Paper appears in: ECAI'98 Workshop on Multilinguality in the Lexicon, Brighton, U.K., August 1998.

Multilingual Grammars and Multilingual Lexicons for Multilingual Text Generation

John Bateman
Centre for Communication and Language
Dept. of English
University of Stirling
Scotland, U.K.
(e-mail: j.a.bateman@stir.ac.uk**)**

Serge Sharoff
Russian Research Institute for Artificial Intelligence
Moscow, Russia
(e-mail: sharoff@mx.iki.rssi.ru**)**

Abstract. In this paper we discuss some practical steps that support the integration of multilingual grammars designed for text generation and multilingual lexicons that describe the lexical stocks of a variety of natural languages. We suggest that grammatical resources for natural language generation and lexical resources for lexical representations are not necessarily commensurate currently in their design and aims, although both contain information necessary for natural language generation. We therefore discuss methods by which heterogenous generation systems can be constructed in which large-scale lexical and grammatical components can be maintained separately but allowed to interact as is necessary for generation.

1 Introduction

In this paper we consider some of the practical steps that we are taking in order to provide support for the integration of multilingual grammars designed for text generation and multilingual lexicons that describe the lexical stocks of a variety of natural languages. Given the current acceptance of the lexicalization of linguistic resources as an almost unquestioned goal, it may at first glance appear curious to talk of grammars and lexicons as separate entities. But both the need for, and utility of, doing so is made clear when we consider using current lexical resources for practical multilingual natural language generation (MLG). As we will clarify below, lexical resources are, in fact, not typically organized effectively for MLG—nor, perhaps, should they be.

Some of the most effective multilingual generation grammars are organized around the notion of *communicative function*: such grammars are concerned crucially with mapping semantic specifications, including statements of discoursal and interpersonal semantics, to grammatical constructions appropriate for those semantics. This is a goal distinct from that pursued for lexicon construction and, as a natural consequence, the organizational features of generation grammars and general lexicons differ. Nevertheless, both kinds of resources develop and maintain information that is important for multilingual applications, including MLG.

As a consequence, we are currently involved in the development of a class of large-scale generation systems in which the full lexico-morphological-grammatical specifications of languages may be distributed systematically across heterogeneous components. We use systemic-functional linguistics (cf. Halliday 1978) as a metatheory for constraining the use made of this implementational heterogeneity. In particular, two dimensions of SFL theory allow us to compose composite descriptions: the *rank* scale of grammatical categories (consisting typically of 'clauses', 'groups/phrases', 'words' and 'morphemes') and *stratification* for relating different levels of linguistic abstraction.

Experiences with multilingual generation grammars based on the SFL notion of communicative function show that the organization of such grammars generally exhibits greater stability cross-linguistically than more form-based organizations (cf. Bateman 1997). This stability holds for a wider range of languages than those that would be considered as 'related' in the structural typological tradition. Bateman, Matthiessen, Nanri & Zeng (1991b), for example, introduced the idea of multilingual functional grammatical descriptions with a simple speech function classification for English, Japanese and Chinese. The description allows the more formal, structural differences between these languages to be abstracted away from, leaving a generic, mostly shared set of functional alternations including 'imperative'/'indicative', 'question'/'assertion', 'element-question'/'polarity-question', etc. These categories are precisely those which are relevant for constructing a detailed mapping from semantics to form, which is crucial for natural language generation to work. Current generation grammars in this framework are already large-scale resources and so their re-use is a particularly effective strategy for developing broad grammatical coverage that supports generation. Moreover, while the approach appears formally to be similar to the more recently proposed view of multilingual lexicons described by Cahill & Gazdar (1995), it is

ECAI 98. *13th European Conference on Artificial Intelligence*
Workshop on multilinguality in the lexicon.

considerably more general with respect to its view of interlanguage 'relatedness'.

For good functional reasons, however, functional congruence is more likely to be maintained at the higher ranks of grammatical description (e.g., the clause) than at the lower ranks (e.g., words) (cf. Bateman, Matthiessen, Nanri & Zeng 1991a). At lower ranks, form-based considerations come into play more often. Thus, we require both the functionally motivated organizations of clause descriptions and the more form-motivated organizations of word/morpheme descriptions to co-exist, without allowing one to compromise the other. This is provided by the treatment of heterogeneous resources that we describe here.

We organize the paper as follows. First, we briefly summarize the kinds of interaction with lexical information originally foreseen in the language generation architecture on which our MLG components are based. Second, we describe how this was simply extended in the face of multilinguality. Third, we set out how these extensions have now been generalized to allow interaction with lexical resources developed independently of the issues of natural language generation. Each section will take examples from ongoing work on actual resources that illustrate the points made.[1]

2 Lexical information in the Penman-style generation architecture

The multilingual grammatical descriptions introduced by Bateman et al. (1991b) and described in more theoretical detail in Bateman et al. (1991a) with which we are concerned here are being developed using the KPML grammar development environment (cf. Bateman 1997). This system is a further development of the Penman text generation system developed at USC/Information Sciences Institute throughout the 80s (cf. Matthiessen & Bateman 1991). One characteristic of grammars based on this architecture, in addition to their functional orientation for generation, is that they adhere to the deterministic model of generation argued for on practical grounds by, for example, Reiter (1994). Generation of this kind allows neither backtracking nor unification and requires particular solutions for the use of lexical information.

