# AGILE

*Automatic Generation of Instructions in Languages of Eastern Europe*

---

Title ***Grammatical resource implementation for Bulgarian, Czech and Russian***

Authors John Bateman
Elke Teich
Danail Dochev
Nevenna Gromova
Jiri Hana
Ivana Kruijff-Korbayova
Hana Skoumalova
Sergei Sharoff
Elena Sokolova

Deliverable *IMPL1-Bu, IMPL1-Cz, IMPL1-Ru*

Status *Final*

Availability *Public*

Date *June 1998*

---

INCO COPERNICUS PL961104

## Abstract:

This document describes the Deliverables IMPL1-BAS, IMPL1-CU and IMPL1-RU of Work Package 7, task 7.1 of the AGILE project. We present the theoretical and methodological underpinnings of the approach taken and introduce the grammar development system that is used for grammar implementation in AGILE, the Komet-Penman MultiLingual system (KPML). This is followed by descriptions of the implementations of lexico-grammatical resources for Bulgarian, Czech and Russian. We conclude with a summary and an outlook on the next phase of implementation in this Work Package (task 7.2).

More information on AGILE is available on the project web page and from the project coordinators:

| | |
|---|---|
| URL: | http://www.itri.brighton.ac.uk/projects/agile |
| email: | agile-coord@itri.bton.ac.uk |
| telephone: | +44-1273-642900 |
| f | +44-1273-642908 |

## Table of Content

1. Overview of this document.....1

2. Tactical generation in AGILE.....1

2.1 Theoretical foundations: Systemic Functional Linguistics (SFL).....1

2.2 SFG in Natural Language Generation.....3

3. The methodology of multilingual generation and multilingual resource development adopted in AGILE.....8

3.1 Resource sharing: contrastive grammar.....8

3.2 Multilingual resource development in KPML.....10

4. Grammatical resource implementation for Slavonic languages: Bulgarian, Czech and Russian.....11

4.1 Bulgarian.....12

4.2 Czech.....17

4.2 Russian.....23

5. Summary and outlook.....27

References.....28

Appendix

I KPML references

II Sample texts for Initial Demonstrator English, Bulgarian, Czech and Russian

III System network specifications for Bulgarian, Czech and Russian

IV Input specifications (SPL) for sentences of sample texts for Bulgarian, Czech and Russian

## Table of Figures

Figure 1: Organizing principles of representation in SFG.....3

Figure 2: Input to generation with KPML.....5

Figure 3: KPML architecture.....6

Figure 4: Output structures (clause level) for SPL-1 and SPL-2.....7

Figure 5: Dimensions of multilingual description.....9

Figure 6: Graph of a system network having used the INSPECT option.....10

Figure 7: Notational conventions in Systemic Functional Grammar.....11

Figure 8: Syntax of computational system network specifications.....12

# 1. Overview of this document

This document describes the Deliverables IMPL1-BAS, IMPL1-CU and IMPL1-RU in the AGILE (Automatic Generation of Instructions in Languages of Eastern Europe) project (Agile, 1997), which are the implementations of lexical and grammatical resources for tactical generation for Bulgarian, Czech and Russian. We start out with a description of the platform chosen for the implementation of lexico-grammatical resources, the **K**omet-**P**enman **M**ulti**L**ingual system (KPML) (Bateman, 1997a) and present the theoretical basis of this framework for generation (Section 2) and for multilingual resource development (Section 3). We then describe the implementations of grammar fragments of Bulgarian, Czech and Russian for the initial demonstrator (Section 4), the goal of which was the generation of the sentences of the sample texts given in Appendix II. We conclude with a summary and sketching the strategy for the next round of implementation (Section 5).

# 2. Tactical generation in AGILE

The platform for implementation of the grammatical resources of Bulgarian, Czech and Russian chosen in AGILE is the Komet-Penman MultiLingual (KPML) system (Bateman, 1997a; Bateman, 1997b), which is based on the Penman system for generating English (Mann, 1993; Mann and Matthiessen, 1983; Penman, 1989).

In Section 2.1, we briefly describe the theoretical basis of KPML, Systemic Functional Linguistics, and in Section 2.2 we describe KPML as a generator. This is followed by a presentation of KPML's multilingual design and its use as a workbench for developing grammatical resources (Section 3).

## 2.1 Theoretical foundations: Systemic Functional Linguistics (SFL)

The linguistic-theoretical basis of the KPML system is Systemic Functional Linguistics (SFL). SFL is a British school of linguistic thought, belonging to the tradition of functional approaches to language (Hjelmslev, 1943; Dik, 1978; Halliday, 1973, Halliday, 1985) and showing affinities with the Continental-European Prague School (Firbas, 1966; Danes, 1974; Sgall *et al*., 1986).

SFL is characterized by the notions of function and system. SFL is functional in that it acknowledges three broad functions languages have: the ideational, the interpersonal and the textual (see below). SFL is systemic in that the main focus in description is on the grammatical paradigm (or: system). The kernel of SFL is Systemic Functional Grammar (SFG). The grammar of a language is represented as a *system network*, which can be read as a declarative statement of grammatical features and the co-occurrence constraints holding between them. Systemic Functional Grammar is thus a classification-based approach to grammar, rather than a rule-based one---very similar to other models of grammar currently used in computational linguistics, such as Head-Driven Phrase Structure Grammar (Pollard and Sag, 1987; Pollard and Sag, 1992), in which a classification hierarchy of grammatical (and lexical) types constitutes the grammatical description, where subtypes inherit the constraints of their supertypes.[1] What makes SFG stand out from such approaches is the functional motivation of grammatical types.

---

[1] The type-subtype relation is called *delicacy* in SFG.

The grammatical types are functionally motivated in the following way.

The notion of function in SFL is predominantly manifested in the concept of *metafunctions*, a set of generalized functions that language is said to fulfil. The ideational metafunction encodes a language's propositional content. Its grammatical aspect is notably reflected in the clause in the system of transitivity, which gives rise to configurations of processes and the participants and circumstances involved, such as Actor, Goal, Spatial Adjunct, Temporal Adjunct etc. The interpersonal metafunction encodes speakers' roles in an interaction, their attitudes and evaluations. One of the major grammatical reflexes is the clause system of mood, which distinguishes between declarative, interrogative and imperative and accounts for the differences in syntactic structure that these different moods come along with. The textual metafunction encodes properties of textual organization, such as global text structure, coherence and cohesion. In the grammar, this is reflected in systems of theme-rheme patterning and information structuring at clause level and determination at the level of the nominal group. The functionally motivated systems and their features are associated with *realization statements*. Realization statements are the attributes of a functional grammatical class and specify the syntagmatic, surface-syntactic constraints that the functional classes exhibit. For instance, a surface-syntactic constraint associated with the functional class declarative of finite clauses in English is that in syntactic structure the Subject is ordered before the Finite verb. This distinction between paradigmatic, functionally motivated classes and syntagmatic structure is referred to as *axiality* : a linguistic description in SFG always has these two aspects that are linked by the relation of realization.

Another organizing principle of linguistic representation in SFG is *rank*. Rank is implemented as the top system in the grammatical classification and distinguishes between clauses, nominal groups, prepositional phrases, adjectival and adverbial groups, words and sometimes morphemes. This *rank scale* gives the basic paradigmatic grammatical classes for which particular sets of systems and their features hold. In the overall model of SFL, the grammar constitutes the lexico-grammatical *stratum*, where the other, more abstract strata are semantics and context. The strata are in a relation of inter-stratal realization, where contextual categories are said to be realized by semantic categories and these, in turn, are realized by lexico-grammatical categories. See Figure 1 for a graphical summary of the organizing principles of SFL.[2] In the following section we relate the properties of SFG to the tasks involved in tactical generation and describe the process of tactical generation with an SFG as implemented in the Penman-based generator KPML.

