A generative perspective on verbs and their readings

Manfred Stede*
Technische Universität Berlin
University of Toronto

Abstract

We sketch the architecture of a sentence generation module that maps a language-neutral "deep" representation to a language-specific sentence-semantic specification, which is given to a front-end generator. Lexicalization is the main instrument for the mapping step, and we examine the role of verb semantics in the process. In particular, we propose a set of rules that derive a range of verb alternations from a single base form, which is one source of lexical paraphrasing in the system.

1 Overview

This paper examines the role of several aspects of verb semantics for natural language generation. We assume a scenario of producing sentences in multiple languages from a common underlying representation—although the problems of multilinguality will not be an explicit topic here. Section 2 briefly describes the architecture of an implemented generator, called MOOSE, and characterizes the two levels of representation involved: a language-neutral level for representing various kinds of events, and a language-specific level of semantic sentence specification, which largely corresponds to the level of 'sentence plans' used in systems based on a linguistic upper model, specifically PENMAN [Bateman et al. 1990]. The lexicon of a target language is the primary vehicle for mapping from one level to the other. Then, sections 3 and 4 discuss our primary topic: verb semantics. First, a treatment of several *Aktionsart* features is proposed, and the role of *valency* in the upper model idea is examined, and improvements suggested. Section 4 proposes a set of rules for systematically computing *alternations* (or *diatheses*) of a verb from a single base form.

*For their helpful comments and suggestions for improving this paper, I thank two anonymous reviewers. Author's address: TU Berlin, Projekt KIT-VM11, Sekr. FR 5-12, Franklinstr. 28/29, D-10587 Berlin, Germany. Email: stede@cs.tu-berlin.de

2 Two-step sentence generation

The MOOSE sentence generator grew out of experiences with building the TECHDOC system [Rösner, Stede 1994], which produces instructional text in multiple languages from a common representation. Specifically, MOOSE accounts for the fact that *events* can receive different verbalizations even in closely related languages such as English and German. It is designed as a sentence generation module that pays attention to language-specific lexical idiosyncrasies, and that can be incorporated into a larger-scale text generator.

2.1 MOOSE in a nutshell

For this brief description of the system architecture, see figure 1. The generator assumes a language-neutral level of event representation, the situation specification or *SitSpec*. Using parts of the target lexicon (see section 2.3), the lexical options for verbalizing the SitSpec are determined. For verbs, the applicable alternations and extensions are computed (see section 4) and added to the set of options. Then a language-specific semantic specification *SemSpec* is constructed in accordance with generation parameters pertaining to brevity and stylistic features. The SemSpec is then handed over to a surface generator: Penman [Penman Group 1989] for English, and a variant developed at FAW Ulm for German. As opposed to the 'traditional' Penman idea, the domain model in which the input SitSpec is represented has been de-coupled from the linguistic upper model, in order to achieve variety in verbalization that would otherwise not be possible [Stede and Grote 1995]. MOOSE is implemented in Macintosh Common Lisp and uses MacPenman; a full description of the system is given in [Stede 1996].

2.2 Levels of representation

A central assumption of the research reported here is that the "deepest" level of representation is in general *not* a linguistic representation; instead, we assume a domain model of some sort, implemented in a KR language. Thus, an explicit transition between instantiated domain knowledge and a language-specific semantic sentence representation is seen as the central step in generation.

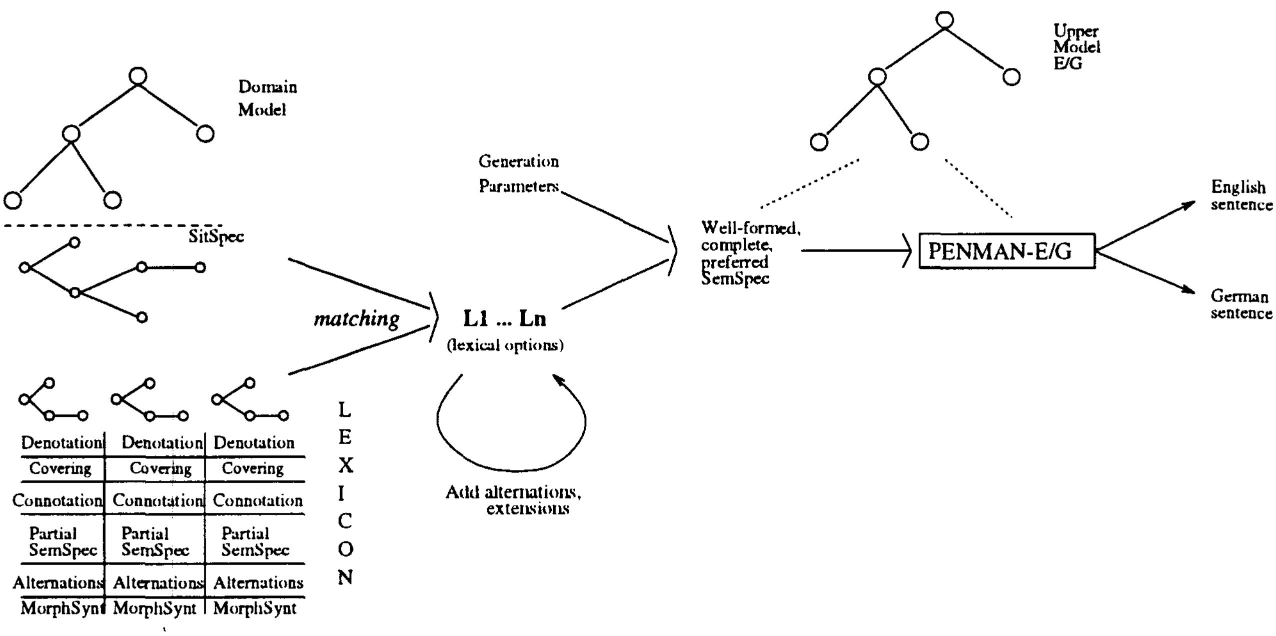

Figure 1: MOOSE system architecture

SitSpec A SitSpec is meant to be neutral between the target languages and between particular paraphrases. It is organized along a variant of the ontological categories proposed by Vendler [1967] and developed further, *inter alia*, by Bach [1986]. We have extended Bach's ontology by breaking up *events* so that their internal structure is explicitly represented (similar to Pustejovsky's [1991] proposal): An event is composed of a *pre-state* (holding before the event commences), a *post-state* (holding when the event is over), and an optional *activity* that brings the transition about. An event without such an activity is a mere state *transition*, e.g., *The room lit up*. An event including an activity is a *culmination*; as an example, consider the event of oil draining from an engine, which is given here in an abbreviated KL-ONE notation (roles names in capital letters, instance names in lower-case):

```
(event1 (PRE-STATE
          (fill-state1 (VALUE 'not-empty)
                       (CONTAINER engine1)
                       (CONTENT oil1))
        (ACTIVITY
          (move-1 (OBJECT oil1)
                  (PATH (path1 (DESTINATION tank1)))))
        (POST-STATE
          (fill-state2 (VALUE 'empty)
                       (CONTAINER engine1))))
```

