

Multilingual resource sharing across both related and unrelated languages: an implemented, open-source framework for practical natural language generation

John A. Bateman (bateman@uni-bremen.de)
University of Bremen

Ivana Kruijff-Korbayová (korbay@coli.uni-sb.de)
University of the Saarland

Geert-Jan Kruijff (gj@dfki.de)
German Research Center for Artificial Intelligence

Abstract. This article reports on our experience with developing multilingual grammar resources for natural language generation. We employ a strong notion of multilinguality: (i) Grammars for different languages share their overall organization, as well as those descriptions that reflect similarities between languages, and (ii) a single realization engine is used to generate with these grammars. This strong notion arises from the functionalist approach we adopt: We hypothesize that languages are likely to share communicative functions, despite possibly differing in how these functions are realized. We discuss the advantages of this view in the development of large-coverage generation grammars for a broad variety of languages.

1. Introduction

In this article, we present a strongly multilingual approach to grammar development, orienting ourselves primarily at natural language generation (NLG). At the heart of this approach is the adoption of *integration* and *integrity* as fundamental principles, resulting in grammars of different languages that share their overall organization as well as those linguistic descriptions that the languages have in common. Integration maximizes the factoring out of generalizations across the languages and the particulars of individual languages, whereas integrity supports consistent access from both the multilingual viewpoint and the viewpoint of each individual language resource.

The strong notion of multilinguality arises from a *functionalist* perspective on grammar organization: Although form may vary across languages, we hypothesize that the underlying communicative *functions* are largely shared. In other words, we take paradigmatic choices to be more often congruent across languages than the syntagmatic ones. We do not, however, enforce a single common paradigmatic organization. The advantage of this approach is that it allows flexible and variable congruence across linguistic descriptions: Paradigmatic choices can be



© 2005 Kluwer Academic Publishers. Printed in the Netherlands.

shared as much as possible but no more than needed, while at the same time they can be made sufficiently fine-grained to accurately reflect language-, genre- and register-specific realization preferences.

Over the past ten years we have put this approach to use in the development and use of multilingual grammars for NLG applications. This has mostly involved the provision of general strategic and tactical generation components that are intended both to cover some particular application register—such as, for example, instructional texts—and to provide firm foundations for extension into other registers. This serves to combine both *system-oriented* multilinguality, where we compare and construct resources for an entire language, and *instance-oriented* multilinguality, where we compare and contrast particular text instances as required in the languages and registers under study. Our generators must therefore be able to produce surface rendition(s) of semantics that are appropriate both for particular text types and for structural positions within those text types. Such renditions are determined by the practices of the target language and culture, and vary considerably: they cannot be derived from semantic specifications of individual sentences isolated from their textual context.¹

A simple example of this, which we will return to later in more technical detail, is the generation of headings and subheadings in instructional texts. Whereas the English subheading in a Computer-Aided Design software manual is “To draw a polyline”, the corresponding natural headings in Bulgarian, Czech and Russian were found to be:

Bulgarian	<i>Chertane</i>	<i>na polilinija</i>	
	drawing-nom	of polyline	
Czech	<i>Nakreslení</i>	<i>křivky</i>	
	drawing-nom	polyline-gen	
Russian	<i>Chtoby</i>	<i>narisovatj</i>	<i>poliliniju</i>
	in-order-to	draw-infin	polyline-acc

Here we see that, on the one hand, English and Russian are similar to each other in that they opt to produce a purposive-like clause construction whereas, on the other hand, Bulgarian and Czech are similar to each other (and different from English and Russian) in that they employ a nominalizing construction. Nevertheless, even though the English and Russian grammatical systems have nominalizing constructions analogous to those of Czech and Bulgarian, and both Czech and Bulgarian have purposive clausal constructions analogous to the Russian and

¹ Although this has a close relationship to work on translation, space precludes further discussion here; see, e.g., Bateman (1992) and Hartley and Paris (1997).

English, it would still not be appropriate to use the same form in all four languages. Despite their availability in the respective language systems and the obvious similarity of the underlying semantics, register considerations rule this out.

This raises several crucial issues for effective multilingual generation. First, we need effective ways of generalizing our descriptive statements across diverse languages: here, for example, we need to capture the observed similarities between the English and the Russian forms, even though the fine structural details of the corresponding constructions are different. Second, we need to be able state both similarities and differences in the preferred strategies selected for expressing a given semantics: e.g., the English/Russian selection of purpose clauses *vs.* the Bulgarian/Czech selection of nominal forms. And third, these preferences must all be related to particular registers, text types and places in document structure, since they are not free-floating preferences but are closely tied to their contexts of use. All of these issues need to be dealt with simultaneously. Moreover, the correct choice of forms can only be determined on the basis of corpus examination of the actual styles used for the text element in the particular text type in the languages under study. We will see below how our approach to multilingual generation combines these aspects.

Such variation in forms and preferences is widespread when considering documents from different languages and represents a major research interest of NLG as such: NLG is crucially concerned with the precise reasons for choice of particular forms over others (cf. McDonald, 2000) and has established that these reasons are often only specifiable in textual and registerial terms. Once these reasons for choice have been isolated and incorporated into the generation process, generation can aim to produce texts that are sensitive to the stylistic requirements of not only the output language in general, but also of various sections of documents and of various types of intended readers.

The instructional text examples above are taken from our work on the Agile project (Automatic generation of Instructions in Languages of Eastern Europe)² in which we constructed extensive grammars of Bulgarian, Czech and Russian. The resulting system provides variation of generated results according to document and reader, and allows for motivated variation in style—such as that between the titles and body in instructional texts as shown above or, rather more sophisticated, that between the instruction sections of a software manual presented in either a personal or impersonal style. In the personal style, the reader

² EC INCO-COPERNICUS project PL961104. <http://www.itri.brighton.ac.uk/projects/agile/>

(software user) is addressed directly; in impersonal style, the tone is more formal. In our evaluations of the system we found that the generated texts were comparable to texts produced by native speakers when writing instructions in the three languages targeted by the project; further details and sample generated texts are presented in Kruijff et al. (2000).

In developing the generation grammars for the Agile system, we followed the strategy of maximal re-use of existing resources supported by the KPML multilingual linguistic resource development environment (Bateman, 1997).³ This methodology has also been applied in similar projects involving German, Dutch, Spanish, French and Chinese, as well as in smaller scale experiments for Greek and Japanese. In all cases the goal has been to achieve linguistic resources that support flexible generation of connected texts with varying styles and for various text types.

In the rest of this paper, we explain the details behind this work, motivating the approach taken and the kinds of linguistic descriptions employed. We begin in §2 by placing our approach within the historical context of multilingual NLG in general. This is important for understanding many of the particular architectural and theoretical decisions that we build on later. In §3 we then turn to the actual organization that we adopt for linguistic resources. We also make a particular effort here to relate our framework to other kinds of approaches to multilingual resource engineering that are now emerging, such as the HPSG-based LinGO Grammar Matrix (Bender et al., 2002) and the LFG-based ParGram (Butt et al., 1999) initiatives. Finally, in §4 we summarize the methodology that our approach has led to and exemplify its achieved re-use statistics with respect to some of the full-fledged grammars that we have developed.

2. The multilingual natural language generation perspective

Unlike work in natural language understanding (NLU), NLG has always needed to concern itself with issues of text structure and planning as much as it does with grammatical realization and the mapping of semantics to linguistic form (cf. Vander Linden, 2000). An argument has to be structured effectively to make its point, a set of instructions needs to make it clear what is being instructed, and a weather report has to look like a weather report rather than an encyclopedia entry on climate.

³ KPML stands for 'Komet-Penman Multilingual', showing the two generation systems it was developed from and the fact that it was re-engineered to be inherently multilingual.

This unavoidable orientation to text organization has consequences for the less abstract levels of representation and, in particular, for grammar, because particular forms must be associated with appropriate pragmatic as well as semantic information. Our own framework is no exception and requires input specifications for surface generation that include information concerning the intended speech function, discourse recoverability, rhetorical structuring and information structure.