The original implementation of lexical selection for English in the Penman system was as follows. A syntactic structure is constructed on the basis of a semantic specification given as input. The generation procedure employed for this is that of 'traversing' a network of functional alternatives—called a system network—according to the semantic input. Particular options selected during that traversal call for constraints to be set on the subconstituents of the unit being generated. These constraints are expressed by 'realization statements' that are associated with particular options in the grammar network, or grammatical 'features'. The constituents of such a structure may either contain further substructure, in which case the generation procedure recurses and further traversals of the system network of functional alternatives are made, or be considered sufficiently articulated to receive a lexical realization directly. Constituents are identified in this framework by means of functional labels, analogous to the f-structure categories in Lexical-Functional Grammar or traditional grammatical notions such as Subject and Object. An example of the result of one traversal through the grammar is shown in Figure 1. This shows a simple clause structure compatible with sentences such as 'He dislikes apples'. Typically any individual constituent is made up of the 'conflation' of a number of functional constituents: this is shown in the figure by conjoining labels with a slash.

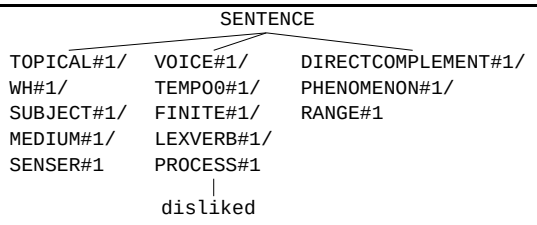

Figure 1. Example of a generated functional structure fragment

When constituents are considered sufficiently specified for lexical realization, the lexical items that are permitted to fill such slots are typically constrained both semantically and grammatically. Semantic constraints come from lexical associations among the concepts of a knowledge base (cf. Albano, Cumming & Sondheimer 1989), while grammatical constraints come from 'lexical classification' constraints imposed on constituents during generation. This default lexical selection algorithm generally assumes that associations between domain concepts and lexical items (i.e., their names) have been established. This selection process then seeks the conceptually nearest lexical item that matches the grammatical lexical classification constraints that have been set during generation.

Lexical classification constraints are expressed in terms of *classify* realization statements. Thus, a realization statement of the form `(classify Process reaction-verb)` constrains the grammatical constituent labelled `Process` to be filled by a lexical item that possesses at least the feature 'reaction-verb'. This might be selected, for example, when the semantic input is expressing that a particular subtype of mental activity or event concerned with reacting to some stimulus is to be generated.[2] Constituents realized in this manner are typically co-constrained by several lexical features selected during generation. For example, the full set of classification constraints placed on the Process constituent of Figure 1 actually consists of: {experience-verb middle-verb reaction-verb dislike-verb}. Furthermore, since each classify constraint is associated with a partic-

[1] The work described has been, and is being, pursued in several projects; these include: KIT-MARKER for German, TECHDOC and VISDOK for German and English, GUME for Spanish, AGILE for Bulgarian, Czech and Russian, DRAFTER-I for French, and KOMET for German, English and Dutch. These all employ functional multilingual generation grammars as originally set out in Bateman et al. (1991b) and used as reference point in this paper.

[2] These semantic types are drawn from a linguistically motivated ontology of abstract semantic categories called the Upper Model originally developed within the Penman project by Mann, Matthiessen and others (cf. Bateman 1990).

ular grammatical feature and the grammatical features of a functional grammatical description serve to capture the rich range of kinds of meaning necessary for effective text generation (i.e., they include and combine semantic, pragmatic, and discoursally-conditioned grammatical phenomena), this provides a natural mechanism for multiply constraining lexical selection forms that are propositionally, interpersonally, and textually appropriate for deployment in texts.