Figure 2 shows the overall taxonomy of situation types. Subsumed by the general ontological system, a domain model is defined that holds the concepts relevant for representing situations in a technical sample domain and that specifies the exact conditions for the well-formedness of situations. It is implemented in the KL-ONE language LOOM [MacGregor, Bates 1987].

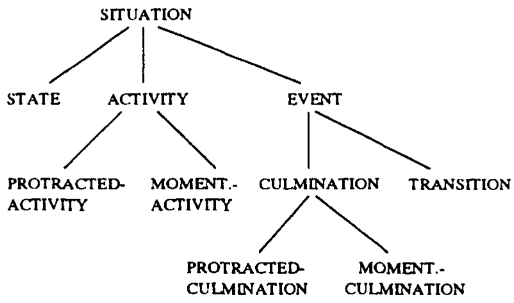

Figure 2: Situation types in the ontology of MOOSE

SemSpec The level of SemSpecs is motivated by the notion of "upper modelling" [Bateman et al. 1990] and is a subset of the input representation language that was developed for Penman, the sentence plan language (SPL) [Kasper 1989]. As opposed to a general SPL term, a SemSpec must contain only upper model concepts and no domain concepts—recall that the domain model in MOOSE is not subsumed by the upper model. Furthermore, since our system takes lexicalization as the decisive task in mapping a SitSpec to a SemSpec, the UM concepts referred to in a SemSpec must be annotated with `:lex` expressions; thus, a SemSpec is a lexicalized structure. Accordingly, we see the upper model as a taxonomy of lexical classes.

SemSpecs are constructed from SitSpecs by select-

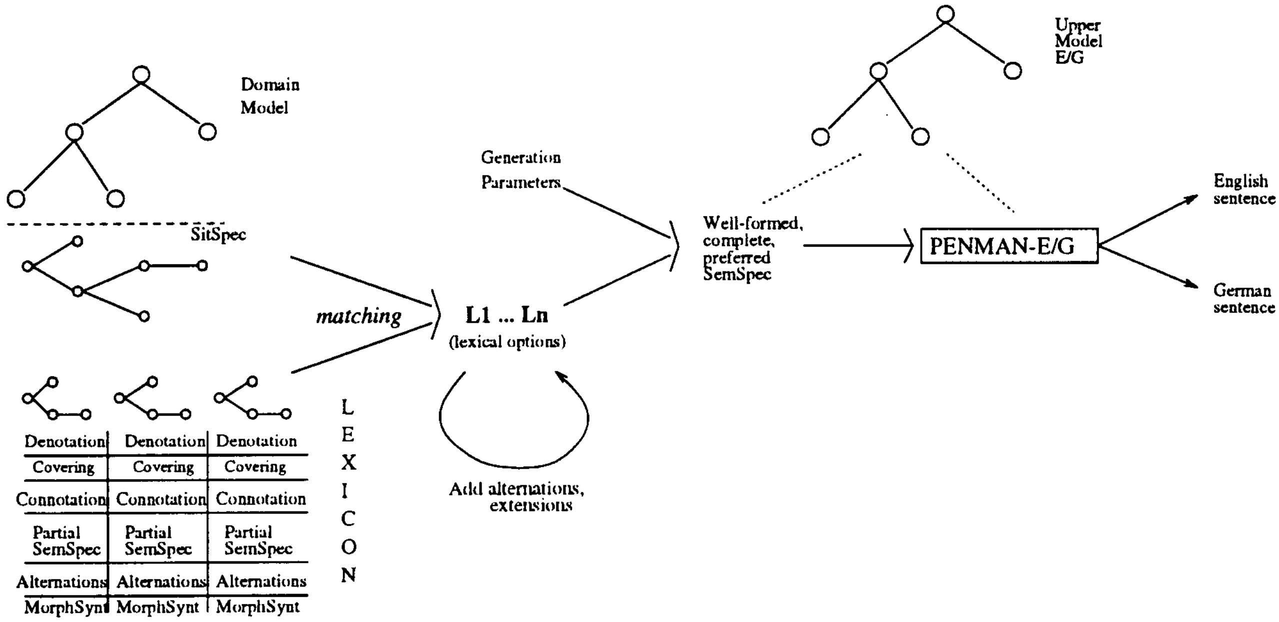

Figure 1: MOOSE system architecture

SitSpec A SitSpec is meant to be neutral between the target languages and between particular paraphrases. It is organized along a variant of the ontological categories proposed by Vendler [1967] and developed further, *inter alia*, by Bach [1986]. We have extended Bach's ontology by breaking up *events* so that their internal structure is explicitly represented (similar to Pustejovsky's [1991] proposal): An event is composed of a *pre-state* (holding before the event commences), a *post-state* (holding when the event is over), and an optional *activity* that brings the transition about. An event without such an activity is a mere state *transition*, e.g., *The room lit up*. An event including an activity is a *culmination*; as an example, consider the event of oil draining from an engine, which is given here in an abbreviated KL-ONE notation (roles names in capital letters, instance names in lower-case):

```
(event1 (PRE-STATE
          (fill-state1 (VALUE 'not-empty)
                       (CONTAINER engine1)
                       (CONTENT oil1))
        (ACTIVITY
          (move-1 (OBJECT oil1)
                  (PATH (path1 (DESTINATION tank1)))))
        (POST-STATE
          (fill-state2 (VALUE 'empty)
                       (CONTAINER engine1))))
```

Figure 2 shows the overall taxonomy of situation types. Subsumed by the general ontological system, a domain model is defined that holds the concepts relevant for representing situations in a technical sample domain and that specifies the exact conditions for the

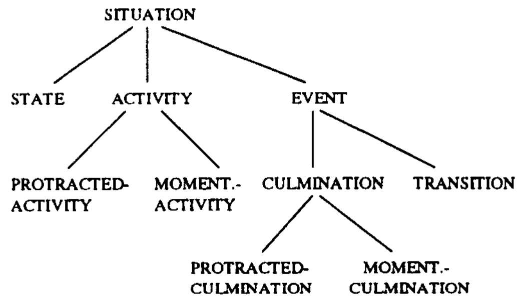

Figure 2: Situation types in the ontology of MOOSE

well-formedness of situations. It is implemented in the KL-ONE language LOOM [MacGregor, Bates 1987].