[2] For a fairly concise introduction to Systemic Functional Grammar see (Bateman, 1992). More comprehensive accounts of the theory can be found in (Halliday, 1978; Halliday and Matthiessen, to appear).

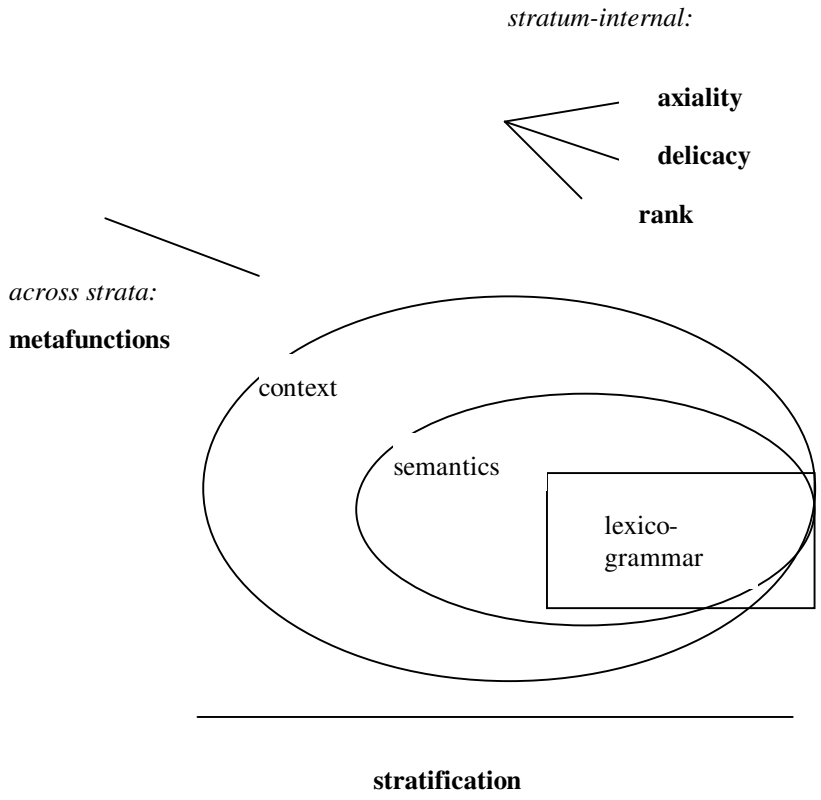



Figure 1: Organizing principles of representation in SFG

## 2.2 SFG in Natural Language Generation

Adopting a systemic functional model as a linguistic basis for generation generally supports the modeling of tasks involved in generation because many of the characteristics of the systemic functional model are suggestive of the kinds of resources needed in Natural Language (NL) Generation. This is evidenced by the fact that many generation systems are based on or have been inspired by SFL (e.g., Penman (Penman, 1989), KPML (Bateman, 1997a), COMMUNAL (Fawcett and Tucker, 1989), Multex (Matthiessen *et al.*, 1995; Matthiessen *et al.*, 1998), Comet (McKeown *et al.*, 1990), Surge (Elhadad, 1990; Elhadad and Robin, 1996)) and many projects have used the SFG-based tactical generators Penman and KPML, e.g., the German KOMET project (Bateman *et al.*, 1991a; Bateman and Teich, 1995; Teich *et al.*, 1996; Bateman *et al.*, 1998), the German TECHDOC project (Roesner and Stede, 1992; Roesner and Stede, 1994), the British DRAFTER project (Paris *et al.*, 1995; Hartley and Paris, 1997) and the Canadian Healthdoc project (DiMarco *et al.*, 1995; Wanner and Hovy, 1996).

For describing how SFG-based generation works in Penman-style generators, let us briefly recall the tasks involved in NL generation and then see how these tasks are fulfilled by an SFG-based generator.

The tasks involved in Natural Language Generation can be briefly described as follows. Generally, for generation what needs to be modeled is the relation of a communicative situation (what is talked about, when and how) to a linguistic utterance (a text, a sentence) and the processes and resources involved in getting from the former to the latter. Traditionally, the tasks involved in generation are categorized as strategic and tactical (Thompson, 1977). The tasks involved in strategic generation are to do with the determination of the propositional content to be expressed in a text and its organization into a text plan. The tasks involved in tactical generation are to do with breaking this text plan up into units that are grammatically realizable, such as clauses, nominal groups etc.

We can subdivide the tasks involved in tactical generation into two types (cf. (Teich, in press)):

- Task 1: To interpret a semantic input expression in terms of the grammar available
- Task 2: To spell out the syntactic constraints of the language in which an utterance is to be generated.

(Yang *et al.*, 1991) characterize task 1 as involving "deciphering the goals given by the speaker program and determining how they can be realized in language" (Yang *et al.*, 1991, p.207) and task 2 as ensuring "that the end product adheres to the syntactic rules of the language" (Yang *et al.*, 1991, p.207). The first is the functional aspect of tactical generation, the latter is the syntactic aspect.

For an SFG-based generator like KPML, the 'generation question' can then be formulated as follows: When to choose which feature from the grammatical system network (together with its associated syntactic constraints in the form of realization statements) according to a semantic input expression?

The tactical generation process starts with an input expression in the form of SPL (Sentence Planning Language) (Kasper, 1989). See Figure 2 showing two such SPL expressions for the sample sentences

(1) *The user chooses the PLINE command.*

and

(2) *Choose the PLINE command.*

The SPL expressions (henceforth: SPL) are instances of the Upper Model (Bateman *et al.*, 1990; Henschel and Bateman, 1994), which is the resort of ideational (i.e., propositional) meaning in KPML. Besides the ideational information expressed in an SPL, also interpersonal information (e.g., `:speechact command`) and textual information (e.g., `:identifiability-q identifiable`, `:theme o1`) is contained. The non-ideational kinds of information typically come from text planning.[3]

[3] Three kinds of keywords are defined for SPL: Upper Model or domain concepts (e.g., `choose`, `actor`), inquiries (e.g., `:identifiability-q identifiable`) and macros, which consist of a set of inquiries (e.g., `:speechact command`, `:theme o1`).

input: (SPL-1) *The user chooses the PLINE command.*

```
(p / choose
    :speechact assertion
    :theme o1
    :actor (o1 / user
                :identifiability-q identifiable)
    :actee (o2 / command
                :identifiability-q identifiable
                :class-ascription (c / software-command
                                      :name PLINE)))
```