An example of a lexical item compatible with the above classification constraint is the following. The set of features under the `:features` slot contains those features which the grammar may use for classificatory purposes when constraining lexical selection, e.g., 'reaction-verb' and the other classify features given above. Verbs similarly classified include 'detest', 'hate', etc.

```
(LEXICAL-ITEM
  :NAME       DISLIKE
  :SPELLING   "dislike"
  :FEATURES   (DISLIKE-VERB REACTION-VERB
               EXPERIENCE-VERB MIDDLE-VERB
               LEXICAL S-D UNITARYSPELLING
               INFLECTABLE VERB)
  :EDITOR     "Susanna Cumming"
  :DATE "Tuesday the eighth of October, 1985; 2:26:00 pm"
  )
```

Importantly, the lexical item contains no *grammatical* information directly because this is represented properly in the grammar. The constraint that a 'reaction-verb' is required (cf. the classify realization statement above) is only activated when the grammar is constructing a syntactic unit with which this type of lexical item is compatible (as illustrated in Figure 1). Although this is similar to the avoidance of redundancy in lexical entries by placing them in an inheritance hierarchy, it differs from this in that the classification system for grammatical units such as, for example, clauses, and that for words are distinct. The connections between these classifications are managed solely by the classify realization statements present in the grammar; thus the set of classify features given above serves to constrain the lexical items that may appear. This allows communicative function to play a determining role in classification where it does the most work (i.e., at clause rank) without compromising lexically motivated organizations at lower ranks (e.g., words). It also maintains deterministic processing for efficient generation performance.[3]

The information remaining in the lexical entry is then primarily restricted to morphological patterning. The feature 'inflectable' indicates, for example, that this lexical item is subject to morphological variation, while the feature 's-d' specifies that that variation is of a regular kind for English verbs where the third person singular is formed in *-s* and the past tense in *-d*. Such morphological features are not 'classifying': that is, they play no role in constraining the lexical items that may be considered to realize some particular constituent. Their selection is therefore distinguished in the grammar by the use of a different realization statement: *inflectify*; for example: `(inflectify Finite past-form)`. Such a constraint is associated with the grammatical feature that is selected when it has been decided (on the basis of the input semantics) that the clause being generated concerns an event where a past tense is appropriate. The constraint specifies that, whatever lexical item is selected to realize the constituent labelled Finite, that constituent is also constrained to have the morphological property 'past-form'. Thus, if the Finite constituent is 'conflated' with that labelled Process (as is the case in Figure 1), and Process is realized by the lexical item `DISLIKE` above, then these constraints (together with several other inflection constraints imposed during traversal, such as 'thirdperson-form', 'plural-form', etc.) are sufficient for the correct regular morphological form of the verb to be generated.

An example of a lexical item involving irregular morphology is the following.

```
(LEXICAL-ITEM
     :NAME            FIND-MENTAL
     :SPELLING        "find"
     :SAMPLE-SENTENCE "In five minutes I found that the
                       animals all had the same sire."
     :FEATURES        (EDPARTICIPLEFORM PASTFORM S-IRR
                       UNITARYSPELLING MIDDLE-VERB
                       EPISTEMIC-VERB COGNITION-VERB
                       EXPERIENCE-VERB LEXICAL INFLECTABLE
                       VERB)
     :PROPERTIES      ((PASTFORM "found")
                       (EDPARTICIPLEFORM "found"))
     :EDITOR          "Lynn Poulton"
     :DATE "Monday the first of June, 1987; 10:51:25 am"
     )
```

Irregular forms are placed in the `:properties` slot and are identified by a single label, e.g., `PASTFORM`. In this case, the inflection feature 's-irr' of the lexical item indicates that for part of its paradigm (i.e., for the past forms), irregularity is to be expected, while the *features* 'edparticipleform' and 'pastform' indicate which particular forms are irregular. When irregularity is indicated, the generation algorithm consults a mapping from sets of inflection features as used in the grammar (e.g., {past-form thirdperson-form plural-form}) to individual property names that are used in the property slots of lexical items (e.g, `PASTFORM`). This enables the correct morphological form of that lexical item to be produced (e.g., in this case, "found").

In the Penman generation system, all regular forms were generated by special purpose code built into the generator, while irregular forms were specified in the lexical entries as shown here.

3 Extensions for multilingual morphology

This approach to lexical information, and particularly to morphological information, is clearly unsuitable for multilingual descriptions. In the initial design of the KPML multilingual development environment, therefore, several alternatives to this basic scheme were implemented. The first languages for which generation grammars were added into the multilingual resource maintained by KPML were German and Dutch, two languages both more complex morphologically than English.

[3] The equivalence of this representation to a nondeterministic form is discussed by Henschel (1995).

For German an initial interim working solution of a full-form lexicon was adopted. This was modelled simply by using only *classify* realization statements and no *inflectify* realization statements. Individual full-form lexical items were then selected by allowing the classify statements to refer not only to semantically classifying features, but also to individual morphologically conditioned form. While possible, this solution is clearly not to be recommended for large-scale development.

In contrast, for Dutch the *grammar* was extended to include not only classifications of grammatical units such as 'clauses' and 'groups/phrases' but also classifications of 'words' and 'morphemes' (Degand 1996). In this model, words are generated in exactly the same way as are clauses. An example is shown in Figure 2. Here the past participle form 'gestudeerd' ('studied') has been constructed as a grammatical unit consists of three subconstituents, indicating the regular formation of past participles of this class in Dutch. The clause unit above the unit shown specifies one inflectify constraint for this constituent: `(inflectify Tempo1Dependent edparticiple)` and, whereas in the Penman implementation for English, a constraint such as this would have been interpreted by the hardcoded morphology code, here it is interpreted as a constraint on the construction of substructure—as is the case for any subconstituents where substructure is to be generated. Irregular forms are still given in individual lexical items as in the English example above, however, and, when applicable, stop further traversal of the grammar network by supplying the appropriate morphological form directly.