SemSpec The level of SemSpecs is motivated by the notion of "upper modelling" [Bateman et al. 1990] and is a subset of the input representation language that was developed for Penman, the sentence plan language (SPL) [Kasper 1989]. As opposed to a general SPL term, a SemSpec must contain only upper model concepts and no domain concepts—recall that the domain model in MOOSE is not subsumed by the upper model. Furthermore, since our system takes lexicalization as the decisive task in mapping a SitSpec to a SemSpec, the UM concepts referred to in a SemSpec must be annotated with `:lex` expressions; thus, a SemSpec is a lexicalized structure. Accordingly, we see the upper model as a taxonomy of lexical classes.

SemSpecs are constructed from SitSpecs by selecting a UM-process and mapping SitSpec elements to participant roles of that process, so that all elements of the SitSpec are covered. This choice of process and participants in effect establishes a perspective on the situation; SitSpec is underspecified in this respect. SemSpec is still underspecified with regard to, for example, constituent order and lexical choice between near-synonyms (that have the same semantics with respect to SitSpec yet differ in terms of style, collocational restrictions, etc.). These and other decisions are made, on the basis of verbalization parameters, by the surface generators.

2.3 The role of the lexicon

MOOSE is designed with the goal of strong lexical paraphrasing capabilities in mind. Therefore, its lexicon is rich in information so that lexical choices can be made on the basis of various generation parameters (which are not discussed in this paper). A lexical entry in MOOSE has the following components:

Denotation A partial SitSpec that defines the *applicability condition* of the lexeme: if its denotation subsumes some part of the input SitSpec, then (and only then) it is a candidate lexical option for the verbalization.

Covering The subset of the denotation nodes that are actually expressed by the lexeme. One of the constraints for sentence production is that every node be covered by some lexeme.

Partial SemSpec (PSemSpec) The contribution that the lexeme can make to a sentence SemSpec. By means of shared variables, the partial SemSpec is linked to the denotation.

Connotations Stylistic features pertaining to formality, floridity, etc. See [DiMarco et al. 1993].

Salience assignment (for verbs only): A specification of the different degrees of prominence that the verb assigns to the participants.

Alternation rules (for verbs only): Pointers to lexical rules that represent alternations the verb can undergo (see section 4).

Morphosyntactic features Standard features needed by the surface generator to produce correct utterances.

The goal of the SitSpec—SemSpec mapping is the production of a compl[illegible] the partial SemSpecs (PSemSpecs) associated with a subset of the lexical options, such that the lexemes in this subset collectively cover the entire SitSpec. This unification process is driven by the candidate verbs; their PSemSpec consists of an upper model process and the mappings from situation elements to process participants, which is achieved by co-indexing with positions in the denotation. By means of sharing this information between denotation and PSemSpec, the lexicon entries serve as a "bridge" between the SitSpec to be verbalized and the intermediate representation SemSpec; thus, the role of the lexicon in MOOSE is somewhat similar to that in DIOGENES [Nirenburg and Nirenburg 1988].

Importantly, the denotation of a lexeme need not be a single concept; instead, it can be a complete configuration of concepts and roles (cf. Horacek [1990]). This is necessary since we want to break up the internal event structure in the representation of verb meaning. Consequences are higher computational cost in finding lexical options, but also a higher flexibility in finding different verbalizations of the same event. As an example, consider the denotation of the causative reading of *to fill*:

```
(event (PRE-STATE (fill-state (VALUE (not 'full))
                              (CONTAINER A)))
       (ACTIVITY (CAUSER B))
       (POST-STATE (fill-state (VALUE < D 'full >)
                               (CONTAINER A)
                               (CONTENT C))))
```

The variables are bound to instances or atomic values of the SitSpec when the two are matched against each other. The filler of the VALUE role in the POST-STATE appears in angle brackets because it is a default value, which we do not discuss further here, though. The accompanying partial SemSpec of *to fill* contains the same variables:

```
(x / directed-action  :lex fill
   :actor B  :actee A  :inclusive C  <:destination D>)
```

When the denotation is matched against a SitSpec, the variable bindings are propagated to the partial SemSpec; and when it is unified with the partial SemSpecs corresponding to the other elements, a complete SemSpec results, from which PENMAN produces a sentence like *Jill filled the tank with oil.* (If the VALUE is different from 'full, it also gets verbalized, such as in *Jill filled the tank to the second mark.*)

3 Verb semantics

3.1 Aktionsart

Since verb denotations are complex enough to reflect certain parts of event structure, they can be related to the notion of *Aktionsart*: the verb-inherent features characterizing (primarily) the temporal distribution of the event denoted. The variety of phenomena in Aktionsart are far from clear-cut, and there is no generally accepted and well-defined set of features. In the following, we use the terms given by Bussmann [1983] and discuss only those Aktionsart features that are directly relevant for us because they relate types of SITUATIONS to denotations of verbs. Thus, within the context of our system, we define Aktionsart features in terms of patterns of verb denotations. The following table lists the correspondences.