A related important question for *multilingual* NLG (henceforth MLG⁴) is the degree of relatedness or, as we shall term it, *congruence* between the various levels of representation present in a generation system across the different languages that the system is to deal with. A typical decomposition of representational layers for NLG is that set out in the Reference Architecture for Generation Systems (Cahill et al., 2000). This includes a level of grammatical, lexical and morphological layers, a semantic representation for clauses, a discourse-level representation along the lines of rhetorical structure theory (Mann and Thompson, 1988), and domain knowledge. For a multilingual generation system—for example one that is to generate weather reports, news reports or software instructions in several languages—the obvious question is then at what level do issues of multilinguality first arise.

We can distinguish here a spectrum of systems in which progressively more system resources are shared across languages. At the lower end of the spectrum we find systems that we term, following Rösner (1994), *weakly multilingual*. Such systems achieve multilinguality by replacing the grammar of one language by the grammar of another, other components remaining unaffected. In the context of MLG, this assumption views the generation process as one in which sentence content is planned independently of the target language and texts are formed by planning sequences of sentence content specifications. Multilingual versions of texts can then be produced in parallel simply by replacing the syntactic information maintained in the tactical generator with that appropriate for another language.

⁴ Multilinguality has had a long history in NLG. The first MLG work included Goldman's (1975) BABEL as well as several early landmark generation systems originally developed for a single language (typically, but not always, English) that were later extended to encompass generation capabilities in other languages. Examples include the extension of Kukich's (1983) ANA English database report generator to French (FRANA: Contant, 1986), the conversion of Rösner's German news report generator (SEMTEX: Rösner, 1987) to English (Rambow, 1988), Kehl's (1989) bilingual German/English geometry description generator GEOTEX, Danlos's (1987) generation of French and English, Maybury's (1989) generation of English and Italian, Jacob and Maier's (1988) extension of McDonald's Mumble system to generate German, and Peters et al.'s (1991) KLEIST system for Japanese and German generation.

The effort of the latter step is reduced further when information from existing grammars is re-used for new grammars. This strategy, sometimes called ‘grammar porting’ (e.g., Kim et al., 2003), appears to represent a saving in effort regardless of the linguistic framework adopted:⁵ it appears that *any* linguistic description that has achieved a sufficient degree of maturity and comprehensiveness to cover a useful fragment of a language is going to offer a beneficial starting point for a comparable description of any language that is sufficiently ‘similar’ to the originally covered language. We return to the question of whether this can be improved upon below.

Considering ‘equivalent’ texts from differing languages quickly establishes, however, that a representation of difference purely within the grammar and lexicon is not sufficient. For natural equivalence sentence boundaries will often be different and even the semantic content of sentences will vary. A number of telling examples are discussed by Kittredge (1995) in the domain of ‘Labor Force Surveys’, and by Bateman et al. (1999) in the domain of instruction manuals. Wide divergences occur even between languages that are closely related typologically and quickly go beyond the kinds of sentence-to-sentence divergences observed in machine translation work (cf. Dorr, 1994).

The response to this problem in MLG was then to consider the effects of multilingual variation at increasingly higher levels in the generation system architecture. For example, in the TECHDOC system (Rösner and Stede, 1992) for the generation of technical documentation (originally in English and German), there is no requirement that the sentence generator inputs for a given set of ‘translationally equivalent’ texts be identical: What is held constant across languages is the rhetorical structure of the texts to be generated. However, based on corpus work on instructional texts in Portuguese, Italian, English, French and German preparatory to designing the resources for the DRAFTER-I multilingual text generation system, examples were found which even appeared to require different rhetorical structures. One such set of contrasting examples discussed in Delin et al. (1994) is shown in Figure 1. The distinct rhetorical structures posited as text-level descriptions of the three ‘translations’ are quite different in their communicative force.

Various conclusions can be drawn from these observations and, despite the suggestion of Vander Linden and Scott (1995) that the only level at which an MLG system can really share representations across languages is the knowledge base of an application, there is still room for significant linguistic system-related congruence. From a typological

⁵ As evinced by the wide range of frameworks where the strategy has been applied (Jacob and Maier, 1988; Lee et al., 1991; Alshawi et al., 1992; Rösner and Stede, 1994; Rayner et al., 1996; Novello and Callaway, 2003; Kim et al., 2003).

- English: *The stepping load can be altered by loosening the locking lever and changing the position of the cylinder foot.*
- French: *Pour modifier la charge d'appui, desserrer les leviers puis déplacer le pied des vérins.*
- German: *Nach Lockern der Klemmhebel kann durch Verschieben des Zylinderfußes die Tretbelastung verändert werden.*

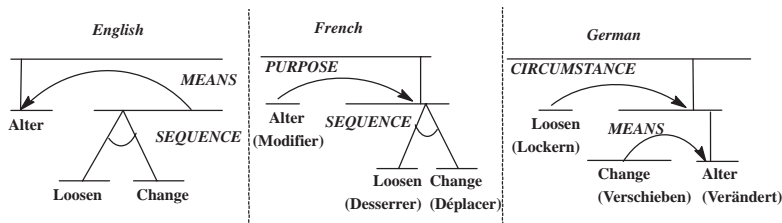


Figure 1. Examples of varying rhetorical structures across translation equivalents found in work on the DRAFTER system (Delin *et al.*, 1994)

viewpoint, this corresponds to rejecting the Chomskyan idea of a Universal Grammar, replacing it with the notion of a Typological Universal Grammar where it can be assumed both that languages differ and that shared characteristics will nevertheless be observed. Thus, although the German, French and English rhetorical organizations proposed by Delin *et al.* (1994) are different, they are all drawn from a collectively shared and multilingually valid statement of rhetorical structure. The question remains, therefore, to what extent such system-level congruences can be maintained even though, as we saw in §1 with our example CAD-software manual headings, individual texts and languages may make very different use of those systems during the generation of particular texts. Maintaining congruence of description as far as and wherever possible is the position that we term *strong multilinguality*.

In the following sections, we focus on the technique that we have developed for constructing shared representations primarily at the grammatical levels of description, in the context of a system exhibiting strong multilinguality.

3. Multilingual linguistic descriptions within the KPML system

In this section, we set out the kinds of linguistic descriptions used within our framework, suggesting how these relate to similar layers of representations in other frameworks. Our linguistic descriptions are

shaped directly by their theoretical foundation in systemic-functional linguistics (SFL: Halliday and Matthiessen, 2004 and Matthiessen and Bateman, 1991). Accounts of this kind are decomposed along several orthogonal dimensions of linguistic description. Those that we focus on here and relate to multilinguality are:

- Stratification (degree of linguistic abstraction): i.e., lexicogrammar *vs.* semantics *vs.* genre/register,
- Axiality: paradigmatic (network, i.e., essentially type lattices) *vs.* syntagmatic (functional structure),
- Rank (hierarchies of structural units): e.g., clause *vs.* groups/phrases *vs.* words *vs.* morphemes.

As we shall see, all of these are fully implemented within the KPML development environment and play an active role during resource development and maintenance (cf. Bateman (1997) and <http://www.purl.org/net/kpml>). The linguistic information within any stratum is represented in terms of paradigmatic and syntagmatic descriptions (by lattices and functional structures respectively) in which the functional structures are decomposed hierarchically according to distinct structural ‘ranks’. We can then state, for example, that our grammatical descriptions are situated within the stratum of lexicogrammar and are therefore made up of a lattice of grammatical paradigmatic ‘features’ and associated syntagmatic constraints holding over structural units of different grammatical ranks (e.g., clause, phrase, etc.).

The essential move that we have made to incorporate multilinguality is then to provide, both theoretically and within the KPML implementation, a further organizing dimension of *language variety*—interpreted for the purposes of this paper as ranging over individual languages. This addition applies most directly to axiality, allowing language variety conditionalization for all paradigmatic and syntagmatic specifications at any stratum.

3.1. GRAMMAR AND SYSTEM NETWORKS

The basic organizing construct for all systemic-functional linguistic descriptions is the classification network of the paradigmatic axis. These are the *system networks* from which SFL takes its name (Halliday, 1966). Each point of diversification in a system network represents a point of abstract functional choice: a place where further refinement requires appeal to semantic distinctions in order to be made appropriately. Each such point of alternation is called a *system*. A simple example, which we will embroider as we proceed, is the system that

presents the grammatical alternation between ‘declarative’ and ‘interrogative’. This distinction is motivated grammatically in the English grammar because the grammatical structures that fall under the two terms differ systematically: all subtypes of ‘declarative’ have a syntagmatic structure that differs from those under ‘interrogative’. But, more importantly, there is also a distinction in the functional work in discourse that the two alternatives perform: they are different *inter-actionally* and support diverging possibilities for the continuation of a text or dialogue. It is in this sense, then, that the network alternations are *functionally* motivated.