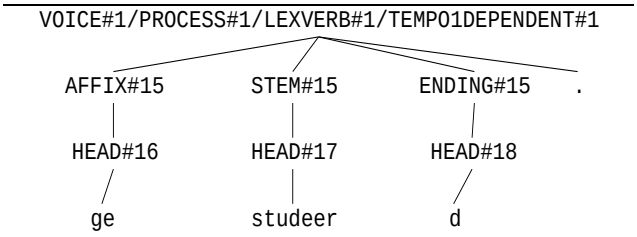

Figure 2. Example of a generated Dutch word structure

The interaction between grammatical structure and lexicon is therefore still based on the original model developed for English, but the hardcoded morphological component of Penman is replaced by an extended grammatical description that can be shared (as partially as required) across languages in exactly the same way as other parts of the multilingual grammatical description supported by KPML can be shared. A similar approach was later taken for French in the DRAFTER-I project (Paris, Linden, Fischer, Hartley, Pemberton, Power & Scott 1995). Partial sharing across languages even at this level of linguistic detail has been shown to be a useful for multilingual morphological specifications by examples such as those given by Cahill & Gazdar (1996).

Work on morphology and lexical resources has generally preceded the development of the functional descriptions most supportive of natural language generation, however, and so adopting this approach for externally developed morphological classifications would require that they first be translated into the format accepted by KPML. This could be a considerable overhead. A useful alternative approach can be seen in a variation of the Penman system developed within the TECHDOC multilingual generation system for instructional texts (Rösner & Stede 1994). There are several extensive computational treatments of German morphology and so, in TECHDOC, the Penman generator was modified[4] so that an external component—the German morphological component MORPHIX (Finkler & Neumann 1988)—intervened between Penman's internal code for producing morphological forms and the construction of the constituent tree containing the final generated strings. When the language being generated was German, the set of 'inflectify'-features selected by the grammar for a constituent (i.e., those features for constraining morphological forms) were mapped not to features corresponding to Penman-style lexical entries, but to features appropriate for MORPHIX. MORPHIX was then given the stem-form of the lexical item selected and the mapped morphological features, and returned the appropriate string for inclusion in the surface structure. The following briefly illustrates this process.

TECHDOC plans instructional texts and produces sequences of semantic specifications that are passed on for tactical generation in English and German. For example, the following semantic input, expressed in the standard Sentence Plan Language (SPL: Kasper 1989) notation for generation in Penman-style architectures:[5]

```
(c / clean
   :actor (hearer / person)
   :actee (di / dirt :determiner any)
   :source (spb / spark-plug :determiner the)
   :speechact imperative))
```

would causes TECHDOC to generate the strings:

English: `Clean any dirt from the sparkplug.`
German: `Jeglichen Schmutz von der Zündkerze entfernen!`

The particular lexical items selected are found by inspecting the semantic types given in the input (`clean`, `dirt`, etc.) which are defined in the domain model. These typically have sets of lexical items associated with them.[6] For example, the semantic type `dirt` might have an association:

`dirt` ≡ {:english `dirt-e`, :german `dirt-g`}

However, the associated lexical items for German now contain no morphological information. They serve solely to provide a link between the grammatical contexts established by the classify constraints given in the grammar and particular sets of *stems* that may occur in those contexts and which can be processed by MORPHIX. Relevant examples are:

[4] Original code by Dietmar Rösner.

[5] The items under `:determiner` are *macros* that simply expand to a collection of semantic constraints: they are *not* syntactic constraints since SPL specifications only contain semantic information.

[6] A more complex (and interesting) lexical selection procedure in this general framework is described for TECHDOC by Stede (1996).

```
(lexical-item
  :name     REMOVE-g
  :spelling "entfern"
  :features (VERB INFLECTABLE UNITARYSPELLING
             LEXICAL EFFECTIVE DOVERB DISPOSAL
             PASSIV))

(lexical-item
  :name     DIRT-g
  :spelling "Schmutz"
  :features (NOUN NOINFLECTIONS COMMON
             NOT-NOMINALIZATION COUNTABLE
             NONSUBSTITUTE))
```

The specification of morphological information is then handled by giving, in addition to these lexical entries, definitions of the particular MORPHIX inflection patterns to which the lexical stems belong:

```
(l-s "entfern" '(213100 (PPRF . "entfernt")))
(l-s "schmutz" 110300)
```