Aktionsart	Denotation pattern
stative	`(state X)`
durative	`(protracted-activity X)`
semelfactive	`(momentaneous-activity X)`
transformative	`(event (PRE-STATE X) (POST-STATE not-X))`
resultative	`(event (ACTIVITY X) (POST-STATE Y)`
causative	`(activity (CAUSER X))`

Simple cases are *stative* verbs like *to own* or *to know*. *Durative* verbs characterize continuous occurrences that do not have internal structure, like *to sleep, to sit*. In the class of non-durative verbs we find, amongst others, the opposition between *iterative* and *semelfactive* ones. The former are durative activities that result from repeating the same occurrence. In contrast, a semelfactive verb denotes a single occurrence, thus in our system a MOMENTANEOUS-ACTIVITY, as for example *to knock*. *Transformative* verbs involve a change of some state, without a clearly recognizable event that would be responsible for it: *The room lit up*. The denotation of such verbs thus involves a pre-state and a post-state, which is the negation of the former. In our ontology, these are TRANSITIONS. *Resultative* verbs, on the other hand, characterize situations in which something is going on and then comes to an end, thereby resulting in some new state (CULMINATIONS in our ontology). Their denotation includes an activity and a post-state. In the literature, such verbs are often also called *inchoative*.[1] The final verb-inherent feature we use is the well-known *causative*, which reflects the presence of a CAUSER in the denotation.

3.2 Valency

Valency, as introduced by Tesnière [1959], refers to the distinction between *actants* and *circumstantials* (central participants associated with the verb versus temporal, locational, and other circumstances). This separation is in principle widely accepted, but views differ on where to draw the line and how to motivate it. The notion of valency was further developed predominantly in German linguistics, with a culmination point being the valency dictionary of German verbs by Helbig and Schenkel [1973]. They made an additional distinction between 'obligatory' and 'optional' actants; Somers [1987, ch. 1] proceeded to propose six different levels of valency binding. He also pointed out that there are different opinions on the *type* of entities that are subject to a verb's valency requirements: different authors describe them by syntactic class, some by semantic deep cases, and some by their function (subject, object, etc.).

In our approach, which is driven by the (practical) needs of MLG, we aim at encapsulating syntactic matters in the front-end generators and here look at valency in the SitSpec–SemSpec mapping: When characterizing the linking between SitSpec elements and SemSpec participants/circumstances, we describe valency in terms of upper model concepts.

We wish to distinguish cases like the following:

- *Tom disconnected the wire {from the plug}*. *To disconnect* requires a SOURCE, but it can be omitted in a suitable specific context.
- *Sally ate*. While *to eat* usually requires a direct object, it can also be used intransitively due to the strong semantic expectation it creates on the nature of the object—*independent* of the context.
- *Tom put the book on the table*. *To put* requires a DESTINATION, and it cannot be omitted, no matter how specific the context.
- *The water drained from the tank {in the garage}*. Locative circumstances like *in the garage* are not restricted to particular verbs and can occur in addition to PATHS required by the verb.

Adopting the three categories proposed by Helbig and Schenkel [1973], we distinguish between obligatory and optional participants on the one hand, and circumstances on the other. The criterion of optionality, as indicated above, singles out the obligatory complements. But how, exactly, can we motivate the distinction between optional participants and circumstances in our framework? By relating the PSemSpec to the SitSpec, via the denotation. In the *disconnect* case, for instance, the two items CONNECTOR and CONNECTEE are both integral elements of the situation. The situation would not be well-formed with either of them absent, and the domain model encodes this restriction. Therefore, both elements also occur in the denotation of *to disconnect*, and a co-indexed variable provides the link to the PSemSpec. Only when building the sentence SemSpec is it relevant to know that the CONNECTEE can be omitted. The CONNECTEE in the denotation therefore must have its counterpart in the PSemSpec—that is the SOURCE, but there it is marked as optional (see figure 6 below).

[1] The term 'inchoative' is used to cover a rather broad range of phenomena, including the beginning of an event (e.g., *to inflame*) or its coming to an end. We think the term is overloaded and prefer to use 'resultative' for the latter group.

With circumstances, the situation is different: A SitSpec is complete and well-formed without the information on, for instance, the location of an event. Hence, a verb's denotation cannot contain that information, and it follows that it is not present in the PSemSpec, either.

3.3 Verbs and the upper model

Now, since our instrument for ensuring the well-formedness of PSemSpecs and SemSpecs is the upper model, we need to inspect the role of valency information in the UM. On the one hand, Bateman et al. [1990] are well aware of the problems with ascribing simple valency patterns to verbs, but for the practical implementation of Penman and the UM, some strict—and simplifying—category distinctions had to be made. Thus, all participants of process types, as listed above, are coded in LOOM as obligatory roles. Circumstances, on the other hand, are in the UM coded as LOOM relations, and there are no restrictions as to what circumstances can occur with what processes. Spatio-temporal information is generally seen as a circumstance. Concerning the linguistic realizations, Penman and the UM in their present form essentially go back to the Tesnèrian suggestion that participants are realized as nominal groups (with some obvious exceptions, as in *say that x*), and circumstances as prepositional phrases or as adverbs.

But neither this syntactic division corresponding to participants and circumstances (direct or indirect object versus adverbs or prepositional phrases) nor the UM's semantic postulate that spatio-temporal aspects are circumstances hold *in general*. Regarding spatial relationships, we find verbs that specifically require PATH-expressions, which cannot be treated on a par with circumstances: Recall *to put*, which requires a direct object and a DESTINATION. Causative *to pour* requires a direct object as well as a PATH with either a SOURCE, or a DESTINATION, or both: *pour the water from the can into the bucket*. Some verbs, as is well-known, can occur with either a PATH (*Tom walked into the garden*) or with a PLACE (*Tom walked in the garden*), and only *in the garden* can here be treated as a circumstance. And *to disconnect* requires a direct object (the entity that is disconnected) and a SOURCE (the entity that something is disconnected from), which can be omitted if it is obvious from the context: *Disconnect the wire!*

The upper model in its present form cannot make distinctions of this kind. It is not possible to specify a PATH expression, which will be realized as a prepositional phrase, as an obligatory participant. About *to disconnect* (in the causative reading), which is a MATERIAL-PROCESS, the UM can only state that the roles ACTOR and ACTEE must be filled, but not the fact that there is another entity involved—in the domain model called the CONNECTEE—which is verbalized as a SOURCE. Moreover, the UM does not know that the CONNECTEE is optional in the verbalization; it does not distinguish between obligatory and optional participants.