Formally, system networks are equivalent to subsumption lattices over sorts in typed feature structures (cf. Copestake, 2002) or over terminological concepts (TBox) in a description logic (cf. Baader et al., 2003); this relationship has been characterized most closely by Henschel (1997) building on earlier work of Kasper (1988), Mellish (1988), Brew (1991) and Bateman et al. (1992). This also offers a useful point of similarity with linguistic accounts that rely on typed feature structures, such as HPSG. The grammar features found in an SFL system network play an analogous role for the linguistic description as the sorts of the type lattice in HPSG. It is across grammatical features organized as a system network that inheritance of structural information flows. Partial structural fragments specified in the system network must then be well-typed just as in HPSG. However, as indicated above, SFL places particular descriptive weight on the system network. Each grammatical system making up the network consists of a named disjunction over immediate subtypes. This is used to capture the ‘minimal functional alternations’ at the heart of the account.

Another useful point of comparison is provided by the LFG distinction between f-structure and c-structure. The SFL descriptions we find under lexicogrammatical syntagmatic representations correspond almost directly to LFG f-structures. There are, in the SFL description, no corresponding c-structures however. SFL syntagmatic constraints consist of structural fragments linked to particular grammatical features of the system network. The constraints linked with features belonging to a single system should capture exactly and only how the alternatives in that alternation differ from one other. The constraints themselves are called *realization statements*; they constrain linear precedence, immediate dominance and ‘unification’ of functional constituents, and introduce further type constraints to hold for subconstituents. A detailed description of the realization statements and of their place in a systemic description is given in Matthiessen and Bateman (1991). The fragments given by a complete paradigmatic description are sub-

sequently combined in order to build actualized syntactic structures, i.e., the structural results of generation.

There are also some significant differences between the SFL syntagmatic constraints and LFG f-structures. First, there is no use of re-entrancy in the SFL structures (thus restricting unification to unification over atomic values). And second, a far broader range of ‘grammatical functions’ is adopted. This latter follows from the *multifunctional* commitment of SFL linguistic descriptions in which, in addition to the ‘generalizing’ functions Subject, Object and so on, there are extensive sets of functions corresponding to the roles of different kinds of process types (Actors, Goals, Recipients, Beneficiaries, Sayers, Sensers, and so on) and to different textual statuses (Themes, Rhemes, Given, New and so on). A systemic functional structure is then typically ‘layered’ in that groups of functional elements drawn from different functional components are unified during generation. An element may thus serve simultaneously as grammatical Subject, grammatical Actor and grammatical Theme. This is the only kind of structure sharing that is employed in the framework. Linear precedence is then defined directly over these grammatical functions rather than over constituent tree configurations.⁶

The potentially confusing partial overlap in terminology across systemic-functional grammar, HPSG and LFG can perhaps be helped by the following diagram, in which some important correspondences across the frameworks are highlighted:



The points of contrast and similarity reflect well the priorities taken up in the originally more NLU-centered accounts of LFG and HPSG and the NLG-centered account of SFL. Whereas the LFG f-structure is anchored via constraints spread over the c-structure configurations, the SFL syntagmatic structures are anchored via constraints spread over the paradigmatic type hierarchy of the system network. Their use in generation is then given by insisting on their being well-typed according to the system network rather than by phrase structure. Then, since the lattice organization in SFL is directly related *upwards*, via its functional motivation, to the stratum of semantics, this makes it easier to select constraints on the basis of a given semantics (the tactical generation task). This makes a considerable contribution to effective

⁶ A complete introduction to the functions and their layering possibilities is given in Halliday and Matthiessen (2004).

multilinguality. Our experience has been that substantially more of the system network may then be maintained unchanged across languages than might otherwise have been expected.

Broad-coverage monolingual generation grammars in this framework start typically at around 700 or more grammatical systems defined over 1500 or more types. Broad-coverage in the context of the ‘generic NLG’ at issue here is taken to mean that the majority of constructions required for expressing texts in any genre and style are already available within the grammar and have been linked appropriately to semantic input parameters. The resources provided are thus explicitly intended to be task, genre and domain independent. The large grammar of English used with the KPML system, Nigel, has been under development since the mid-1980s (cf. Mann and Matthiessen, 1985) and is a good example of a broad-coverage generation grammar of this kind; substantial lists of examples documenting the coverage of this grammar are available from the KPML website.

Generation as a whole is then seen simply as semantically guided type refinement: each refinement is motivated by a specified semantic configuration and brings with it further constraints on grammatical form via a combination of the underspecified functional structures defined by the realization statements.

We can illustrate this briefly as follows. Prior to the generation of a single grammatical unit, a semantic specification is formulated (either by a grammar developer for testing or by a text planning component); we return to the form of these semantic specification in §3.3 below. Generation is then driven by the grammatical network. The first alternation found in any systemic grammar is that between the available ranks in that grammar: i.e., a clause, group (nominal, verbal, etc.) or prepositional phrase, word or morpheme. The selection among these alternatives is mediated by an interface that, on the one hand, examines the semantic configurations in the input and, on the other, selects a particular grammatical alternative from those ‘on offer’. These mediated decisions are all local to each individual grammatical system or choice point. They typically examine semantic types, the presence or otherwise of semantic roles, and additional information concerning the textual statuses of discourse entities. If the semantic specification contains speech act information and an event-like propositional content, then the alternative ‘clause’ is selected and type refinement can continue; otherwise a rank such as ‘nominal group’ may be selected depending on further choice criteria. The alternations encountered subsequently cover valency, diathesis alternations, fronting and other ordering phenomena, tense and modality selections, noun phrase structures, etc. (cf. Matthiessen, 1995).

As refinement proceeds, syntagmatic constraints are triggered whenever their associated grammatical feature is selected. Since the grammar and its functional structures are required to be well-typed, it can be assumed that whatever syntagmatic constraint is reached, it can be added safely to the set of constraints appropriate for the grammatical unit being generated. If this is not the case, the grammar is considered to be in error; such eventualities must be weeded out during development by test runs and whatever static checking can reveal.

Mediation between grammar and semantics is maintained in a strictly modular fashion by channeling all the semantic information through the grammatical functions of the syntagmatic structure. Thus, particular linking rules are activated during traversal of the network in order to associate grammatical functions with entities in the input semantics. These associations define the areas of semantic content that are examined when making choices about which paths to refine further in the network. Moreover, since the grammar is in control of precisely which links are formed, type refinement can be made to ‘consume’ the semantics as is required for the particular syntactic structures that are being formed.⁷

The type lattices employed within a typical systemic grammar rely heavily on both multiple disjunctive and conjunctive inheritance in order to express linguistic generalizations. Conjunctively related types require the syntagmatic constraints collected (i.e., contributed by each type and its supertypes) to be combined (unified). Disjunctively related types present alternative paths for subsequent refinement according to the semantics to be expressed. This kind of formalization then requires specialized implementation techniques in order to remain practically useful; a full unification implementation is still prohibitively slow for the size of networks generally employed for realistic generation.

Both KPML and the Penman system upon which it builds therefore provide a deterministic left-to-right (i.e., specificity-increasing) traversal of the classification network in order to produce semantically and contextually appropriate linguistic forms. The framework also permits semantic specifications to remain silent concerning preferred lexicogrammatical alternatives. When further type refinement is not driven directly by a semantic constraint, a range of defaults comes into play. These can be dynamically fine-tuned and allow the framework to model very fine register-based selections or sublanguages. Similarly, when linear precedence constraints do not combine to uniquely specify an or-

⁷ A short but slightly more detailed illustration of generation is given in Bateman et al. (1999); substantially more detailed traces of clause generation are given in Mann and Matthiessen (1985) and Appendix II of Matthiessen and Bateman (1991).

dering, further conditionalizable defaults are used to produce an appropriate result.⁸ With this implementation, sentences of lengths of 30-40 words are typically produced from a semantic specification within a second or so on average performance machines. Generation time varies approximately linearly with the length of sentences.

The generation process can be restricted effectively to a deterministic traversal by accurately tailoring the constraints from the type hierarchy in a way that guarantees that only strictly conforming contributions to the partial functional grammatical configurations are formed. That is: traversal cannot be blocked by a failure of unification of partial structures. While this is clearly a far weaker descriptive framework than that supported by full unification, it is the price that we pay for fast and robust generation performance.