Thus, given the selection of a lexical item and a set of inflection features specified from the grammar, it is possible to map these into calls to MORPHIX as required. For example, if the inflection features were of the form {third-person-form present-form plural-form ...} then the inflection mapping mentioned above would help produce a call to MORPHIX as follows rather than providing information for accessing the (now empty) `:properties` slot of a lexical item:

```
(MORPHIX::VERB-INFLECTION "entfern"
   MORPHIX:PRAESENS MORPHIX:INDIKATIV
   MORPHIX::AKTIV 3 MORPHIX:PL)
=> "entfernen"
```

In this approach, interfacing between components is achieved simply by relating the features used in the grammatical component and those required by the morphological component. This preserves the modularity of the two components. Note that it would also be possible to change the inflection constraints used in a grammar to specify the necessary information for the morphological component *directly*: this would simplify the interface mapping, but at the cost of making that grammar dependent on a particular set of morphological features.

4 Generalized methods for interfacing with external morphology and lexical components

With the experience of the TECHDOC extensions, it was clear that the ability to link a generation grammar with externally developed components was a considerable time-saver and furthered re-use of existing components. Building on this, therefore, KPML now supports interaction between the generation process and external lexical components by providing methods (in the object-oriented programming sense) that (a) pinpoint the places in the generation process where access to lexical information is required and (b) may be specialized for particular languages as required. It is thus no longer, as was the case with the first MORPHIX integration, to make changes to the code of the generator. The methods defined provide a modular API for interfacing between generator and external lexicons and morphology components.

We show this briefly here for current work involving Russian.

Lexical items for Russian can be defined in the same way as used for English and German described above, but with more complex information provided in the morphological slot `:properties`. In general, the general API encourages information concerning morphological class to be placed properly in the `:properties` slot rather than in a separate specification as was the case with the MORPHIX solution. Two example lexical items for Russian verbs are the following:[7]

```
(LEXICAL-ITEM
     :NAME            zapustitj
     :STEM            ("запусти" "запущ")
     :SAMPLE-SENTENCE  "запустите команду Л"
     :FEATURES        (PERFECT VERB EFFECTIVE-VERB
                       DISPOSAL-VERB)
     :PROPERTIES      (( св  4 а))
     :COMMENTS        "start the L command")

(LEXICAL-ITEM
     :NAME            vospoljzovatjsja
     :STEM            ("восполь зова" "восполь зу" )
     :SAMPLE-SENTENCE  "... восполь
                        зовавшись "
     :FEATURES        (PERFECT VERB REFLEXIVE-VERB)
     :PROPERTIES      (( св нп 2 а -ся))
     :COMMENTS        "... using ...")
```

Sometimes grammatical and morphological data duplicate. For example, the aspect "perfect" is described as a grammatical feature, but it is duplicated on the morphological level ("св" in the `properties` slot). This is an example of a dissociation between lexicogrammatical functions (presented by features of a lexical item) and phonological/morphological structures (realized by the morphological generator). In this case the difference is justified by the fact that the combination of the perfective aspect and the unmarked tense (present) results in the morphological *future* tense. The second example of dissociation is the reflexivity of the verb *vospoljzovatjsja*: its reflexivity in grammatical terms is expressed in its feature 'reflexive-verb', but the fact that it has no non-reflexive morphological form is defined in its morphological index.

During grammar traversal, certain choices result in the specification of morphological properties, for example, in the Russian grammar when a direct complement is inserted, it is supplied with the realization statement: `(inflectify directcomplement accusative)`, in the manner described above. Then, when morphology is not being handled by a multilingual grammatical description in the form of a system network (as was the case in the Dutch example given above), then the KPML method:[8]

[7] The morphological information conforms to the elaborate classification system for Russian morphology developed by Zaliznjak (1977). A further extension here over the above the inclusion of lists of stems; this provides data for selection, in the case of verbs, between infinitival and personal stems.

[8] KPML is implemented in ANSI-standard Common Lisp. The methods referred to here are therefore defined in the Common Lisp Object System (CLOS) component of that standard. Non-latin characters are being represented here by a Unicode-coding throughout.

```
REALIZE-INFLECTIFY
  (chosen-word inflection-feature-list lg)
```

is called. This has been specialized for Russian so that the features selected from the grammar are analyzed and mapped appropriately for calls to the external lexical form generation procedures. These procedures are implemented in a library of classes which describe grammatical features of Russian words. This preexisting system is based on quite different principles than the generation grammar (Sharoff 1995). In order to generate forms listed in the sample sentences above, a Russian-specific method `generate-form` is called analogously to the use of MORPHIX above. Examples of such calls are then:

```
(generate-form zapustitj
   :form 'imperative :rnumber '(plur))
=> " запустите "
(generate-form vospoljzovatjsja
   :form 'adj-participle :tense 'past)
=> " восполь зовавшись "
```

This provides a clean interface between the two components, allowing both to do the particular aspects of the total generation task for which they work best, without requiring access to any internal details of the generation algorithm.