As a step forward to a more fine-grained distinction between participants and circumstances, we differentiate between requirements of process types (as coded in the UM) and requirements of individual verbs, which are to be coded in the lexical entries. In a nutshell, valency (as a lexical property) needs to supplement the participant/circumstance requirements that can be stated for types of processes. To encode the valency information, we use the partial SemSpec of a lexicon entry. The participant roles stated there are either obligatory or optional, in which case they are marked with angle brackets:

```
to disconnect
PSS: (x / directed-action
        :actor A  :actee B  < :source C >)
```

With obligatory participants, the verb is only applicable if the elements denoted by these participants are present in the SitSpec. Optional participants need not necessarily be included in the verbalization: If they are present in the SitSpec, they may be omitted if there is some good reason (e.g., a stylistic preference); if they are not present in the SitSpec, the verb can be used anyway.

4 Alternations and extensions

Having explained denotations and PSemSpecs, specifically for verbs, we can now deal with the task of accounting for the different *alternations* a verb can undergo. A generator needs to know that a verb like *to fill* can occur in a variety of configurations: *Water filled the tank, The tank filled with water, Tom filled the tank with water*. The most comprehensive source of information on verb alternations is the compilation by Levin [1993]; we inspect some of the more prominent alternations listed there and characterize them in terms of changes in denotation and valency of the verbs.

4.1 Alternations as meaning extensions

A simple way of treating alternations is using a separate lexical entry for every configuration, but that would clearly miss the linguistic generalizations. Instead, we wish to represent the common "kernel" of the different configurations only once, and use a set of lexical rules to derive the alternation possibilities. Jackendoff [1990] is concerned with this problem for a number of alternations; specifically, in his LCS framework he seeks to explain the relationships between stative, inchoative, and causative readings of a verb (such as those of *to fill* given above). In Jackendoff's analysis, the forms are derived sequentially by embedding in the primitives INCH and CAUSE, respectively:

- stative: $\text{BE}([_{Thing}\ ]_{\langle A\rangle}, [\text{IN}_d\ [_{Thing}\ ]_A\ ])$
- inchoative: $\text{INCH}\ [\text{BE}([_{Thing}\ ]_{\langle A\rangle}, [\text{IN}_d\ [_{Thing}\ ]_A\ ])]$
- causative: $\text{CAUSE}([_{Thing}\ ]_A, \text{INCH}\ [\text{BE}([_{Thing}\ ]_{\langle A\rangle}, [\text{IN}_d\ [_{Thing}\ ]_A\ ])]$

For our NLG purposes, the idea of deriving complex verb configurations from more basic ones is attractive, but it is necessary that we relate verb meaning to our explicit treatment of event structure, instead of masking that structure with a primitive like INCH. The idea is to see verb alternations not just as relations between different verb forms, but to add directionality to the concept of alternation and treat them as functions that map one into another. From this viewpoint, there are two groups of alternations: (1) Alternations that do not affect the denotation of the verb. Examples are the passive or the substance–source alternation (*The tank leaked oil; Oil leaked from the tank*): The truth conditions do not change. (2) Alternations that do change the denotation of the verb.

The critical group is (2), because if we derive verb configurations from others and rewrite the denotation in this process, it has to be ensured that the process is monotonic. Therefore we define the directionality for group (2) to the effect that an alternation always *adds* meaning: the newly derived form communicates more than the old form—the denotation gets extended. This notion is different from the standard, non-directional way in which alternations are seen in linguistics; to label the difference, we call alternations of group (2) *extensions*. In this section, we will introduce a number of extension rules for which we can give a clear definition in terms of Aktionsart features, as they were introduced in section 3.1. These rules extend the denotation of a verb and rewrite its PSemSpec in parallel to reflect the change in valency; the result is a new verbalization option, which can differ from the previous one in terms of coverage or attribution of salience (not discussed here). The rules will be conveniently simple to state, thanks to the upper model, which provides the right level of abstraction from syntax.

We illustrate our goal with an example. If a SitSpec encodes the situation of Tom removing all the water from a tank, then the verb *to drain* is a candidate lexeme. While it can appear in a number of different configurations, we wish to match only one of its forms against the SitSpec, though. This is the most basic one, denoting an ACTIVITY: *The water drained from the tank*. Here, the case frame of the verb has to encode that *from the tank* is an optional constituent. Now, an extension rule has to systematically derive the CAUSATIVE form: *Tom drained the water from the tank*. And also from the first configuration, another rule derives the RESULTATIVE reading, which adds the information that the tank ended up empty: *The tank drained of the water*. Here, *of the water* is an optional constituent. To this last form, a causative extension can apply and yield *Tom drained the tank of the water*.

To compute these configurations automatically, we define an alternation or extension rule as a 5-tuple with the following components:

`NAM`: a unique name;
`DXT`: extension of denotation;
`COV`: additions to the covering-list;
`ROC`: role changes in PSemSpec;
`NRO`: additional PSemSpec roles and fillers.

The `DXT` contains the denotation subgraph that the new verbalization has in addition to the old one. The syntax is, of course, the same as that of the denotation of a lexical entry. Specifically, it can contain variables; these can co-occur in the `COV` list: the items that the new verbalization covers in addition to those of the old one. `ROC` is a list of pairs that exchange participant role names or the UM-type in the PSemSpec; this replacement can also change optionality. For example, `(< :actee > :actor)` means "replace the term `:actee` in the PSemSpec of the old verbalization, where it was optional, with `:actor`, which is not optional." Finally, `NRO` contains new roles and fillers that are to be added to the new PSemSpec; these will also contain variables from the denotation extension.

Applying such a rule to a verbalization option *vo* works as follows: Add the contents of `DXT` to the denotation of *vo*, and match the new part against the SitSpec. If it matches, make a copy *vo′* of *vo* and assign it a new name as well as the denotation just formed. Add the `COV` list, which has been instantiated by the matching, to the covering-list of *vo′*. Exchange the role names in the PSemSpec of *vo′* as prescribed by `ROC`, and, importantly, in the order they appear there. Finally, add `NRO` to the PSemSpec.

4.2 Lexical rules for extensions

Stative–resultative Example: *Water filled the tank / The tank filled with water*. In discussing verbs that denote a STATE, Jackendoff [1990] points out that *fill*, *cover*, *surround*, and *saturate* can describe either a

STATE or an inchoative event, and encodes the difference with the primitive INCH we have shown in the introduction to this section. Our goal is to do without the primitive, and to define the change in terms of the Aktionsart of the verb; to this end, we use RESULTATIVE in the place of 'inchoative' (see section 3.1).