3.2. INTRINSIC MULTILINGUALITY

In §2 above, we saw that the experiences of MLG raise little hope for a ‘safe haven’ as far as multilingual effects are concerned. Our design within the KPML framework has therefore pursued a generic approach to multilinguality that allows any statement made in a linguistic resource to be further positioned according to the language or languages that it holds over. This approach builds on two goals for multilingual description set out originally in Bateman et al. (1991) and Matthiessen et al. (1991). These goals, although potentially conflicting, need to be reconciled for a truly effective multilinguality:

- **Integration** of the different languages so that commonality is separated from particularity and re-used: resources should *maximize the factoring out* of generalizations across the languages of the system and the particulars of individual languages in the linguistic resources;
- **Integrity** of each individual language so that it can be used separately: integrated resources should support consistent access from *both* the point of view of their multilinguality *and* the point of view of the individual languages.

The following simplified example shows this in action; the simple type ‘medium-qualifier’ defined in Figure 2 indicates a shared point in the feature lattice for the three languages Bulgarian, Czech and

⁸ Space again precludes us here going into these interesting aspects of generation. For a stochastically-driven version of the generation process, for example, see Langkilde and Knight (1998). For a use of ordering defaults conditionalized by textual information-status, see Kruijff-Korbayová et al. (2002).

```

(gate
 :name MEDIUM-QUALIFIER
 :inputs processual-mediated
 :outputs
  ((1.0 medium-qualifier
    :bulgarian :czech :russian
                    (preselect Medium nominal-group)
    :czech :russian (preselect Medium genitive)
    :bulgarian      (insert MediumQualifierMarker)
                    & (lexify MediumQualifierMarker na)
                    & (order MediumQualifierMarker Medium)
  ))
 :region QUALIFICATION)

```

Figure 2. Shared system for Bulgarian, Czech and Russian

Russian. Here we are within the lattice area that is concerned with expressing events through the linguistic resources of nominal groups (i.e., nominalizations) such as is required for our Bulgarian and Czech section heading examples in §1. This particular type is placed further within the functional region Qualification, one of the regions defining construction possibilities within nominal groups in general.

The type is determining how a ‘goal’ or ‘theme’ (indicated in systemic-functional terms by a constituent functionally labelled as the ‘Medium’) should be expressed in the nominalization. The syntagmatic constraints defined are to a certain extent distinct for the three languages: in this case, Czech and Russian pattern together and use a genitive nominal group while Bulgarian selects instead a pre-nominal marker ‘na’. All three languages agree that the Medium should be a nominal group however, and Czech and Russian agree further that the Medium should receive genitive case marking. We can illustrate these alternatives concretely by filling in the relevant parts of the syntagmatic functional structure for our CAD-software section headings of §1 as shown in Table I. Here it is interesting to note how the grammatical congruences cross-cut the usage congruences. Within the lexicogrammar it is Czech and Russian that are similar, whereas in the preferences observed for section headings, Czech groups with Bulgarian. This again underlines our claim that statements of multilingual congruence need to be separated across each of the orthogonal dimensions of description that we employ in order to maximize the opportunities for resource re-use.

The representation used in Figure 2 is the normal text-based input representation used for defining a type lattice in KPML. Each such definition defines a disjoint collection of types, named under the `:outputs`

Table I. Structural results for ‘medium-qualifier’ across Bulgarian, Russian and Czech

Bulgarian	Medium- QualifierMarker <i>na</i> of	Medium [nominal-group] <i>polilinija</i> polyline
Czech		Medium [nominal-group, genitive] <i>křivky</i> polyline-gen
Russian		Medium [nominal-group, genitive] <i>polilini</i> polyline-gen

slot. These types define the grammatical features that belong to a given grammatical system. The system is given a name with the `:name` slot. The position of these types within the lattice as a whole is indicated by the `:inputs` slot. In general, an input slot will be a boolean expression over supertypes involving conjunction and/or disjunctions; negation is not supported within KPML. The definition’s allocation to a functional region is given by the `:region` slot. The introduced paradigmatic types may specify syntagmatic constraints associated with them; this is then expressed in brackets following the type name—i.e., in the current example, following the type name ‘medium-qualifier’.

This is all exactly as is used for monolingual descriptions apart from the addition of *language conditionalization*. At any point in the specification a language condition can be added (e.g., `:czech`, `:chinese`, etc.) and this indicates that the following item in the specification is only relevant for the indicated language. Language conditions can be strung together and their effects last over the following component of the specification plus any further components that are joined with the specified ‘knot’: `&`. In Figure 2, therefore, we see that the grammatical function Medium is constrained (preselected) to be a nominal group for all three languages, but in Czech and Russian it is further constrained to be in the genitive case, whereas in Bulgarian an additional functional constituent is introduced (MediumQualifierMarker) which is subject to its own constraints.

This representation is used by KPML both as a contribution to the definition of the type lattices for all three languages and as an additional multilingual contrastive statement. It is this latter aspect that is of particular interest multilingually and for multilingual development. It

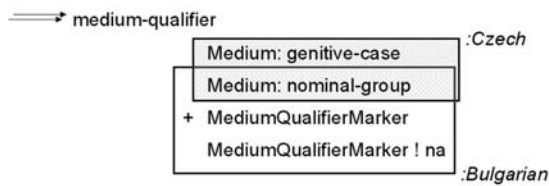


Figure 3. Bilingual (Czech-Bulgarian) view of a portion of the multilingual type lattice

follows directly from the fact that we support both integration and integrity. A properly contrastive view entails both: we see how resources relate to one another and how they each are unique. At any point, we can consider languages individually, as collections of more or less related languages, or as a single integrated multilingual resource. In Figure 3, for example, we show a graphical rendition of a bilingual view of the information of Figure 2, but here focusing only on the Czech-Bulgarian contrast. This is one of the basic presentation modes provided for developers by the KPML environment.

This has as a consequence that developers are always aware of the language extent of the resources with which they are working, i.e., if they are making changes for one language or several. If they select to make a change for a range of languages, then they must also ensure that those languages are not compromised by the change. If, however, they decide to work with only one language, then the other languages are not effected; they are sealed off by the appropriate conditionalization.

Although the multilingual mechanism provides in theory for an extremely high granularity in multilingual description, in practice (and as motivated by the empirical linguistic facts) we find that there are clear lines along which languages tend to differ or resemble one another. The most significant gain for multilingual resource construction has been found to correspond to the clear theoretical separation adopted between a paradigmatic and functionally-oriented classification hierarchy, as represented in the systemic network, and the functional syntagmatic configurations. Paradigmatic organizations may often be maintained as congruent across language descriptions. The paradigmatic-syntagmatic distinction then enables this congruence to be preserved *even when features are expressed structurally in differing languages in quite different ways*. The value of maintaining a congruent paradigmatic description is then that all of the semantics-grammar linking rules still apply: these rules only make reference to grammatical features from the type lattice and so maintaining the type lattice unchanged is a clear win in terms of re-use.

This is in many respects analogous to the situation generally cited for LFG, where a divergent c-structure can be effectively ‘absorbed’ by the f-structure (Bresnan, 1982); we take this one step further and absorb differences in the f-structure so that they are not visible to the type lattice organization over such structures. Within the more inclusive view of functional structure employed within a systemic-functional description, there is no expectation that this represent a ‘universal’ level of linguistic description as claimed for LFG f-structure (Bresnan, 1982; Butt et al., 1999). There is also no stipulated requirement that paradigmatic descriptions of distinct languages must be entirely congruent: this would reduce the account to a, clearly false, grammatical interlingua. Intrinsic multilinguality means that congruency can be maintained when and where it occurs but is not enforced when the language facts would be violated: this is, then, for us, ‘strong’ multilinguality taken to its logical conclusion.

3.3. SEMANTICS AND THE INPUT FOR TACTICAL GENERATION

The semantic stratum within the SFL account also has to support the input specifications for tactical generation. That is, a semantic specification has to provide sufficient information for an appropriate linguistic surface form to be generated. This then of necessity will often include information in addition to a truth-conditional semantic representation. Such additional information is commonly provided by the text planning processes alluded to in §2 above.