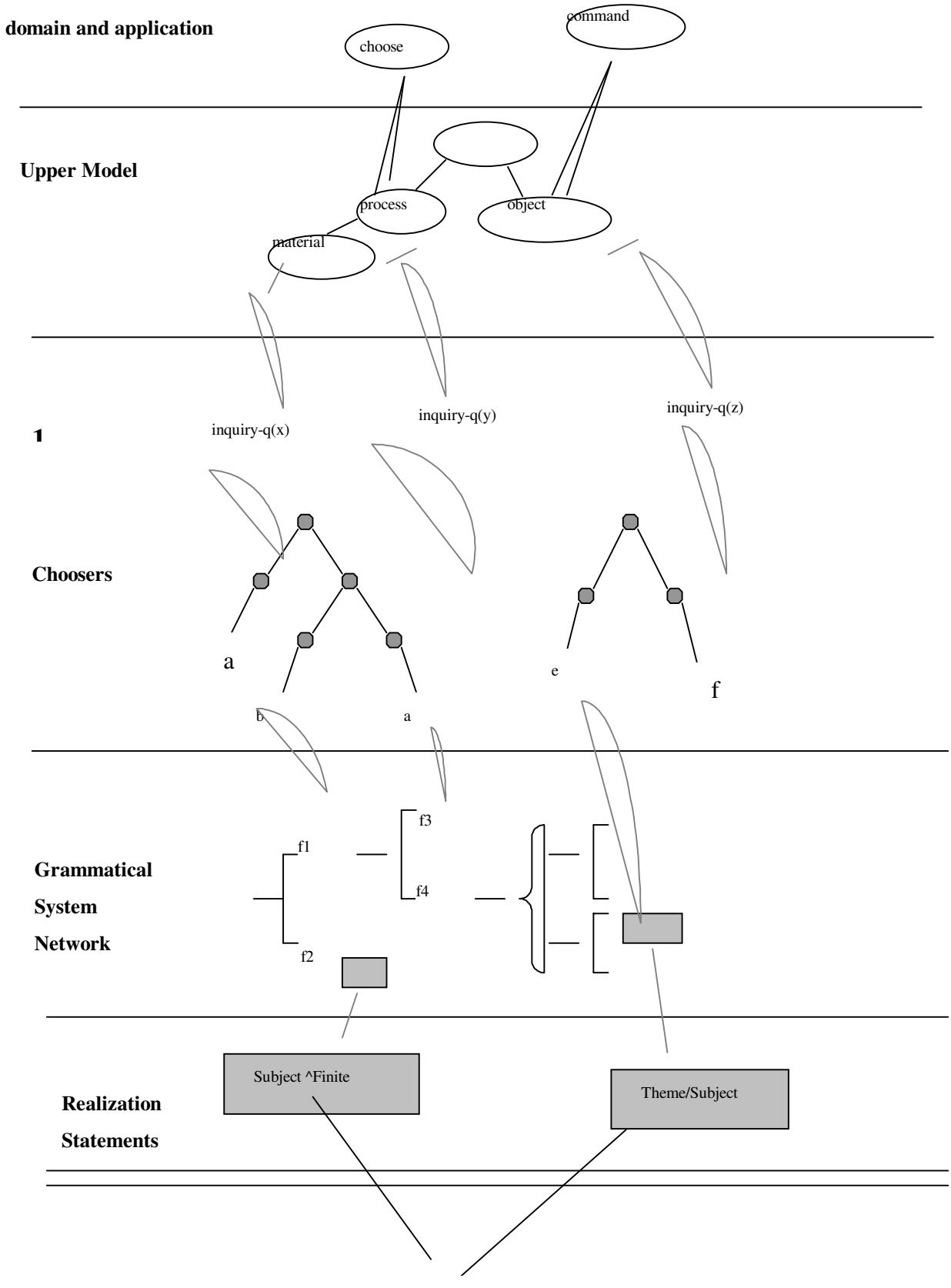
input: (SPL-2) *Choose the PLINE command.*

```
(p / choose
    :speechact command
    :actee (o2 / command
                :identifiability-q identifiable
                :class-ascription (c / software-command
                :name PLINE)))
```

Figure 2: Input to generation with KPML

Given an input such as (SPL-1) or (SPL-2) (Figure 2), the traversal of the grammatical system network is started. For English, this is the NIGEL grammar (Mann and Matthiessen, 1983). At each choice point (each system), a *chooser* is invoked that poses its inquiries to the semantics (i.e., it checks the input representation for the grounds to make a choice). A chooser is a decision procedure that is associated with a system (Matthiessen, 1988). Its task is to mediate between semantic and grammatical information. A chooser is organized as a tree, the nodes of which are *inquiries*, which are the actual interpreters of semantic knowledge for the grammar. The process of choosing is applied throughout the traversal of the system network, invoking the chooser of each system, and realization rules successively build up syntagmatic structure at all ranks. For a graphical overview of this process see Figure 3;[4] the results of generation of (SPL-1) and (SPL-2) are shown in Figure 4.

[4] The realization statement 'Subject ^ Finite' means 'order Subject before Finite' 'Theme/Subject' stands for conflation of Theme and Subject into one constituent. --- For a full list of notational conventions in SFG see Figure 7 in Section 4.


domain and application
command
choose
Upper Model
process
object
material
inquiry-q(x)
inquiry-q(y)
inquiry-q(z)
1
Choosers
a
e
f
b
a
Grammatical
System
Network
f1
f3
f4
f2
Subject ^Finite
Theme/Subject
Realization
Statements

”strings”

Figure 3: KPML architecture

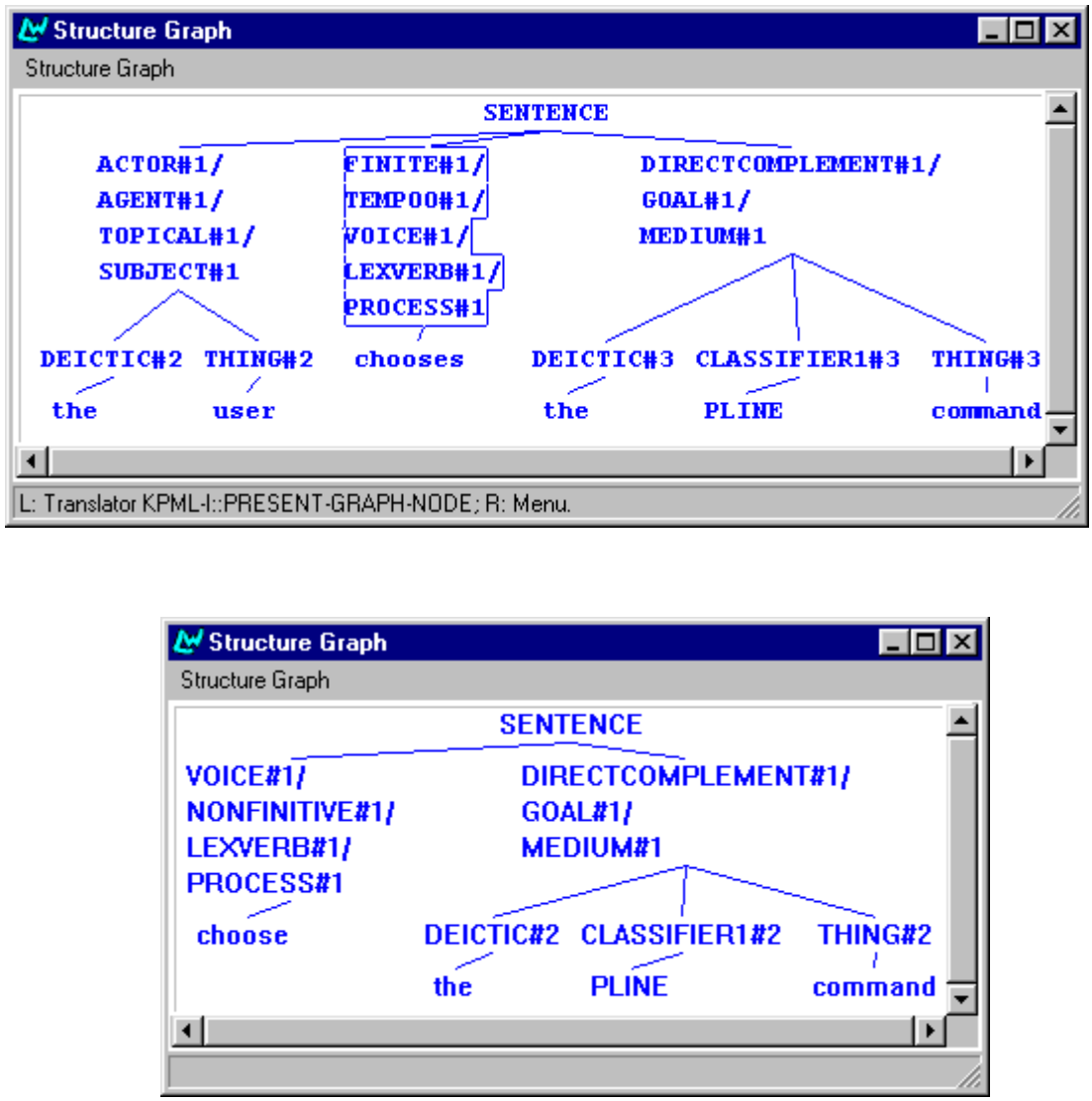



Figure 4: Output structures (clause level) for SPL-1 and SPL-2

For a systemically-based approach to tactical generation such as the one applied in KPML, task 1 of tactical generation---the grammatical interpretation of the information contained in an input semantic representation---means choice of a set of paradigmatic

features from the grammatical system network; and task 2 of tactical generation---building up a syntactic structure---means spelling out the realization statements that are associated with the paradigmatic features.[5]