Both the German and Russian generation components as we have described them here make use of lexical items defined internally to the generation component (by means of the *lexical-item* definitions we have seen as examples above). Information contained in these lexical items (typically the `spelling` or `stem`) are subsequently modified by an external morphological component according to features selected by the grammar during generation and depending on the inflection classes denoted by the values in the `properties` slot. KPML now, however, also supports the option of maintaining the lexical resource database entirely externally to the generation resources. This is where a number of current experiments are focusing: given the current proliferation of lexical information (or at least of projects that are intended to produce such lexical information), it is important for MLG that its generation resources obtain some benefit.

To do this, the generation algorithm needs to be able to access the lexical features of a lexical item without requiring that that lexical item be maintained internally to KPML. This is managed by means of the method:

```
SELECT-TERMS-FROM-LEXICON
  (positive-features negative-features lg)
```

This method is responsible for returning a set of lexical items (i.e., some identifying labels for items) given a set of positively classifying features and a set of negatively classifying features. This is used in the grammatical filtering process by which a candidate set of lexical items that might be *semantically* applicable—by means of their being associated with appropriate domain concepts as indicated above—are reduced to just those items that may also fit the currently required grammatical context. The returned list then provides a starting point for processing according to the specified inflection features either by means of a grammatically coded morphology (as in the Dutch and French cases) or again by use of an external morphological component (which may be the same component as manages the lexical item selection but which does not need to be). The lexical items themselves need not be defined internally to the generation component as long as the morphology methods described above are able to obtain the information they require.

5 Interaction between lexical information and generation

When moving to consider German and Dutch, it was also necessary to address dependencies between lexical selections and grammatical selections that had not been so relevant for English. For example, the selection of a lexical item of a particular lexical gender, or lexical number, obviously has repercussions for other structural elements in the sentence being generated. Here a generalization of the Penman notion of the 'environment' of the generation system provided a solution while remaining within the deterministic generation framework that is required for practical generation. Information from the environment (originally the semantic specification for generation) is accessed in the Penman-style architecture by *inquiries* (cf. Matthiessen & Bateman 1991). The notion of 'environment' has now been extended to include not only knowledge representations but also *lexical* databases. Thus, an inquiry can rightfully ask questions concerning the particular lexical features that a lexical item possesses and, depending on the answers, trigger grammatical decisions accordingly. This mechanism imports dependencies on lexical information into the generation process at the points where they are necessary for grammatical decisions to be made. This re-orients the description slightly, but preserves the determinism necessary for efficient and practical generation performance.

This solicitation of lexical information by the grammatical generation process described above is now also supported by KPML API-methods specializeable for language. The methods provided are the following:

1. `ACCESS-LEXICAL-INFORMATION (function language)`
   This method takes a grammatical function (such as Process, Subject, etc.) and a language and produces three values: the features of the lexical item selected for the grammatical function, its name, and it definition.
2. `CHECK-WORD-RANK-IRREGULARITY (Chosen-Word Grammatical-Feature-List language)`
   This provides a generic test for considering that a particular use of a lexical item needs to be treated as being irregular instead of going on to regular morphology, however that may be defined. This enables the decision concerning whether, for example, a systemic morphology is to be used or rather a form is to be obtained directly from a (possibly external) lexical database to be properly parameterized.
3. `FETCH-LEXICON-FEATURES (lex-entry language)`
   This takes a name of a lexical item and a language and produces the features of that lexical item. If a grammar has been produced considering the lexical features that its lexicon employs, then the constraints used in classify-realization statements should match with those this method produces from some external lexicon. If not, some conversion wrapper must be defined.

Specializing these methods on particular languages for particular lexicons enables standard lexical access functions to interface seamlessly between the generation grammars as defined for generation within KPML and external lexical and morphological components—thus preserving maximal modularity.

6 Further issues and some discussion

Whereas we have discussed a variety of means for maintaining distinct morphological, lexical and grammatical components, it is also clear that there are interesting interactions that call for closer contact.

Consider, for example, two possibilities in Russian for the heading of a section from an instruction manual in the CAD/CAM domain:[9]

(a)	*Chtoby*	*narisovatj*	*poliliniyu*
	In order to	draw	polyline [acc]
(b)	*Risovanije*	*polilinii*	
	Drawing	of polyline	
	[nominalization]	[genitive]	

These correspond approximately to a distinction in English between the possibilities 'To draw a polyline' and 'The drawing of polylines' respectively; that is, the former is purposive and clausal, and the latter is a nominalization.

As a first approximation, all of these expressions (both English and Russian) might be generated from a partial semantic specification of the form (again using SPL notation):[10]

```
(d / draw
   :actor (user / person)
   :actee (p / polyline))
```

That is, both cases involve expression of a drawing action in which the semantic `:actee` is the domain concept 'polyline'; however (a) is realized by an infinitival clause and (b) by a nominal group. However, the situation in Russian appears more complex when compared to that in English since whereas in English both variants utilize the single word (*draw*), in Russian the nominalized expression requires the use of a different stem—the verb нарисовать (narisovatj) uses the 'perfective' stem while the noun рисование (*risovanie*) is built on the 'imperfective' stem. The availability of stems expressing this aspectual distinction is a central component of Russian lexical organization.