On a similar matter, Levin [1993] describes the 'locatum subject' alternation, which for instance holds between *I filled the pail with water* and *Water filled the pail.* It thus relates a causative and a non-causative form. Levin states that the alternation applies to a class of 'fill verbs', which are many more than the four given by Jackendoff, and her alternation is not exactly the one we need here, since it also involves a causative form—deriving this, however, is in our framework a separate step.

What we need here is a mixture of Jackendoff's and Levin's insights: Several of Levin's fill verbs can be both transitive and intransitive; and some of the intransitive readings denote 'to become Xed'. Among these verbs are *fill, flood, soak, encrust,* or *saturate*: *The kitchen flooded with water* means the same as *The kitchen became flooded with water.* For this subgroup of the fill verbs we define an extension rule that derives from a STATE reading a RESULTATIVE one. Notice that this is different from Levin's 'locatum subject' alternation, since it does not involve a causer.

```
NAM: stative-resultative
DXT: (event (Y (ACTIVITY X)))
COV: (X Y)
ROC: ((:actor :inclusive)
      (:actee :actor)
      (directed-action nondirected-action))
NRO: ()
```

To illustrate the rule with an example, consider the denotation and PSemSpec of the STATE reading of *fill*:

```
DEN: (fill-state (CONTAINER A)
                 (CONTENT B)
                 (VALUE C))

PSS: (x / directed-action :lex fill_el
      :actor B  :actee A  < :destination C >)
```

When matching it against a SitSpec with a tank and water, this yields the verbalization *The water filled the tank*, covering only the POST-STATE of the SitSpec. Now, the alternation rule extends the denotation to also covering the EVENT and the ACTIVITY that brings the filling about. Applying the changes to the PSemSpec results in

```
(x / nondirected-action :lex fill_el
  :inclusive B :actor A < :destination C >)
```

from which PENMAN produces *The tank was filled with the water.*

A few stative verbs cannot be RESULTATIVE without being also CAUSATIVE. Consider *to cover* in these examples from Jackendoff:
Snow covered the ground.
**The ground covered with snow.*
Bill covered the ground with snow.
For these, a 'stative–culmination' extension derives the RESULTATIVE+CAUSATIVE form directly from the STATIVE one. The rule is defined similar to the one given above, so we do not show it here.

Causative extensions Example: *The napkin soaked / Tom soaked the napkin.* Levin discusses a 'causative/inchoative' alternation that applies to a large number of verbs. The class formed by them is somewhat heterogeneous with respect to the Aktionsart, though; it contains for example *to move* as well as *to open.* The former is in its basic form DURATIVE (*The cat moved*), and the latter TRANSFORMATIVE (*The door opened*). Accordingly, we split the alternation in two, which only differ in the DXT component, reflecting the difference in Aktionsart. The alternation adds a CAUSER to the denotation, makes the former :actor the new :actee, and accordingly changes the overall UM-type from NONDIRECTED-ACTION to DIRECTED-ACTION, because there is now an ACTEE present.

```
NAM: durative-causative
DXT: (activity (CAUSER X))
COV: ()
ROC: ((:actor :actee)
      (nondirected-action directed-action))
NRO: (:actor X)

NAM: resultative-causative
DXT: (event (ACTIVITY (X (CAUSER Y))))
COV: ()
ROC: ((:actor :actee)
      (nondirected-action directed-action))
NRO: (:actor Y)
```

The first rule derives, for example, *Tom moved the cat* from *The cat moved*, and the second *Tom closed the door* from *The door closed.*

Locative extensions Example: (a) *Sally sprayed the wall with paint./* (b) *Sally sprayed paint onto the wall.* We have mentioned the locative extension above; its characteristic is that configuration (a) of the verb conveys that something is performed in a 'complete' or 'holistic' manner, whereas configuration (b) lacks this facet of meaning. Levin points out that this alternation has received much attention in linguistics research and notes that, in spite of the efforts, a satisfactory definition of the 'holistic'-facet has not been found. Jackendoff, in his treatment of the alternation, suggests encoding the 'holistic' feature in a primitive: the

Sally sprayed paint onto the wall.

```
(spray-1 (CAUSER sally-1)
      (OBJECT paint-1)
      (PATH (path-1 (DESTINATION wall-1))))
```

Sally sprayed the wall with paint.

```
(event-1 (PRE-STATE (covered-state-1 (OBJECT wall-1)
                                     (VALUE (not 'covered))))
      (ACTIVITY (spray-1 (CAUSER sally-1)
                         (OBJECT paint-1)
                         (PATH (path-1 (DESTINATION wall-1)))))
      (POST-STATE (covered-state-1 (OBJECT wall-1)
                                   (VALUE 'covered))))
```

Figure 3: SitSpecs for configurations of *to spray*

function ON_d is a derivative of ON and means that something 'distributively' covers a surface, e.g., the paint covers all of the wall. Introducing a primitive, though, amounts to conceding that no explanation in terms that are already known can be given. We cannot solve the question of 'holisticness', either, but we want to point to the fact that the two verb configurations correlate with a change in Aktionsart: *Sally sprayed paint onto the wall* is durative (she can do it *for two hours*), whereas *Sally sprayed the wall with paint* is transformative (she can do it *in two hours*). That observation leads us to propose that the example is best analyzed as involving a mere ACTIVITY in the *with* configuration, and an additional TRANSITION in the *onto* configuration. Hence, we assign two different SitSpecs for the sentences, one ACTIVITY and one EVENT, as shown in figure 3.

The crucial point now is that the first SitSpec is fully embedded in the second; this is in correspondence with the truth conditions: If Sally has sprayed the wall with paint, then she also has sprayed paint onto the wall. To generalize the correspondence to an extension rule, we need to assume in the domain model a concept like COMPLETION-STATE, which is to subsume all those STATES in the domain model that have "extreme" values: an empty bucket, a fully loaded truck, and so forth. The exact interpretation of COMPLETION-STATE is the open question that Levin [1993] referred to, and that Jackendoff treated with his '$_d$' subscript. We do think, though, that an abstract STATE in the domain model, which subsumes a range of the concrete STATES, is preferrable to introducing a primitive on the linguistic level (unless the primitive is relevant for other linguistic phenomena as well).