The form of semantic specification used within KPML is based on the Sentence Plan Language (SPL), designed for use with the Penman text generation system by Kasper (1989). It is a tribute to Kasper’s careful consideration of the prerequisites of a semantics for generation that the representation has proved so long-lived. Formally, a Sentence Plan Language specification, commonly termed an SPL, consists of a list of typed variables representing instances of semantic configurations and grouped together with additional roles, relations or properties that may hold for the instance:

```
( ... ( v0 / type-0
      :role-1 v1 :role-2 v2 ... :role-N vN)
  ...)
```

The semantic types are drawn from a multilingually validated hierarchy undergoing progressive development and extension as further languages are considered (cf. Henschel and Bateman, 1994, Bateman et al., 1995 and Halliday and Matthiessen, 1999). The hierarchy consists of around 200 types; we are also currently constructing a version expressed in the Web Ontology Language, OWL (Smith et al., 2004).

Although portions of the hierarchy are conditionalized for particular languages in precisely the same way as with the lexicogrammatical stratum, very much less use has been made of this facility. All of the grammars currently defined for use with KPML in effect make reference to the same semantic hierarchy. We do not, however, consider this to be an indication of the ‘universality’ of the result: we have no doubt that more detailed semantic investigations will uncover evidence of significant divergence.

SPL specifications also commonly contain properties concerning the information status or the attitudinal position (e.g., probability, obligation, appraisal) adopted with respect to some semantic element. They thus form a collection point for the kinds of varied, multifunctional semantic inputs that are motivated by the requirements of NLG. The origins and definitions of these diverse kinds of information are maintained by the distinct semantic hierarchies from which their types are drawn. The situation is again exactly analogous to that found in the lexicogrammar. The system network divides its work according to functional regions, thereby separating out areas of the lattice that are concerned with predicate-argument structure, interaction, information structure and so on. These distinct strands of functionality must be combined, however, in the functional syntagmatic structures produced. Each single string generated represents all of the functional strands that went into its construction. The SPL specification therefore serves the role of a syntagmatic structure, combining diverse sources of information, but at the semantic stratum rather than the lexicogrammatical. Finally, the semantic types actually found in semantic specifications may also be subtypes, or subconcepts, defined for populating a particular domain model; these are then subordinated to types of the general semantic lattice in order to simplify interaction between a generator and its application in the manner described in Bateman (1990).

As an example, the following SPL fragment corresponds to the English nominal group “the old red house”.

```
((b / house :identifiability-q identifiable
      :quantity 1
      :relations ((ca aa)))
  (ca / color-property-ascription :domain b :range r)
  (aa / age-property-ascription :domain b :range o)
  (r / red)
  (o / old))
```

Within this expression, `house` is a subtype of the general semantic types `object` and `three-d-location` and both `red` and `old` are subtypes of particular kinds of qualities. Linking relationships between the semantics and the grammar encode just what possibilities for grammatical

type refinement are available for each semantic type—e.g., nominal groups are available for **objects**, while modifications within nominal groups are available for **qualities**. Lexical selection is handled in the most simple case by linking semantic concepts with sets of lexical items in a language; in the event that lexical items are missing for any particular concept, lexicalization strategies of varying complexity can be employed (cf. Stede, 1995).

Considerable care was taken to make SPL input expressions as ‘grammar-free’ as possible. This means that they should support a variety of ‘paraphrases’—usually selected among by variation of the textual statuses specified (e.g., Iordanskaja et al., 1991, Kittredge, 1995). Since such paraphrases commonly rearrange the constituency structure of corresponding grammatical units, this required very early a commitment to ‘flat’ semantic representations that did not impose particular constituency dependencies on their corresponding surface forms. Thus this particular SPL is also the semantic specification, given appropriate grammars, for the German ‘das alte rote Haus’, the Czech ‘starý červený dům’, and so on. The presence of semantic information in the SPL can be motivated by very diverse linguistic phenomena when considered cross-linguistically. The **identifiability** property, for example, expresses information status. However, while this is used by the grammars of, for example English, German and French, to constrain determination options (e.g., ‘the’ *vs.* ‘a’), for Czech and Russian it plays more of a role in determining preferred word order within the containing clause (Kruijff-Korbayová et al., 2002). A particular semantic specification can therefore have widely diverging lexicogrammatical consequences.

More interesting semantic specifications are necessary for information capable of being expressed through clauses. These kinds of SPLs are event-based, following Davidson’s (1967) proposals for event semantics, and accordingly introduce variables for events that may be placed in a variety of semantic relationships, including relationships such as temporal, evaluative, circumstantial as well as a range of defined ‘thematic’ roles in the spirit of Parsons (1990). The possibilities for these are also given by the general semantic type lattice.⁹

Finally, we note here that the same kind of organizational methods are applied within the KPML system for text organizing resources such

⁹ There are interesting similarities between the SPL semantic specifications and the kinds of representations currently being promoted as semantic representations within typed feature structure based approaches, such as Minimal Recursion Semantics (Copestake et al., 1999), or within categorial approaches, such as Hybrid Logic Dependency Semantics (Baldrige and Kruijff, 2002). We are currently considering this confluence of developments for an update of the SPL specification.

as rhetorical structure and text macrostructure. This is important because it means that the techniques for multilingual representation are automatically applicable (and have been applied) to these levels of representation also. There is then no sharp discontinuity in the approach taken to multilinguality when we move away from the central linguistic areas of grammar. The paradigmatic/syntagmatic axis distinction, for example, corresponding to system networks and realization statements, and its conditionalization according to language is applied to *all levels of linguistic description*: each linguistic stratum is represented in this way, with linking rules relating selections at one structure with those at another. This fully implements our notion of intrinsic multilinguality and provides automatically for the kind of cross-linguistic variation that has been established as necessary in MLG work (cf. §2). If one language has a general preference for one kind of construction rather than another, this is captured directly by a general register-constraint conditionalized for that language and applying to a broad range of registers. Register-imposed constraints of this kind take up the task of setting default options for lexicogrammatical type refinement when not directly controlled by semantic options.

4. Discussion: development methodology and results

Based on our experiences concerning the development and re-use of linguistic resources across languages within the framework described so far, we adopt the following staged developmental methodology for resource construction. While we also exploit alternatives, for example, combining functional regions from different starting grammars, the steps described here represent a basic methodology that is known to work in all cases, although it might not always provide the best solution first.

1. Take a copy of the starting point grammar and re-conditionalize this as belonging to the target language(s). Reconditionalization is an automatic one-click process producing a single-language view for each target language.
2. Construct a corpus of target language behavior to benchmark progress during development: for grammar, this is typically a set of sentences which should be generated and corresponds to a *target suite*.¹⁰

¹⁰ The approach we use to benchmark grammar development progress is similar to the use of *test suites*, e.g., for HPSG (Open et al., 2002), although their application for generation shows up some interesting differences (Bateman and Hartley, 2000).

3. Construct appropriate semantic specifications for the elements of the target suite: these can often be taken from existing examples for English or other languages or be derived anew. The function of these specifications is to guide developers directly to necessary grammar changes, not to restrict the coverage of the grammar produced. They thus serve as minimal requirements for generation coverage, not maximal.
4. Select particular functional regions where the source and target languages are known to, or can be shown to, exhibit differing grammatical structures. The semantic specifications of (3) are also used here: whenever a result generated on the basis of these semantic specifications produces inappropriate structures in the target language, an ‘incongruence’ that needs repair has been discovered. Pinpointing the position of these incongruences is strongly supported by the development environment.
5. Distribute these functional regions across available working groups, each of which then alters its functional regions so as to produce structures conforming to the target language (examples of how this is supported concretely have been given in, for example, Bateman, 1997).
6. Perform regression testing/revision on the grammar as modified in steps (4–5) by using the semantic specifications of (3); ‘success’ is defined when the grammar generates a surface rendition of the semantics appropriate for the target language.
7. Re-import improved solutions in newly developed functional regions back into the original resources, evaluating any competing descriptions that may be found and reconciling differences through active discussion involving the developers.
8. Repeat steps (4–7) extending the target suites covered and the linguistic treatments as required.

While this developmental sequence is broadly similar to many ‘re-use’ approaches, it is made particularly effective by the function-oriented organization of the grammars and the direct support of the KPML development environment.