The selection of one stem rather than another appears to be grammatically conditioned and might not be thought to influence the semantic representation. Here a single semantic action would be maintained regardless of the fact that a variety of contrasting lexical forms are required depending on the particular grammatical realization selected. Doing otherwise would lead to an apparent repetition of tactical grammatical choices in the strategic text planner.

The selection of distinct forms still needs to be managed appropriately, however, and there are in theory two places to handle this problem: in the morphological component and in the lexicogrammar. If the selection of forms is placed within the morphological component entirely, then an appropriate nominalization of this kind would be produced simply by setting a 'nominalized' inflection realization constraint on the head noun of the nominal group. It would then be expected that the morphology component would have the necessary information to select an appropriate stem. Here again we have preserved a modularity between morphological and grammatical linguistic knowledge.

Placing the selection of form in the morphology component in this way ignores, however, the fact that the phenomena exhibited in (a) and (b) have semantic restrictions that the lexicogrammar is better placed to enforce. The nominalization of actions in Russian by means of the -ние (*nie*) suffix attached to an infinitival imperfective stem, for example, corresponds to a common semantic pattern: the nominalization is a *process* nominalization that presents a process view on the action being nominalized. Thus, while grammatically a nominal group, a more strictly accurate English gloss of (b) above would be on the lines of 'drawing polylines', thereby maintaining the processual perspective.[11] The selection of the perfective form in (a), the purposive clause case, also suggests that a more accurate gloss in English would be something on the lines of 'In order for a polyline to be drawn', thus capturing the perfective perspective on the action of drawing.

If the selection of imperfective/perfective forms of the kind considered here is made the responsibility of the grammar, then it is also necessary for the semantic specifications provided as input to the grammar to contain sufficient information for the grammar to make its selections. This simplifies the task of a morphological component at the cost of requiring a more complex temporal component in the semantics—but the need for a more sophisticated temporal semantics and event structure is in any case well established (cf. Moens & Steedman 1988, Pustejovsky 1991) and so should not be seen as something to avoid.

It is not always straightforward, then, to determine where information is best maintained. Information that appears largely idiosyncratic, morphological or lexical, may turn out to be susceptible to broader treatments when considered in terms of their communicative function, and to have ramifications for both grammatical and semantic representation. While distinct modules may be maintained in the ways described above, the contents of those modules may still need to show some drift as accounts develop.

7 Conclusions and future work

The functional descriptions of grammar used in generation are highly re-usable across languages and need to be maintained in any complete multilingual account. However, with the current state of the art, it is not clear that this grammatical organization and that of lexical resources are immediately commensurate. Therefore, in this paper, we

[9] This is the adopted domain of application of the EU AGILE project.

[10] Naturally further semantic contraints need to be added to distinguish the purposive element but that is not our main concern here.

[11] This form of nominalization contrasts with a further productive case in Russian using the suffix *stvo*, which regularly applies to perfective stems and produces meanings involving 'abstract things' or concrete things that realize the result of the process (such as 'guidance' in the contrast: *rukovoditj* 'to guide' *vs. rukovodstvo* 'guidance' or 'instruction manual').

have focused on methods for 'combining' the two kinds of organization—as embodied in large-scale resources developed from the two perspectives—in a manner that preserves the modularity of each. This combination has the practical goal of providing multilingual natural language generation capabilities with maximal re-use of existing resources: both formal-lexical and functional-grammatical.

Although we have in this paper drawn loosely on a distinction along the lines of 'formal-lexical' and 'functional-grammatical', this is, of course, not correct. Lexical information does not necessarily align only with form, nor does functional information limit itself to grammar. It would be singularly advantageous, therefore, if there were further dialogue between the grammatical information maintained in form-based lexicons and that maintained in functional grammar descriptions, as well as between the lexical information maintained in functional grammars and that found in form-based lexicons. Only by means of this kind of dialogue can a more unifying view of the distinct kinds of organization applied in this paper be found. Such a unifying view is certainly a worthwhile aim both theoretically and practically; until it is achieved, however, mechanisms such as those described in the present paper can be employed for at least allowing use of multilingual lexicons in one area—multilingual natural language generation—where they would otherwise find rather limited application.

ACKNOWLEDGEMENTS

We would like to thank Brigitte Grote and Knut Hartmann of the University of Magdeburg for detailed discussions and input concerning both the use of MORPHIX within the TECHDOC system and the general problems of interfacing between grammatical and morphological components. The work reported in this paper was partially funded by the European Commission under the INCO-Copernicus programme project AGILE: 'Automatic Generation of Instructions in Languages of Eastern Europe' (PL961004).

