name))))
    <-->
    ((?top-unit
      (sem-subunits (== ?unit-name))
      (footprints (==0 ,name)))
     (?unit-name
      (sem-cat ,sem-cat))
     ((J ?top-unit)
      (footprints (==1 ,name))))))
```

With the `def-sem-cat-cxn` template, the definition of the entire telic-construction can be folded into the following call:

(29)
```
(def-sem-cat-cxn telicity-sem-cxn
  :meaning (== (event-type ?ev complete))
  :args (?ev ?ctx)
  :sem-cat (==1 (aktionsart telic)))
```

[7] &key means that parameters are named.