Some of the results of applying this strategy have been the Bulgarian, Czech and Russian resources developed within the Agile project that we have used as an example throughout the paper. In Figure 4 we show in more detail the re-use statistics for the final grammars produced by this project; these grammars covered all of the examples in the

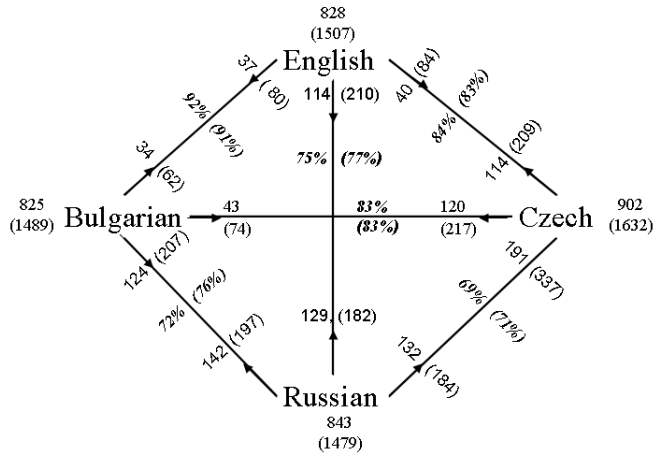


Figure 4. Indication of re-use statistics of the Bulgarian, Czech and Russian grammar networks developed within the Agile project. Numbers in parentheses are counts of features; numbers outside count systems.

respective target suites defined for the application domain. The graphic shows the pairwise overlap and difference between the grammatical networks developed with respect to both systems and their features (shown in parentheses). Systems are shown because these are the usual developmental unit with which grammar developers work in KPML, while features correspond most closely formally to the actual types of the type lattices familiar to those working in HPSG-style frameworks (cf. §3).

The diagram is read as follows. The pairs of figures shown *leaving* each language node are the numbers of systems and features that would need to be *removed* from the starting resource in order to arrive at the specification indicated in the target resource; the figures shown *arriving* at a node show the numbers of systems and features that would need to be *added*. Thus, for example, in order to convert the Bulgarian resource (with 825 systems over 1489 features in its type lattice) to the Russian resource (843 systems over 1479 features), the Bulgarian resource must lose 124 of its systems (207 features) and gain 142 new systems (197 features). We can assess the degree of re-usability of these resources by considering each language pair as a bilingual generation grammar and calculating how much of that bilingual resource is congruent, or shared. For Bulgarian and Russian, this means that 72% of the systems and 76% of the features are shared between the two resources. Although this commonality can be broken down more usefully to consider individual functional regions (as done for several other languages, for example,

in Bateman, 1997), for present purposes this overall indication of re-use will suffice.¹¹

The diagram shows that for all the derived languages, the language that they most closely resemble is English (with respective commonality of systems of 92% for Bulgarian, 84% for Czech and 75% for Russian). This is a clear symptom of the fact that were all originally derived from the English grammar and that not all regions of the grammar have been adapted fully to their target languages.¹² The individual differences between the derived languages is therefore most accurately seen as an indication of the kind of development work that has been carried out rather than a statement of contrastive linguistics. We take this as a cautionary note for premature claims of universality.

This kind of result is typical of development work with KPML following the stages set out above. Full-scale resource development generally goes through two distinct phases. The first, usually driven by the practical requirements that generation in a language should be made possible for an application (and often as quickly and cheaply as possible), corresponds to stages (1–6). This produces generation grammars that adequately cover the target suites of the target applications but which might not be considered optimal treatments from the perspective of the languages covered in their own terms—i.e., they may still resemble the modified starting grammar too much. The second phase of development occurs for a language when there is more time, or money, or interest to carry out detailed grammar work in which functional regions are modified to reflect analyses appropriate when seen from the perspective of that language itself.

Of the three grammars in Agile, only a few functional regions have moved into this second phase of development. This can be contrasted with the state of affairs in the Spanish grammars under development with KPML. The original grammar was produced by transfer comparison from an English grammar in the normal way and rapidly provided Spanish generation for a knowledge-based system in a chemistry domain (Aguado et al., 1998). The current grammar is part of a project investigating the application of generation grammars in language teaching and involves a thorough re-design in linguistic terms motivated entirely from within a treatment of Spanish. The comparative re-use

¹¹ Note that there are also other ways of assessing the re-use statistics. For example, we could ask how many systems/features are re-used by a *monolingual* description: for example, how many of the systems of the Bulgarian resource are provided by the Russian resource. Expressing this as a percentage gives much higher re-use figures, all in the range 80-95% for the language pairs of Figure 4.

¹² Prior to the AGILE project, there were no slavic KPML-compatible grammars available.

Table II. Re-use statistics for two Spanish grammars

version	systems	re-use percentage	features	re-use percentage
1	769	90%	1358	92%
2	1066	32%	1818	38%

statistics of the two grammars with respect to the English grammar are shown in Table II

Although the current Spanish grammar is certainly a more adequate treatment of Spanish (and in fact represents the largest such resource for Spanish that is available), the commonality percentage can still not be considered an unambiguous indication of linguistic diversity. The differences are made up both of genuine linguistic differences and of differences in linguistic treatments. Moreover, stage (7) of the developmental methodology, re-importing improved solutions into the original resources, has not yet been carried out and it is quite likely that changes to the English grammar might also serve to again *increase* commonality. Such re-importing has been done for the Dutch and German grammars for example, and is scheduled for the Czech grammar (particularly with respect to the treatment of the textual control of word order: Kruijff-Korbayová et al., 2002).

A further significant result of this development work is to show that there is very little sense in which one can talk of a common shared area of representation that is re-used for all languages. The re-use profile across language pairs involves different groups of functional regions depending on how similar or different those languages are with respect to those particular grammatical areas. Distinct pairs of languages can resemble each other and diverge in different ways. One of the strengths of our framework is that descriptions are not driven back to a minimal common ‘core’ or kernel that is shared but is instead free to describe similarities and differences individually for each language pair or group of languages. The need for such flexibility is now thrown into relief by typological research where it has been established that traditional typological dimensions of variation are less revealing than might have been hoped concerning individual language behaviour; as Stolz and Urdze (2001) report:

“As a matter of fact, our experience as typologists teaches us that typological homogeneity of languages, more often than not, is a theoretical construct, rather than an empirical fact.” (p281)

The granularity of multilinguality that we pursue is intended to resist imposing such homogeneity when it is not motivated.

The kind of multilingual mechanism provided by KPML also presents interesting issues for multilingual representation. Whereas the representations of the linguistic information maintained for the basic dimensions of the theory (strata, axes, ranks: cf. §3) are relatively unproblematic and can already be related confidently to those of other accounts, the additional requirement that language conditionalization be present suggests a variety of treatments, not all of which are equally effective. Staying strictly within a pure inheritance lattice without extra mechanisms is not, for example, possible. This could only be achieved by adding into the lattice types for the individual language varieties to be covered. These types would then need to be conjoined into the feature structure descriptions for the particular languages those structures are to hold for. This has two significant drawbacks. First, it complicates immensely the type lattice; this problem with unified type lattices over several languages was already noted for the rather more simple multilingual examples first discussed in the ground-breaking description offered by Kameyama (1988). And second, it stands in the way of providing monolingual generalizations because well-typedness constraints will force the paradigmatic descriptions of the type lattice (the ‘features’ in the SFL account) to always be inline with the syntagmatic representation maintained by the feature structures (the ‘functional structure’ in SFL). This latter consideration is particularly damaging for multilingual development because, as our results show, it is primarily in the *decoupling* of the paradigmatic and syntagmatic descriptions that very significant re-use is found.

For these and other reasons, we maintain the multilingual information concerning our resources as an *extra orthogonal layer* of representation that does not impact on the individual type lattices maintained for the individual grammar representations. It is therefore easier to ensure that the individual type lattices remain consistent because they may each be type checked for consistency individually. Diverging language descriptions cannot then compromise each other. This architecture also enables KPML to support free extraction of monolingual to N-lingual multilingual grammar specifications from general multilingual resources as well as the merging of monolingual to N-lingual descriptions in to a single resource because as it is known that each individual resource is well-formed and combinations of resources are not combined at the feature structure level. The extent to which specifications are comparable is left to the individual contributions and their developers rather than being enforced. This means that grammars are not forced into the mould of any single language.