# 3. The methodology of multilingual generation and multilingual resource development adopted in AGILE

In the present section we describe the model of multilinguality underlying the KPML system. We present a set of dimensions of multilingual descriptions that reflect in which respects languages can differ or be the same (Section 3.1). We then introduce the actual KPML system as an environment for the development of grammatical resources (Section 3.2).

## 3.1 Resource sharing: contrastive grammar

The representation of multilingualiy in KPML is based on the observation that languages always exhibit differences *and* commonalities. Depending on the level of linguistic abstraction at which one carries out a contrastive analysis of two or more languages, one will find more or less commonality or more or less differences. However, no matter how abstract the level is one chooses for cross-linguistic comparison, there will most likely always be differences. These are hard to capture, if one assumes an interlingua at one particular level (semantic, conceptual), at which everything is forced to be identical.[6] The method of describing multilinguality in KPML circumvents this problem by not enforcing one level of representation at which things have to be the same. Rather, several dimensions of cross-linguistic commonality *and* contrast are acknowledged and the contrastive-linguistic description can thus be just as flexible as it needs to be.

When we investigate the contrastive-linguistic properties of two or more languages using the representational categories of Systemic Functional Linguistics, we can make the observation that the grammars of different languages may be identical or different along the following dimensions:

- At the level of grammar, languages tend to be similar in terms *of functional paradigms* and different in terms of syntagmatic, surface-syntactic realization.
- Grammatical systems of low *delicacy* (grammatical types located high in the classification hierarchy) tend to be similar across language, and systems of higher delicacy tend to be dissimilar.
- There may be different preferences in different languages concerning the *rank* (clause, nominal group, prepositional phrase etc.) at which a certain phenomenon is expressed (e.g., nominally vs. verbally).
- Languages tend to exhibit more similarity on the higher *strata* of description and more differences on lower strata (e.g., more similarity on the semantic plane than on the lexico-grammatical plane).

[5] For more details on the tactical generation process in KPML see (Bateman, 1997a).

[6] This is a major problem with interlingua-based machine translation.

- All languages can be described based on the ideational, interpersonal and textual *metafunctions*.

The representational constructs of stratification, metafunctional diversification, ranking, axiality and delicacy thus set up a space of dimensions of multilingual variation along which languages can be described as being similar or different. For a graphical overview of these dimensions see Figure 5 (adapted from (Bateman, 1995)). This is what is made use of in contrastive-linguistic representation in KPML and underlies the possibility of what is called *resource sharing*.

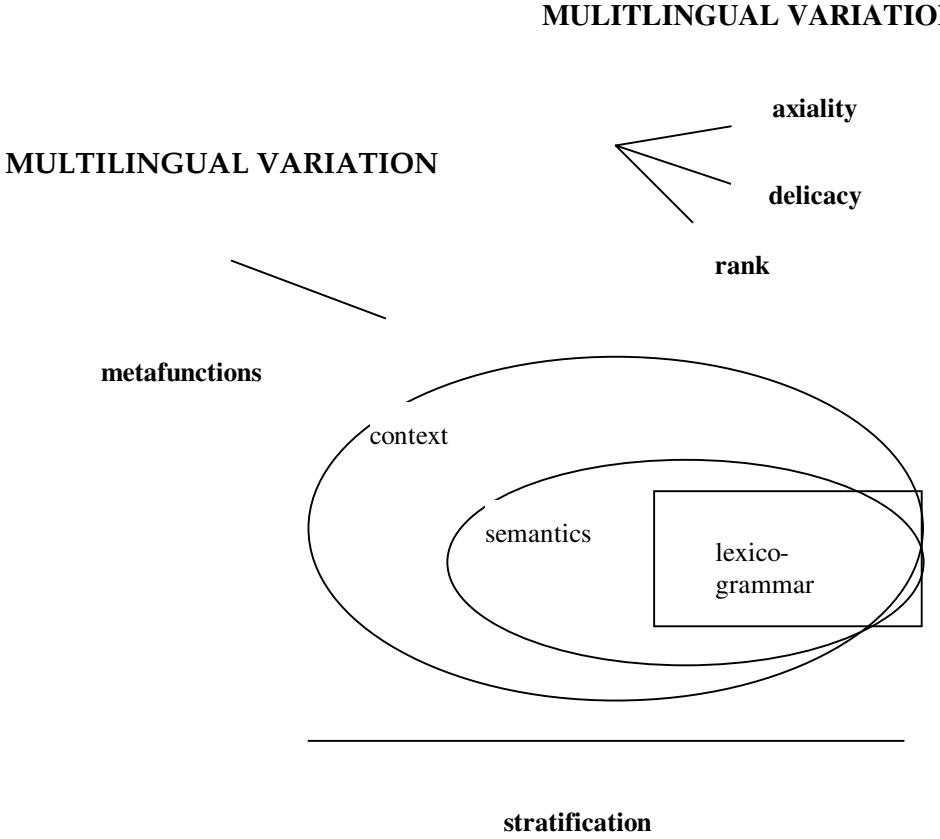



Figure 5: Dimensions of multilingual description

Given a computational specification of the grammar of a language according to rank, metafunction, axis and delicacy, and given that languages can exhibit commonality along these dimensions, one can try to re-use an existing grammatical description for the description of another language. Re-using an existing description will work to a large extent, especially of functional paradigms, but changes to the description taken as a basis must also be allowed for to account for language-specific features. For the languages we deal with here these are notably to do with the fact that Bulgarian, Czech and Russian are inflecting languages and exhibit a rather free word order (see Sections 4 and 5). The method of re-using an existing description of some language for the description of another language including the possibility of adapting that description is what is called *resource sharing* in KPML (cf. Bateman et al., 1991b; Bateman, 1995; Teich, 1995). More generally,

the strategy of building up grammatical descriptions on the basis of existing ones has been referred to as *transfer comparison* in the literature (Halliday *et al*., 1964).

The KPML development environment implements a series of tools that support resource sharing in the sense described above. Multilingual resource development is particularly supported in that multilingual descriptions are constructed around the paradigmatic, functional part of the grammar rather than the syntagmatic, surface-syntactic one. Thus, much greater commonality between languages can be detected and made use of, even between typologically different languages, thus potentially reducing the effort of writing grammars for new languages.

In the next section we give a brief overview of the functionalities of KPML that support developing multilingual resources on the basis of resource sharing.

## 3.2 Multilingual resource development in KPML

The KPML environment (Bateman, 1997a) is an extension of the functionalities of the Penman system (Penman, 1989), going beyond that system notably in terms of multilinguality, but also in terms of easier handling and better development support. Among the added functionalities for enhanced development support are the following features (cf. (Bateman, 1997a, p.2)):

- test suites are interlinked with resource definitions, thus providing the possibility of looking at the resources either from the instance perspective (a string) or from the grammar perspective (the system network);
- debugging is generally graphically driven;
- grammatical resources are highly modularized (monolingually and multilingually);
- extensive graphical and textual inspection of all aspects of the grammatical resources and their use are provided (for an example of one of the graphing facilities see Figure 6).