REFERENCES

Albano, R., Cumming, S. & Sondheimer, N. K. (1989). How to realize a concept: Lexical selection and the conceptual network in text generation, *Reprint Series RS-89-248*, USC/Information Sciences Institute, Marina del Rey, California.

Bateman, J. A. (1990). Upper modeling: organizing knowledge for natural language processing, *5th. International Workshop on Natural Language Generation, 3-6 June 1990*, Pittsburgh, PA. Organized by Kathleen R. McKeown (Columbia University), Johanna D. Moore (University of Pittsburgh) and Sergei Nirenburg (Carnegie Mellon University).
URL: *http://www.darmstadt.gmd.de/publish/komet/papers/general-description.ps*

Bateman, J. A. (1997). Enabling technology for multilingual natural language generation: the KPML development environment, *Journal of Natural Language Engineering* **3**(1): 15–55.

Bateman, J. A., Matthiessen, C. M., Nanri, K. & Zeng, L. (1991a). Multilingual text generation: an architecture based on functional typology, *International Conference on Current Issues in Computational Linguistics*, Penang, Malaysia. Also available as technical report of the department of Linguistics, University of Sydney.

Bateman, J. A., Matthiessen, C. M., Nanri, K. & Zeng, L. (1991b). The re-use of linguistic resources across languages in multilingual generation components, *Proceedings of the 1991 International Joint Conference on Artificial Intelligence, Sydney, Australia*, Vol. 2, Morgan Kaufmann Publishers, pp. 966 – 971.

Cahill, L. J. & Gazdar, G. (1995). Multilingual lexicons for related languages, *Proceedings of the 2nd DTI Language Engineering Conference*, Department of Trade and Industry, London, pp. 169–176.

Cahill, L. J. & Gazdar, G. (1996). A lexical analysis of numerical expressions in three related languages, *Proceedings of the AISB workshop on multilinguality in the lexicon*, AISB.

Degand, L. (1996). A Dutch component for a multilingual systemic text generation system, *in* G. Adorni & M. Zock (eds), *Trends in Natural Language Generation: an artificial intelligence perspective*, number 1036 in *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Berlin, New York, pp. 350–367. (Selected Papers from the 4th. European Workshop on Natural Language Generation, Pisa, Italy, 28-30 April 1993).

Finkler, W. & Neumann, G. (1988). MORPHIX: A fast realization of a classification-based approach to morphology, *Proceedings of the 4th. ÖGAI: Wiener Workshop Wissensbasierte Sprachverarbeitung*, number 176 in *Informatik Fachberichte*, Springer Verlag, Berlin.

Halliday, M. A. (1978). *Language as social semiotic*, Edward Arnold, London.

Henschel, R. (1995). Traversing the Labyrinth of Feature Logics for a Declarative Implementation of Large Scale Systemic Grammars, *in* Suresh Manandhar (ed.), *Proceedings of the CLNLP 95*. April 1995, South Queensferry.

Kasper, R. T. (1989). A flexible interface for linking applications to PENMAN's sentence generator, *Proceedings of the DARPA Workshop on Speech and Natural Language*. Available from USC/Information Sciences Institute, Marina del Rey, CA.

Matthiessen, C. M. & Bateman, J. A. (1991). *Text generation and systemic-functional linguistics: experiences from English and Japanese*, Frances Pinter Publishers and St. Martin's Press, London and New York.

Moens, M. & Steedman, M. (1988). Temporal ontology and temporal reference, *Computational Linguistics* **14**(2).

Paris, C., Linden, K. V., Fischer, M., Hartley, A., Pemberton, L., Power, R. & Scott, D. (1995). A Support Tool for Writing Multilingual Instructions, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 1995*, Montréal, Canada, pp. 1398 – 1404.

Pustejovsky, J. (1991). The syntax of event structure, *Cognition* **41**: 47–81.

Reiter, E. (1994). Has a consensus NL generation architecture appeared, and is it psychologically plausible?, *Proceedings of the 7th. International Workshop on Natural Language generation (INLGW '94)*, Kennebunkport, Maine, pp. 163–170.

Rösner, D. & Stede, M. (1994). Generating multilingual documents from a knowledge base: the TECHDOC project, *Proceedings of the 15th. International Conference on Computational Linguistics (*COLING 94*)*, Vol. I, Kyoto, Japan, pp. 339 – 346.

Sharoff, S. (1995). An application of object-oriented programming for linguistic modelling, *Proc. of the international workshop DIALOGUE'95*, Kazan, pp. 332–339. (in Russian).
p

Stede, M. (1996). Lexical options in multilingual generation from a knowledge base, *in* G. Adorni & M. Zock (eds), *Trends in natural language generation: an artificial intelligence perspective*, number 1036 in *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 222–237.

Zaliznjak, A. (1977). *Grammatical Dictionary of the Russian Language*, Russkij Jazyk, Moscow. (in Russian).