To conclude, we have argued that our approach to multilinguality, including both the theoretical framework and its practical implementation, has followed directly from our commitment to systemic-functional typology, which is primarily concerned with **functional typology** rather than structural typology. Within systemic-functional linguistics there is an expectation that functional typology should draw out more commonalities across languages than a structural typology—we have explored this hypothesis concretely and practically in the context of natural language generation and so far the hypothesis has been strongly supported. This shows that we can extend re-use methods to substantially widen the range of languages for which porting, in some form or other, is an effective technique. We have concentrated on the relation of the layers of description to the phenomenon of multilinguality throughout; this reflects the basic position that we adopt to multilinguality, which we have termed **intrinsic multilinguality**. This is, in short, the position that *all* linguistic description must inherently concern itself with multilingual description—and not, as is more common, to consider this as an extra, extrinsic requirement.

Acknowledgements

Considerable thanks are due to the editors and anonymous reviewers for clarifying the discussion and issues that needed to be raised. We also thank current and former colleagues who have contributed to the grammar development activities we report here. Errors and inaccuracies remain as always our own.

References

- Aguado, G., A. Bañón, J. A. Bateman, S. Bernardos, M. Fernández, A. Gómez-Pérez, E. Nieto, A. Olalla, R. Plaza, and A. Sánchez: 1998, 'ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation'. In: *Proceedings of the ECAI'98 Workshop on Applications of Ontologies and Problem Solving Methods*. Brighton, U.K., European Conference on Artificial Intelligence.
- Alshawi, H., D. Carter, B. Gambäck, and M. Rayner: 1992, 'Swedish-English QLF translation'. In: H. Alshawi (ed.): *The Core Language Engine*. MIT Press, pp. 277–319.
- Baader, F., D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider (eds.): 2003, *The Description Logic Handbook*. Cambridge University Press.
- Baldrige, J. and G.-J. Kruijff: 2002, 'Coupling CCG and Hybrid Logic Dependency Semantics'. In: *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, pp. 319–326.
- Bateman, J. A.: 1990, 'Upper Modeling: organizing knowledge for natural language processing'. In: *Proceedings of the Fifth International Natural Language*

- Generation Workshop*. Pittsburgh, PA., pp. 54–60. Organized by Kathleen R. McKeown (Columbia University), Johanna D. Moore (University of Pittsburgh) and Sergei Nirenburg (Carnegie Mellon University). Held 3-6 June 1990, Dawson, PA.
- Bateman, J. A.: 1992, ‘Towards Meaning-Based Machine Translation: using abstractions from text generation for preserving meaning’. *Machine Translation* **6**(1), 1–37. (Special edition on the role of text generation in MT).
- 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., M. Emele, and S. Momma: 1992, ‘The nondirectional representation of Systemic Functional Grammars and Semantics as Typed Feature Structures’. In: *Proceedings of COLING-92*, Vol. III. Nantes, France, pp. 916–920.
- Bateman, J. A. and A. Hartley: 2000, ‘Target suites for evaluating the coverage of text generators’. In: M. Gavrilidou, G. Carayannis, S. Markantonatou, S. Piperidis, and G. Stainhaouer (eds.): *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC’2000)*. Athens, Greece, European Language Resources Association (ELRA).
- Bateman, J. A., R. Henschel, and F. Rinaldi: 1995, ‘Generalized Upper Model 2.0: documentation’. Technical report, GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany.
- Bateman, J. A., C. M. I. M. Matthiessen, K. Nanri, and L. Zeng: 1991, ‘The re-use of linguistic resources across languages in multilingual generation components’. In: *Proceedings of the 1991 International Joint Conference on Artificial Intelligence, Sydney, Australia*, Vol. 2. pp. 966–971, Morgan Kaufmann Publishers.
- Bateman, J. A., C. M. I. M. Matthiessen, and L. Zeng: 1999, ‘Multilingual natural language generation for multilingual software: a functional linguistic approach’. *Applied Artificial Intelligence* **13**(6), 607–639.
- Bender, E. M., D. Flickinger, and S. Oepen: 2002, ‘The Grammar Matrix: an open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars’. In: *Proceedings of the COLING 2002 Workshop on Grammar Engineering and Evaluation*. pp. 8–14.
- Bresnan, J.: 1982, ‘Control and computation’. In: J. Bresnan (ed.): *The Mental Representation of Grammatical Relations*. Cambridge, Massachusetts: M.I.T. Press, Chapt. 5.
- Brew, C.: 1991, ‘Systemic Classification and its Efficiency’. *Computational Linguistics* **17**(4), 375–408.
- Butt, M., T. H. King, M.-E. Niño, and F. Segond: 1999, *A Grammar Writer’s Cookbook*. Stanford: CSLI.
- Cahill, L., C. Doran, R. Evans, R. Kibble, C. Mellish, D. Paiva, M. Reape, D. Scott, and N. Tipper: 2000, ‘Enabling resource sharing in language generation: an abstract reference architecture’. In: *Proceedings of the 2nd International Conference on Language Resources and Evaluation*. Athens, Greece.
- Contant, C.: 1986, ‘Génération automatique de texte: Application au sous-langage boursier’. Master’s thesis, D’ept. de Linguistique, Université de Montréal.
- Copestake, A.: 2002, *Implementing Typed Feature Structure Grammars*. Stanford, CA: CSLI Publications.

- Copestake, A., D. Flickinger, I. Sag, and C. Pollard: 1999, 'Minimal Recursion Semantics: An Introduction'. <<http://www-csli.stanford.edu/aac/papers/newmrs.ps>>.
- Danlos, L.: 1987, *The linguistic basis of text generation*, Studies in Natural Language Processing. Cambridge, England: Cambridge University Press.
- Davidson, D.: 1967, 'The logical form of action sentences'. In: N. Rescher (ed.): *The logic of decision and action*. Pittsburgh, Pennsylvania: University of Pittsburgh Press, pp. 81–95.
- Delin, J., A. Hartley, C. L. Paris, D. Scott, and K. V. Linden: 1994, 'Expressing Procedural Relationships in Multilingual Instructions'. In: *Proceedings of the Seventh International Workshop on Natural Language Generation, Kennebunkport, Maine, USA, June 21-24, 1994*. Kennebunkport, Maine, USA, pp. 61–70.
- Dorr, B. J.: 1994, 'Machine translation divergences: a formal description and proposed solution'. *Computational Linguistics* **20**(4), 597–634.
- Goldman, N.: 1975, 'Conceptual Generation'. In: R. C. Schank (ed.): *Conceptual Information Processing*. Amsterdam: North-Holland Publishing Co.
- Halliday, M. A. K.: 1966, 'Some notes on 'deep' grammar'. *Journal of Linguistics* **2**(1), 57–67.
- Halliday, M. A. K. and C. M. Matthiessen: 2004, *An Introduction to Functional Grammar*. London: Edward Arnold, 3rd. edition.
- Halliday, M. A. K. and C. M. I. M. Matthiessen: 1999, *Construing experience through meaning: a language-based approach to cognition*. London: Cassell.
- Hartley, A. and C. Paris: 1997, 'Multilingual document production: from support for translating to support for authoring'. *Machine Translation* **12**(1–2), 109–129.
- Henschel, R.: 1997, 'Compiling systemic grammar into feature logic systems'. In: S. Manandhar, W. Nutt, and G. P. Lopez (eds.): *CLNLP/NLULP Proceedings*.
- Henschel, R. and J. A. Bateman: 1994, 'The merged upper model: a linguistic ontology for German and English'. In: *Proceedings of COLING '94*. Kyoto, Japan.
- Iordanskaja, L. N., R. Kittredge, and A. Polguère: 1991, 'Lexical Selection and Paraphrase in a Meaning-Text Generation Model'. In: C. L. Paris, W. R. Swartout, and W. C. Mann (eds.): *Natural language generation in artificial intelligence and computational linguistics*. Kluwer Academic Publishers, pp. 293–312.
- Jacob, D. and E. Maier: 1988, 'Die Übertragung des MUMBLE-Generators für die Generierung von Deutsch'. In: H. Trost (ed.): *4. Österreichische Artificial-Intelligence-Tagung Proceedings*, No. 176 in Informatik-Fachberichte. Springer.
- Kameyama, M.: 1988, 'Atomization in grammar sharing'. In: *Proceedings of the 26th. Annual Meeting of the Association for Computational Linguistics*. Buffalo, New York, pp. 194–203, Association for Computational Linguistics.
- Kasper, R. T.: 1988, 'Systemic Grammar and Functional Unification Grammar'. In: J. D. Benson and W. S. Greaves (eds.): *Systemic Functional Approaches to Discourse*. Norwood, New Jersey: Ablex, pp. 176–199. Also available as USC/Information Sciences Institute, Reprint Report ISI/RS-87-179, 1987.
- Kasper, R. T.: 1989, 'A flexible interface for linking applications to PENMAN's sentence generator'. In: *Proceedings of the DARPA Workshop on Speech and Natural Language*.
- Kehl, W.: 1989, 'GEOTEX-E: Generierung zweisprachiger Konstruktionstexte'. In: B. Endres-Niggemeyer (ed.): *Interaktion und Kommunikation mit dem Com-*