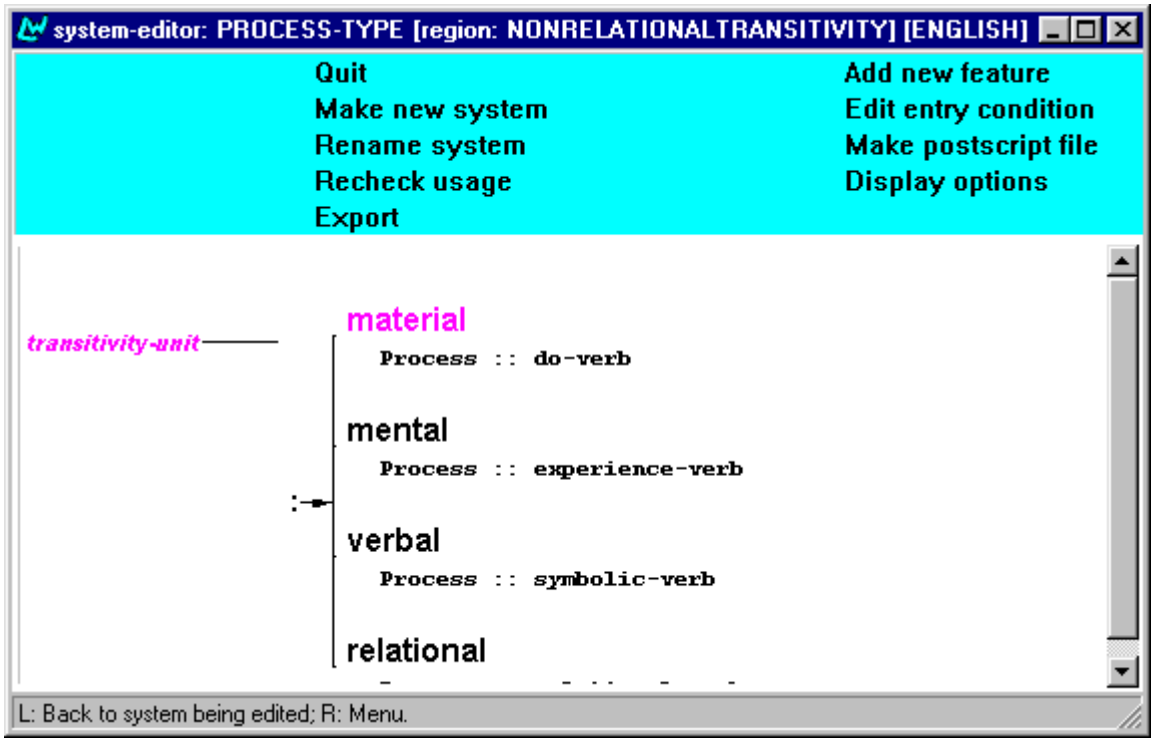



Figure 6: Graph of a system network having used the INSPECT option in the development window

In AGILE, KPML has been chosen as development platform because it has been explicitly designed for the purpose of offering generation projects large-scale resources that are well tested and appropriate for practical applications. Also, since KPML is based on the notion of *resource sharing* (see Section 3.1 above), resource development for new languages in KPML is considerably facilitated and resource development efforts are rather small compared to developing grammars from scratch. Also, KPML supports working in a contrastive-linguistic fashion, which in a multilingual project like AGILE must be the basic method of linguistic work.

The strategy of resource sharing has been followed rather closely in the first phase of resource implementation in AGILE, using the English grammar (NIGEL) contained in the KPML resources. The results that this has yielded are described in the following section.

For a list of references to KPML, the documentation of the system and the resources and recent add-on functionalities see Appendix I.

## 4. Grammatical resource implementation for Slavonic languages: Bulgarian, Czech and Russian

In the present section we describe the grammatical resources for Bulgarian, Czech and Russian implemented in the first phase of AGILE, which allow us to generate the sentences in the sample texts agreed upon for the initial demonstrator. The system network and SPL specifications are given in the Appendix, along with the SPLs for English.[7]

[7] We use English as a reference language throughout the project.

Figure 7 gives the notational conventions used in SFG; Figure 8 describes the syntax of systems network specifications in KPML.

| | |
|---|---|
| **functional elements** | Actor, Subject, etc |
| **system names** | MOOD |
| **grammatical features** | feature |
| **selection expressions:** | |
| **delicacy** | [feature-x : feature-y,...] |
| **simultaneity** | [feature-x & feature-y,...] |
| **realization statements**: | |
| **insert** | +Subject |
| **conflate** | Subject/Actor |
| **expand** | Mood(Finite) |
| **order** | Subject ^ Finite |
| **preselect** | Subject:nominal-group |
| **lexical constraints**: | |
| **classify** | Process::doing-verb |
| **inflectify** | Noun:::singular |
| **lexify** | Noun ! LEXEME |
| **syntactic structures** | box diagrams or trees |

Figure 7: Notational conventions in Systemic Functional Grammar

```
(system
 :name SYSTEM-NAME
 :inputs (OR feature-x
          (AND feature-y feature-z))
 :outputs ((0.5 feature-a
                (insert Function-1))
           (0.5 feature-b
                (conflate Function-2 Function-3)
                (preselect Function-2 feature-c)))
 :chooser SYSTEM-NAME-CHOOSER
 :region REGION-NAME
 :metafunction METAFUNCTION)
```

`:name` gives the name of the system; `:inputs` specifies the features that act as entry conditions to the system; `:outputs` specifies the features of the system, where features may have realization statements attached ( `insert, conflate, preselect` etc.); `:region` specifies the functional region the system belongs to---this can be thought of as a finer grained subclassification of metafunction given in the `:metafunction` slot and is used for the organization of resources.

Figure 8: Syntax of computational system network specifications

## 4.1 Bulgarian

**Clauses**

**Ideationally**, all the clauses of the sample texts express material directed actions, which involve an Actor and a Goal. It has thus been possible simply to re-use the TRANSITIVITY system of the English grammar

The **interpersonal** aspect of the clause is expressed by the MOOD system. The mood of the clauses occurring in the texts for the initial demonstrator is mainly imperative. There is only one verb form that realizes declarative mood, which is the third person singular reflexive *se pojavjava* in sample sentence 2b-2-b. Bulgarian imperative has two forms: a non-polite form (second person singular) and a polite form (second person plural). The options for mood in Bulgarian are thus different from English: while imperative is invariant and realized by a non-finite verb form in English, in Bulgarian it is realized by two options and finite verb forms. Like in English, the subject in imperatives is not expressed – however, since the verb form realizing the imperative is a finite form, person and number are reflected on the verb.

In instructional texts such as the ones we deal with in AGILE, the only option used in Bulgarian for expressing an imperative is the polite form. Therefore, we have simply used the English MOOD system as it is, using the imperative feature. The two imperative forms of the verb are accounted for as lexical entries. In the future, word forms are going to be generated by an external morphology component.

TRANSITIVITY and MOOD thus define the basic clause structure of the clauses occurring in the sample texts. The semantic input expressions from which these are generated always have the following information:

**SPL example 1 (fragment)**

```
(EXAMPLE
    :NAME   DB-TEXT1-3
    :TARGETFORM   "Zadaite ... tochka ..."
    :LOGICALFORM
    (E / DIRECTED-ACTION
         :LEX ZADAVAM
         :SPEECHACT IMPERATIVE
         :ACTEE (D / OBJECT
                 :LEX TOCHKA)))
```

where the actor of the action is not specified because we generate imperatives.