The following alternation rule applies to durative verb readings that denote ACTIVITIES of something being moved to somewhere, and extends them to also cover the POST-STATE, which must be subsumed by COMPLETION-STATE. In this way, it derives reading (a) from (b) in the *spray* example, and analogously for the other verbs undergoing the alternation, e.g.: *Tom loaded hay onto the wagon / Tom loaded the wagon with hay*; *Jill stuffed the feathers into the cushion / Jill stuffed the cushion with the feathers.* The PSemSpec is modified as follows: The former `:destination` (*wall*) becomes the new `:actee`, whereas the former `:actee` (*paint*) now fills the role `< :inclusive >`, and is optional there, because *Jill stuffed the cushion* is also well-formed.

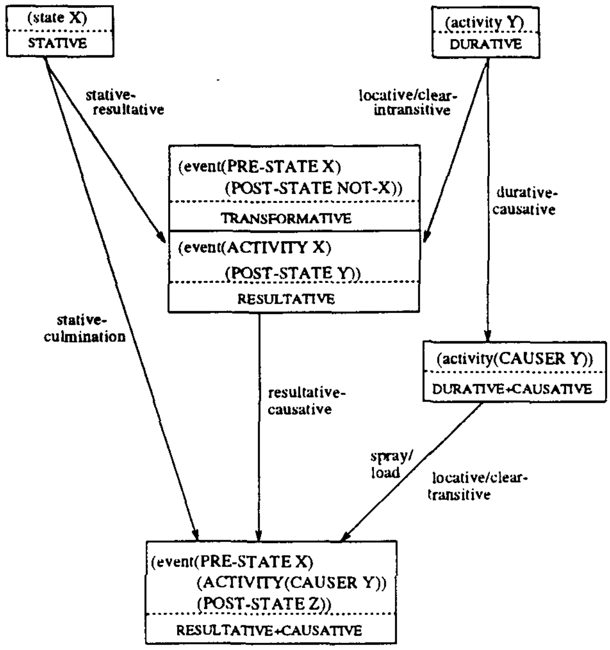

Figure 4: Dependency of extension rules

```
NAM: locative-transitive
DXT: (event
       (MOVE (OBJECT X)
             (PATH (DESTINATION Y)))
       (POST-STATE (Z completion-state (OBJECT Y))))
COV: (Z)
ROC: ((:actee < :inclusive >)
      (:destination :actee))
NRO: ()
```

Most of this rule covers two kinds of locative alternation, which Levin distinguishes: the 'spray/load' alternation and the 'clear (transitive)' alternation. The latter applies only to the verbs *clear, clean, drain, empty* and can be seen as the 'semantic inverse' of the spray/load alternation, because one group of verbs denotes activities of placing something somewhere, and the other describes activities of removing something from somewhere; but both have the same 'holistic' effect in one of the verb configurations. For example, the rule derives *Tom drained the container of the water* from *Tom drained the water from the container.*[2] Thus, the rule for the clear-

[2] We ignore the role of the definite determiner here, which in

```
Denotation: (activity (OBJECT A)
                      (PATH (SOURCE B)))
PSemSpec: (x1 / nondirected-action  :lex drain_el
          :actor A :source B)
```

(0) *The water drained from the tank.*

Locative/clear-intransitive of (0):

```
Denotation: (event (ACTIVITY (OBJECT A)
                             (PATH (SOURCE B)))
                   (POST-STATE (C (OBJECT B))))
PSemSpec: (x1 / nondirected-action  :lex drain_el
          :of-matter A :actor B)
```

(1) *The tank drained of the water.*

Durative–causative of (0):

```
Denotation: (activity (OBJECT A)
                      (PATH (SOURCE B))
                      (CAUSER C))
PSemSpec: (x1 / directed-action  :lex drain_el
          :actee A :source B :actor C)
```

(2) *Tom drained the water from the tank.*

Resultative–causative of (1):

```
Denotation: (event (ACTIVITY (OBJECT A)
                             (PATH (SOURCE B))
                             (CAUSER C))
                   (POST-STATE (C (OBJECT B))))
PSemSpec: (x1 / directed-action  :lex drain_el
          :of-matter A :actee B :actor C)
```

(3) *Tom drained the tank of the water.*

Figure 5: Derivation of *drain*-configurations

alternation is the same as the one shown above, with three exceptions: the keyword replacing `:actee` is not `<:inclusive>` but `<:of-matter>`, the `DESTINATION` in the denotation is a `SOURCE`, and correspondingly, the keyword `:destination` is `:source`.

The *clear* verbs, except for *to clean*, can in addition be intransitive, and Levin states a separate alternation for them. For *to drain*, the first configuration is *The water drained from the tank*, and the second is either *The tank drained* or ?*The tank drained of the water*. According to Levin, "the intransitive form may be best in the absence of the *of*-phrase" [Levin 1993, p. 55]. The SitSpec denoted by the first configuration is:

The water drained from the tank.

```
(move-1 (OBJECT water-1)
        (PATH (path-1 (SOURCE tank-1))))
```

Note that our durative–causative extension rule given above applies in this case and extends the coverage of the SitSpec to one corresponding to *Tom drained the water from the tank*. A rule that is parallel to that for the transitive case is given below; it derives ?*The tank drained of the water*; since the `< :of-matter >` is optional, we can also produce *The tank drained*, which is, according to Levin, preferred.

```
NAM: locative/clear-intransitive
DXT: (event
       (MOVE (OBJECT X)
             (PATH (SOURCE Y)))
       (POST-STATE (Z completion-state (OBJECT Y))))
COV: (Z)
ROC: ((:actor < :of-matter >)
      (:source :actor))
NRO: ()
```

Summary The extensions introduced now can apply in a sequential order to a verb. Figure 4 provides a synopsis: The boxes contain the denotation patterns that correspond to the Aktionsart feature, and the rules transform a configuration with one Aktionsart into another. In this graph, every verb base form has an entry point corresponding to the Aktionsart of its most basic configuration. Examples: *to fill* is STATIVE, *to drain* is DURATIVE, *to open* is TRANSFORMATIVE, *to remove* is RESULTATIVE+CAUSATIVE. The "double box" in the middle is the entry point for both TRANSFORMATIVE and RESULTATIVE verbs, but the incoming arrows produce RESULTATIVE forms.