- puter, No. 238 in Informatik-Fachberichte. Springer. (GLDV-Jahrestagung Proceedings).
- Kim, R., M. Dalrymple, R. M. Kaplan, T. H. King, H. Masuichi, and T. Ohkuma: 2003, 'Multilingual grammar development via grammar porting'. In: E. Bender, D. Flickinger, F. Fouvry, and M. Siegel (eds.): *Proceedings of the ESSLLI Workshop on Ideas and Strategies for Multilingual Grammar Development*. Vienna, Austria, pp. 49–56.
- Kittredge, R.: 1995, 'Efficiency vs. Generality in Interlingual Design: some linguistic considerations'. In: R. Kittredge (ed.): *Proceedings of the IJCAI '95 Workshop on Multilingual Text Generation*. Montréal, Québec, pp. 64–74, AAAI.
- Kruijff, G.-J., E. Teich, J. A. Bateman, I. Kruijff-Korbayová, H. Skoumalová, S. Sharoff, L. Sokolova, T. Hartley, K. Staykova, and J. Hana: 2000, 'A multilingual system for text generation in three Slavic languages'. In: *Proceedings of the 18th. International Conference on Computational Linguistics (COLING'2000)*. Saarbrücken, Germany, pp. 474–480.
- Kruijff-Korbayová, I., G.-J. M. Kruijff, and J. A. Bateman: 2002, 'Generation of contextually appropriate word order'. In: K. van Deemter and R. Kibble (eds.): *Information sharing*. CSLI, pp. 193–222.
- Kukich, K.: 1983, 'Design of a Knowledge-Based Report Generator'. In: *Proceedings of the 21st Annual Conference*.
- Langkilde, I. and K. Knight: 1998, 'Generation that Exploits Corpus-based Statistical Knowledge'. In: *Proceedings of the ACL/COLING-98*. Montreal, Quebec.
- Lee, J.-H., A. Okumura, K. Muraki, and G. C. Kim: 1991, 'An English-Korean machine translation system: Korean synthesis under the environment of Japanese generation system'. In: *Proceedings of 1991 Japan-Australia Joint Symposium on Natural Language Processing*. Fukuoka, Japan, pp. 219–224.
- Mann, W. C. and C. M. I. M. Matthiessen: 1985, 'Demonstration of the Nigel Text Generation Computer Program'. In: J. D. Benson and W. S. Greaves (eds.): *Systemic Perspectives on Discourse, Volume 1*. Norwood, New Jersey: Ablex, pp. 50–83.
- Mann, W. C. and S. A. Thompson: 1988, 'Rhetorical Structure Theory: Toward a Functional Theory of Text Organization'. *Text* **8**(3), 243–281.
- Matthiessen, C. M. I. M.: 1995, *Lexicogrammatical cartography: English systems*. Tokyo, Taipei and Dallas: International Language Science Publishers.
- Matthiessen, C. M. I. M. and J. A. Bateman: 1991, *Text generation and systemic-functional linguistics: experiences from English and Japanese*. London and New York: Frances Pinter Publishers and St. Martin's Press.
- Matthiessen, C. M. I. M., K. Nanri, and L. Zeng: 1991, 'Multilingual resources in text generation: ideational focus'. In: *Proceedings of the 2nd Japan-Australia Joint Symposium on Natural Language Processing*. Kyushu, Japan, Kyushu Institute of Technology.
- Maybury, M. T.: 1989, 'GENNY: a knowledge-based text generation system'. *International Journal of Information Processing and Management* **25**(2), 137–150.
- McDonald, D. D.: 2000, 'Natural Language Generation'. In: R. Dale, H. Moisl, and H. Somers (eds.): *A handbook of natural language processing: techniques and applications for the processing of language as text*. New York: Marcel Dekker, pp. 147–179.
- Mellish, C. S.: 1988, 'Implementing Systemic Classification by Unification'. *Journal of Computational Linguistics* **14**(1), 40–51.

- Novello, A. and C. Callaway: 2003, 'Porting to an Italian Surface Realizer: A Case Study'. In: *Proceedings of the 9th European Workshop on Natural Language Generation*. Budapest, Hungary.
- Oepen, S., E. M. Bender, U. Callmeier, D. Flickinger, and M. Siegel: 2002, 'Parallel distributed grammar engineering for practical applications'. In: *Proceedings of the Workshop on Grammar Engineering and Evaluation, COLING 2002*. Taipei, Taiwan.
- Parsons, T.: 1990, *Events in the semantics of English: a study in subatomic semantics*. Cambridge, MA / London: MIT Press.
- Peters, K., H. Rutz, and M. Siegel: 1991, 'KLEIST: Textgenerierung in deutscher und japanischer Sprache'. Technical Report KOLIBRI Arbeitsbericht Nr. 26, Fakultät für Linguistik und Literaturwissenschaft der Universität Bielefeld. (DFG-Forschungsgruppe Kohärenz).
- Rambow, O.: 1988, 'Teaching a second language to a computer: a programmer's view'. In: H. Trost (ed.): *4. Österreichische Artificial-Intelligence-Tagung Proceedings*, No. 176 in Informatik Fachberichte. Heidelberg: Springer.
- Rayner, M., D. Carter, and P. Bouillon: 1996, 'Adapting the Core Language Engine to French and Spanish'. In: *Proceedings of NLP-IA-96*. Moncton, new Brunswick.
- Rösner, D.: 1987, 'The automated news agency: the SEMTEX generator for German'. In: G. Kempen (ed.): *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*. Boston/Dordrecht: Kluwer Academic Publishers, pp. 133–148. Paper presented at the Third International Workshop on Natural Language Generation, August 1986, Nijmegen, The Netherlands.
- Rösner, D.: 1994, *Automatische Generierung von mehrsprachigen Instruktionstexten aus einer Wissensbasis*. Stuttgart, Germany: Fakultät für Informatik, Universität Stuttgart. Habilitationsschrift.
- Rösner, D. and M. Stede: 1992, 'Customizing RST for the automatic production of technical manuals'. In: R. Dale, E. H. Hovy, D. Rösner, and O. Stock (eds.): *Aspects of automated natural language generation*. Springer, pp. 199–214. (Proceedings of the 6th International Workshop on Natural Language Generation, Trento, Italy, April 1992).
- Rösner, D. and M. Stede: 1994, 'Generating multilingual documents from a knowledge base: the TECHDOC project'. In: *Proceedings of the 15th. International Conference on Computational Linguistics (COLING 94)*, Vol. I. Kyoto, Japan, pp. 339–346.
- Smith, M. K., C. Welty, and D. L. McGuinness: 2004, 'OWL Web ontology language guide'. Technical Report 20040210, World Wide Web Consortium.
- Stede, M.: 1995, 'Lexicalization in Natural Language Generation: a survey'. *Artificial Intelligence Review* 8, 309–336.
- Stolz, T. and A. Urdze: 2001, 'Head-marking vs. dependent-marking in modern Latvian'. *Sprachtypologie und Universalienforschung (STUF)* 54(3), 279–297.
- Vander Linden, K.: 2000, 'Natural Language Generation'. In: D. Jurafsky and J. Martin (eds.): *Speech and Language Processing: an introduction to speech recognition, computational linguistics and natural language processing*. New Jersey: Prentice-Hall, Chapt. 20, pp. 763–798.
- Vander Linden, K. and D. Scott: 1995, 'Raising the Interlingual Ceiling with Multilingual Text Generation'. In: *Proceedings of the IJCAI workshop in multilingual text generation (International Joint Conference on Artificial Intelligence) 1995*. Montréal, Canada, pp. 95–101.