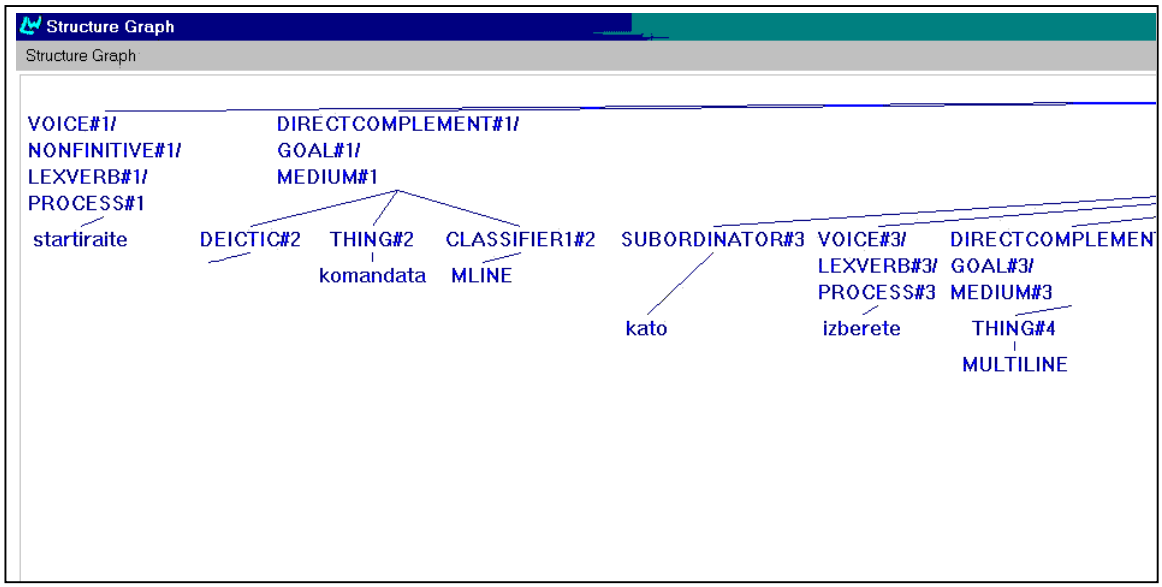
Apart from simplex clauses, also complex clauses occur in the sample texts. The complex clauses in the sample texts are mainly hypotactic, where the subordinating clause expresses a Purpose Adjunct or a Manner Adjunct. A Purpose Adjunct is expressed in Bulgarian by means of a subordinate finite clause (there is a Subject and the Finite agrees with the Subject in gender and number), the Subordinator is "*za da*" ("in order to"), as in sentence 2-3 of sample text 2a.

The expression of the Manner Adjunct in sample sentence 1-1 in the Bulgarian sample

text 1 slightly differs from the one in the equivalent English text in that in Bulgarian the verb in the subordinate clause is again finite, whereas in English it is nonfinite. Here, the Subordinator is "*kato*".

To account for these differences, the English Subordinators have been replaced by the Bulgarian equivalents in the realization statements of the appropriate system and the subordinate clause is preselected to be finite instead of nonfinite. See a sample generated structure exemplifying a Manner clause below.

**Output structure example 1**




## Nominal groups

The functional structure of nominal groups in Bulgarian is very similar to one of English (cf. Halliday, 1985):

| Deictic | Numerative | $Epithet_1$ | $Epithet_2$ | Classifier | Thing | Qualifier |
|---|---|---|---|---|---|---|

The main difference consists in the lack of **agreement** in gender and number between adjectives and ordinal Numeratives and the noun in English and the obligatory agreement between those elements in Bulgarian. Another major feature of the Bulgarian nominal group is that **determination** is expressed by means of a suffix attached to the noun, if the noun is the only constituent of the nominal group, and to the left-most constituent of the nominal group preceding the noun, if the noun is modified. In the sample texts we have nominal groups with and without Modifiers. In the ones with Modifiers, there are adjectives expressing Qualities and ordinals expressing Numeratives, which have to be suffixed, if the nominal group is definite. Also, there are different forms of the definiteness suffix according to gender in singular and one single form for plural for all three genders. So, definiteness is

expressed by *one* morpheme which is attached at the left most constituent of the nominal group and holds over the whole group. For some examples see the following:

*parvata tochka* (*the first point*)

*multiliniata* (*the multiline*)

*plavoshtoto menu* (*the flyout menu*)

At this stage we have simply kept the English Deictic as the first constituent of the nominal group, but it is prevented from being realized. The definite and indefinite forms of nouns, adjectives and numerals are accounted for as separate lexicon entries (see below). When a definite form of a noun, adjective or numeral is needed, the corresponding lexical item is used. With the integration of an external morphological module and the development of a grammatical system for deicticity in the next work phase, the issue of determination will be solved in a more principled way.

**Lexical entries example 1**

```
(LEXICAL-ITEM
    :NAME   MULTILINIA
    :SPELLING   "multiliniata"
    :FEATURES   (OUTCLASSIFY-PROPERNOUN NOUN COMMON-NOUN COUNTABLEES))

(LEXICAL-ITEM
    :NAME   MULTILINIA-NONDET
    :SPELLING   "multilinia"
    :FEATURES   (OUTCLASSIFY-PROPERNOUN NOUN COMMON-NOUN COUNTABLEES))
```

In Bulgarian **Classifiers** can appear before and after the Head noun, the Thing. Nominal Classifiers follow the Thing (as in *komandata PLINE* -- PLINE command), so we had to change the English default order of Classifier ^ Thing and specify this realization explicitly in the PRIMARY-CLASSIFICATION system (see example system 1 below).

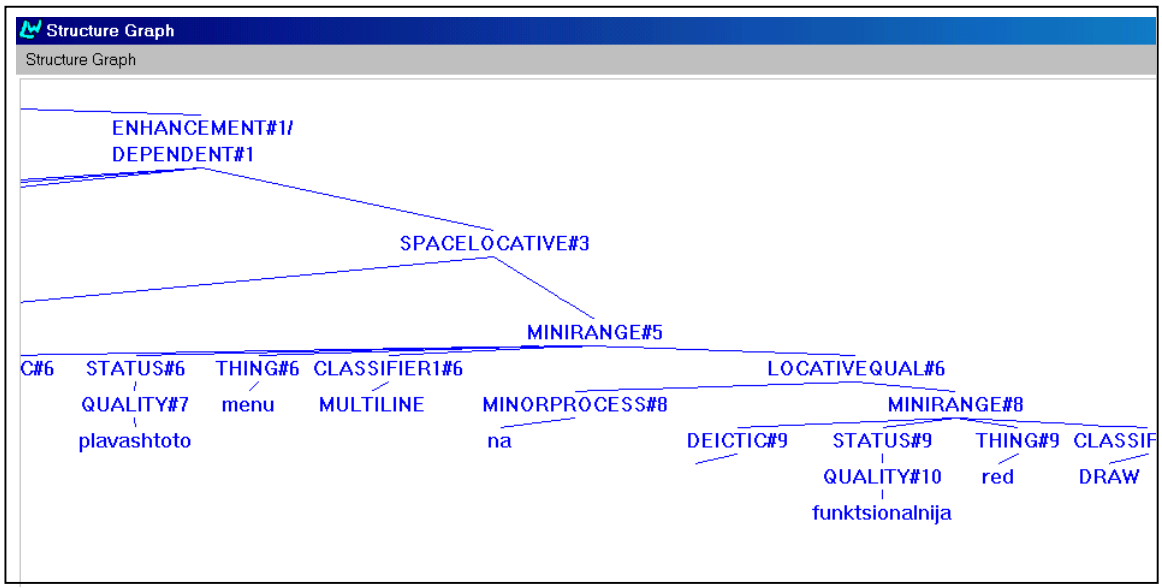
**System example 1**

```
(SYSTEM
    :NAME   PRIMARY-CLASSIFICATION
    :INPUTS   NOMINAL
    :OUTPUTS   ((0.5 PRIMARY-CLASSIFICATION
                     (INSERT CLASSIFIER1)
                     (CLASSIFY CLASSIFIER1 NOUN)
                     (ORDER THING CLASSIFIER1)
                     (INFLECTIFY CLASSIFIER1 NOUN-STEM))
                (0.5 PRIMARY-NONCLASSIFICATION))
    :CHOOSER   PRIMARY-CLASSIFICATION-CHOOSER
    :REGION   CLASSIFICATION
    :METAFUNCTION   IDEATIONAL)
```

Classifiers which are expressed by adjectives have to precede the noun in Bulgarian, however, just like Epithets do. For the time being, we have not implemented this option, but simply used property-ascription to account for Classifiers ordered before the Thing, as e.g.,

in *plvashtoto menu* or *funktsionalija red*, which makes *plvashtoto* and *funktsionalija* Epithets (shown as 'Quality' in the output structure example 2 below) rather than Classifiers.