From the entry point of a verb, arcs can be followed and rules applied if the respective alternation is specified in the lexical entry. Returning to the example of *to drain*, figure 5 shows how the rules successively derive the various configurations.

4.3 Examples: lexical entries for verbs

To illustrate our treatment of valency, argument linking, and alternation/extension rules, figure 6 shows excerpts from lexical entries of eight different verbs. The information is arranged as follows: On the right-hand side is the case frame of the verb, written as the SemSpec participant keywords (each starting with a colon). Optional participants are enclosed in angle brackets. On the left-hand side are excerpts from the denotation: the names of the roles whose fillers are co-indexed with the respective position in the case frame. Thus, the arrows give the argument linking for the base form of the verb, which can be quite simple, as in *open* or *move*. From the perspective of the domain model, the roles on the left-hand side of the arrows are required to be filled—as is encoded in the LOOM definitions of the underlying concept. Only items appearing with an asterisk in front of them are optional in the SitSpec: for example, a SitSpec underlying an

fact has critical influence on the 'holistic' interpretation of mass nouns.

DISCONNECT CAUSER → :actor CONNECTOR → :actee CONNECTEE → <:source> ―――――	**OPEN** OBJECT → :actor *CAUSER ――――― resultative-causative
POUR PATH-SOURCE → :actor OBJECT → <:actee> *PATH-DESTINATION *CAUSER ――――― substance-source durative-causative	**SPRAY** CAUSER → :actor OBJECT → :actee PATH-DESTINATION → :destination ――――― spray-load
DRAIN OBJECT → :actor PATH-SOURCE → <:source> *PATH-DESTINATION *CAUSER ――――― durative-causative locative/clear-intransitive resultative-causative	**FILL** CONTENT → :actor CONTAINER → :actee VALUE → <:destination (default)> *CAUSER ――――― stative-resultative resultative-causative
MOVE/WALK OBJECT → :actor *PATH *CAUSER ――――― durative-causative	**LEAK** PATH-SOURCE → :actor OBJECT → <:actee> *PATH-DESTINATION ――――― substance-source

Figure 6: Excerpts of sample lexical entries for verbs

OPEN event is well-formed without a CAUSER being present. The optional elements are listed here because they can be verbalized with the extension rules that we have introduced. The names of all the applicable rules (those that we have discussed here) for a verb appear below the line.

5 Summary

We have (very briefly) introduced the sentence generator MOOSE and then inspected the role of lexical semantics therein. MOOSE produces a range of different paraphrases for the same underlying SitSpec, and one instrument to that end is the generation of several verb configurations. We proposed a set of alternation/extension rules that derive such configurations from the basic configuration—which is the only one stored in the lexicon. Alternations are derived by rewriting the partial SemSpec, and in the case of extensions, adding a new subgraph to the denotation, and possibly adding nodes to the covering-list. We have shown that the entire alternation space for a verb like *to drain* can be generated in this manner.

References

[Bach 1986] E. Bach. "The algebra of events." In: *Linguistics and Philosophy* 9, 1986

[Bateman et al. 1990] J. Bateman, R. Kasper, J. Moore, R. Whitney. "A general organization of knowledge for natural language processing: the Penman upper model." USC Information Sciences Institute Tech. Rep., 1990.

[Bussmann 1983] H. Bussmann. *Lexikon der Sprachwissenschaft.* Stuttgart, Kröner, 1983.

[DiMarco et al. 1993] C. DiMarco, G. Hirst, M. Stede. "The Semantic and Stylistic Differentiation of Synonyms and Near-Synonyms." In: Working notes of the AAAI Spring Symposium on Building Lexicons for Machine Translation, Stanford University, March 1993.

[Helbig and Schenkel 1973] G. Helbig, W. Schenkel. *Wörterbuch zur Valenz und Distribution deutscher Verben.* VEB Verlag Enzyklopädie, Leipzig, 1973.

[Horacek 1990] H. Horacek. "The architecture of a generation component in a complete natural language dialogue system." In: R. Dale, C. Mellish, M. Zock (eds.): *Current research in natural language generation.* London: Academic Press, 1990.

[Jackendoff 1990] R. Jackendoff. *Semantic structures.* Cambridge/MA: MIT Press, 1990.

[Kasper 1989] R. Kasper. "A flexible interface for linking applications to Penman's sentence generator." In: Proceedings of the *DARPA Workshop on Speech and Natural Language Processing*, University of Pennsylvania, 1989.

[Levin 1993] B. Levin. *English verb classes and alternations.* Chicago: University of Chicago Press, 1993.

[MacGregor, Bates 1987] R. MacGregor, R. Bates. "The Loom Knowledge Representation Language." USC Information Sciences Institute Tech. Rep. ISI/RS-87-188, 1987.

[Nirenburg and Nirenburg 1988] S. Nirenburg, I. Nirenburg. "A framework for lexical selection in natural language generation." In: *Proceedings of the 12th International Conference on Computational Linguistics (COLING).* Budapest, 1988.

[Penman 1989] The Penman group. Unpublished Documentation of the Penman Sentence Generation System. USC Information Sciences Institute, Marina del Rey/CA, 1989.

[Pustejovsky 1991] J. Pustejovsky. "The syntax of event structure." In: *Cognition* 41 (47-81), 1991.

[Rösner and Stede 1994] D. Rösner, M. Stede. "Generating multilingual documents from a knowledge base: The TECHDOC project." In: *Proceedings of the International Conference on Computational Linguistics (COLING).* Kyoto, 1994.

[Somers 1987] H. Somers. *Valency and case in computational linguistics.* Edinburgh: Edinburgh University Press, 1987.

[Stede and Grote 1995] M. Stede, B. Grote. The lexicon: Bridge between language-neutral and language-specific representations. In: Working notes of the IJCAI workshop on multilingual text generation. Montréal, 1995.

[Stede 1996] M. Stede. "Lexical semantics and knowledge representation in multilingual sentence generation." Doctoral dissertation, University of Toronto, 1996.

[Tesnière 1959] L. Tessniére. *Èlèments de syntaxe structurale.* Paris: Klinsieck, 1959.

[Vendler 1967] Z. Vendler. *Linguistics and Philosophy.* Ithaca/NY: Cornell University Press, 1967.