**Output structure example 2**




We are aware that this treatment is not descriptionally adequate; again, in the next phase of work a more principled account will be developed.

## Prepositional phrases

There are only a few types of prepositional phrases we had to deal for the initial demonstrator. Most of the prepositional phrases occurring in the sample texts function as Space-Locative Adjuncts. The following examples occur in the sample texts:

(1) **Source** + Object, e.g., ***ot*** *plavashtoto menu Multiline (from the Multiline flyout)*

(2) **Spatial-locating** + **One-or-two-d-location**, e.g., ***na*** *funktsionalnija red Draw (on the Draw toolbar)*

(3) **Spatial-locating** + **Zero-d-location**, e.g., ***v*** *komandnija red (at the prompt)*

The corresponding SPLs are given below.

**SPL examples 2 (fragments)**

(1)

```
:SOURCE (C1 / OBJECT
```

```
              :PROPERTY-ASCRIPTION (Q1 / QUALITY
                                        :LEX PLAVASHTO)
              :LEX MENU
              :IDENTIFIABILITY-Q IDENTIFIABLE
              :CLASS-(C2 / SOFTWARE-COMMAND
                          :NAME MULTILINE)))
```

(2)

```
:SPATIAL-LOCATING (C3 / ONE-OR-TWO-D-LOCATION
                        :LEX RED
                        :PROPERTY-ASCRIPTION
                              (Q / QUALITY
                                   :LEX FUNKSTIONALEN)
                                    :IDENTIFIABILITY-Q IDENTIFIABLE
                                   :CLASS-ASCRIPTION
                                      (C2 / SOFTWARE-COMMAND
                                            :NAME DRAW)))
```

(3)

```
:SPATIAL-LOCATING (P / ZERO-D-LOCATION
                        :LEX RED
                        :PROPERTY-ASCRIPTION (Q / QUALITY
                                                  :LEX KOMANDEN)
                        :IDENTIFIABILITY-Q IDENTIFIABLE))
```

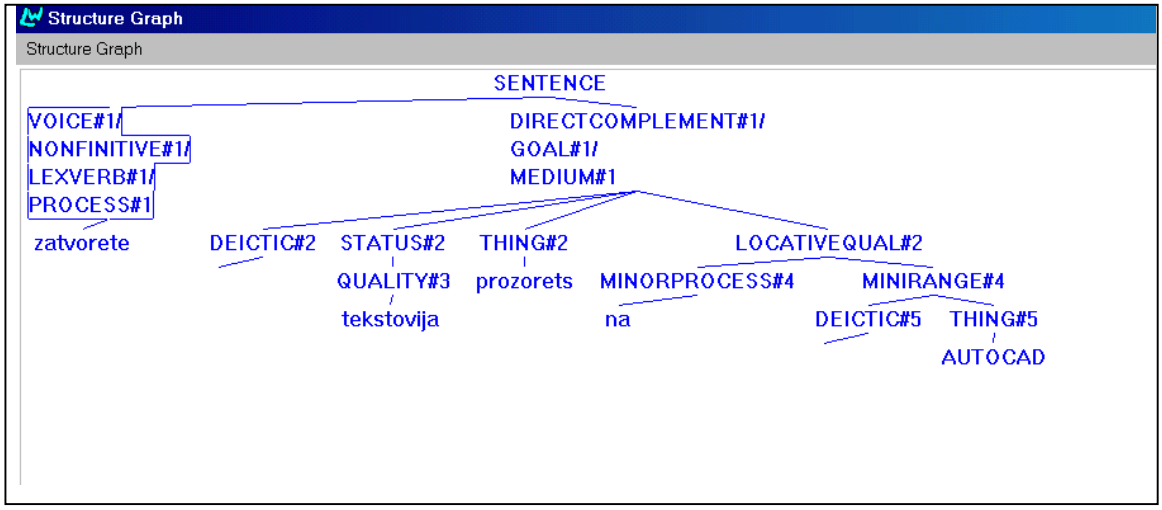
The only other type of prepositional phrase occurring in the sample texts can be modeled semantically by making use of the **part-of** relation. In the English grammar, this generates constructions like *the style **of** the multiline,* where *of the multiline* functions as a Qualifier. The Bulgarian equivalent of such constructions is also an analytic construction with the preposition *na* as in *stila na multiliniata.* Other examples are *mastaba na multiliniata* (*the scale of the multiline*), *spisaka na vidovete linii* (*the list of line types*), *imeto na stila* (*the name of the style*), *tekstovija prozorec na AutoCAD* (*AutoCad Text Window*). The example below shows an SPL for *Zatvorete tekstovija prozorets na AutoCad* (*Close the AutoCad text window).*

**SPL example 3**

```
(C / DIRECTED-ACTION
      :LEX ZATVARJAM
      :ACTEE (SC2 / OBJECT
                      :LEX PROZORETS
                      :PROPERTY-ASCRIPTION
                      (Q / QUALITY
                            :LEX TEKSTOV-NONFULLDET)
                      :IDENTIFIABILITY-Q IDENTIFIABLE
                      :PART-OF (O / OBJECT
                                    :NAME AUTOCAD)))
```

The structure generated from this SPL is shown in the graph below.

**Output structure example 3**




## 4.2 Czech

### Clauses

Like in Russian and Bulgarian, **ideationally** all the clauses occurring in the initial demonstrator texts express directed material actions. See a sample SPL for such a clause in SPL example 1 for the clause *Zadejte koncový bod cáry* (*Enter the end point of the line*).

**SPL example 1**

```
(EXAMPLE
      :NAME   D0-TEXT1-Cz-3-imp
      :SET-NAME   DB-TEXT1
      :TARGETFORM   " Zadejte koncový bod cáry."
      :LOGICALFORM
      (E / DIRECTED-ACTION
            :LEX ZADAT
            :SPEECHACT IMPERATIVE
            :ACTEE (D / OBJECT
                        :LEX BOD
                        :ORDINATION-Q ORDINATIVE
                        :ORDINATION-ID
                        (O1 / THING
                              :ORDINAL-TYPE-Q LAST
                              :ORDINATIVE-DETERMINATION-Q NOTSPECIFIED)
                        :ORDINATIVE-SELECTION-Q NONSELECTION
                        :IDENTIFIABILITY-Q IDENTIFIABLE
                        :PART-OF (P / OBJECT
                                    :LEX CÁRA
                                    :IDENTIFIABILITY-Q IDENTIFIABLE)))))
```

The **interpersonal** aspect of the clause is expressed by mood. In Czech, instructions in user manuals can be expressed by using either the declarative or the imperative mood. We have experimented with both moods. SPL example 1 above shows an SPL yielding a sentence in the imperative mood, which is grammatically realized with the Finite verb in second person plural, while SPL example 2 below shows an SPL yielding a sentence in declarative mood.

**SPL example 2**

```
(EXAMPLE
      :NAME   D0-TEXT1-Cz-3-decl
      :SET-NAME   DB-TEXT1
      :TARGETFORM   " Zadáme koncový bod cáry."
      :LOGICALFORM
      (E / DIRECTED-ACTION
            :LEX ZADAT
            :PROPOSAL-Q proposal
            :PERCEPTIVE-REACTIVE-PHENOMENON-Q PHENOMENON
            :ACTOR (SP / PERSON
                        :NUMBER PLURAL)
            :ACTEE (D / OBJECT
                        :LEX BOD
                        :ORDINATION-Q ORDINATIVE
                        :ORDINATION-ID
                        (O1 / THING
                              :ORDINAL-TYPE-Q LAST
                              :ORDINATIVE-DETERMINATION-Q NOTSPECIFIED)
                        :ORDINATIVE-SELECTION-Q NONSELECTION
                        :IDENTIFIABILITY-Q IDENTIFIABLE
                        :PART-OF (P / OBJECT
                                    :LEX CÁRA
                                    :IDENTIFIABILITY-Q IDENTIFIABLE)))))
```

Czech verbs have a quite rich morphology. There are different finite forms of verbs not only for the singular and plural number in third person as in English, but also for the first, and second person of each number. Moreover, Czech, like Bulgarian and Russian, distinguishes between ”formal” (polite) and ”informal” (non-polite) in the second person. The former is realized by second person plural, the latter by second person singular. Generally, this applies to declarative as well as imperative mood. However, the sentences in our corpus use only the second person plural form. The SPL displayed in SPL example 2 produces the sentence *My zadáme koncový bod cáry* rather than the desired *Zadáme koncový bod cáry*, in which the Subject pronoun *my (we)* is not realized in the surface form of the sentence. Czech is a pro-drop language, so the Subject is dropped when the entity referred to is easily recoverable from the context. The finite verb form carries sufficient information to distinguish between the different grammatical persons and number.

In order to achieve a satisfactory handling of Subject-dropping in Czech, we need to develop a strategy to determine whether a full nominal form of a referring expression should be used or whether pronominal form is or whether the Subject can be left out. This is part of text planning. In the tactical generation resources, systems for Subject-dropping have to be specified. We will address these issues in the next stage of linguistic specification and implementation. For the time being, we achieve Subject-dropping by defining the lexical realization of the pronoun *my* (*we*) as an empty string.

The choice between the declarative versus imperative mood appears to be a matter of style in Czech. Usually, a company producing manuals for their own software or translating manuals for localised software adheres to a particular style. So, one can encounter manuals where declarative mood is used by rule, or manuals where the imperative mood is used. In general, the declarative mood is considered more polite, it has more a feeling of a suggestion, while the imperative mood is commanding.

Although our linguistic judgement is that the declarative mood is more polite, while the imperative mood is perceived as commanding the user, it is the latter that is used in the instructions in our corpus taken from the translation of the AutoCad manual, so we have decided to choose the imperative mood for the purpose of the initial demonstrator. It is possible to include style variation in the later stages of the project.

In Czech, as well as in Bulgarian, English and Russian, the Subject of a sentence in the imperative mood is implicit in the unmarked case, and therefore no changes were required in this respect.

Of course, not all the sentences in the corpus nor in the initial demonstrator texts are instructions for the user to perform an action. The initial demonstrator texts contain one sentence in the indicative mood which expresses the side-effect of an action, namely *Objeví se dialogové okno AutoCad Text Window*. The SPL is shown in SPL example 3 below.

**SPL example 3**

```
(EXAMPLE
      :name DC-Text2b-2-b
      :targetform "Objeví se dialogové okno AutoCad Tex Window".
      :logicalform
            (C / NONDIRECTED-ACTION
                  :LEX OBJEVI-SE
                  :ACTOR (C1 / OBJECT
                              :LEX dialogove-okno
                              :CLASS-ASCRIPTION
                              (C3 / ONE-OR-TWO-D-LOCATION
                                    :NAME AutoCAD-Text-Window)))
```

In our implementation all word forms are specified directly as lexical entries, because the external morphological module is not integrated yet.

Apart from simplex clauses, the texts of the initial demonstrator also include paratactically and hypotactically linked clauses, namely coordinated clauses and sentences in which a Manner or Purpose Adjunct is expressed as a subordinate clause. Another way to express a manner relation is to use the domain of the relation as a kind of heading ended by a semicolon and to express the Manner Adjunct by a separate clause. Hypotactically related clauses of the this kind have not been accounted for yet.

As for Purpose Adjuncts, there is an alternative way of expressing them, namely as prepositional phrases with the preposition *pro (for)* and the argument of the preposition, realized as a nominal group, as a nominalization. Also Manner Adjuncts can be expressed in an alternative way using a nominal group in instrumental case. [8] SPL example 4 below shows the SPL for a sentence in which a purpose relation is expressed by a prepositional phrase, and SPL example 5 shows an SPL for a sentence involving a manner relation expressed by a nominal group.

**SPL example 4**

```
(EXAMPLE
      :NAME DC-TEXT1-4
```

[8] Our linguistic judgement is that these forms of expression are penetrations of the English original into the Czech translations, and they are not really appropriate Czech forms. However, they are the forms that are often used in user manuals, and are also used in our AutoCad corpus.

```
      :TARGETFORM   "Stisknete Enter pro uzavreni krivky."
      :LOGICALFORM
            (R / RST-PURPOSE
                  :DOMAIN
                  (P / DIRECTED-ACTION
                        :LEX Stisknout-Imp
                        :SPEECHACT IMPERATIVE
                        :ACTEE (D / OBJECT :NAME Enter))
                  :RANGE
                  (E / PURPOSE
                        :LEX Uzavreni
                        :RANGE (P2 / ABSTRACTION
                                    :PART-OF (O1 / OBJECT
                                                :LEX Krivka-Gen)))))
```

**SPL example 5**

```
(EXAMPLE
      :NAME DC-TEXT1-1
      :TARGETFORM   "Spustte prikaz PLINE vybranim prikazu Polyline"
      :LOGICALFORM
      (REL / (RST-MEANS MANNER)
            :DOMAIN
             (S / DIRECTED-ACTION
                  :LEX Spustit-Imp
                  :SPEECHACT IMPERATIVE
                  :ACTEE (C1 / OBJECT
                              :LEX Prikaz
                              :CLASS-ASCRIPTION
                               (CL1 / OBJECT  :NAME Krivka-Name1))))
            :RANGE
             (C / (DIRECTED-ACTION ABSTRACTION)
                  :LEX Vybrani-Instr
                  :ACTEE (P / OBJECT
                              :LEX Prikaz
                              :CLASS-ASCRIPTION (CL2 / OBJECT
                                                  :NAME Krivka-Name2)))))
```

**Nominal groups**

The functional structure of nominal groups encountered in the Czech initial demonstrator texts is quite similar to the English one. English does not display agreement, however, while in Czech there has to be **agreement** in case, gender and number between adjectives or ordinal numeratives and the head noun.

Another difference to English is that Czech does not allow premodification of a noun by another noun in the function of **Classifier** (as e.g., in *príkaz PLINE (PLINE command)*). Nominal modifiers always follow their heads. Adjectives, on the other hand, can both precede or follow the head. However, the latter is a marker order, and as such only occurs in special contexts or is considered archaic. The only ordering implemented is therefore the unmarked order (see also below).

A Qualifier in genitive case is used in Czech to express the part-of relation, as in *bod cáry (point of the polyline)*. An SPL fragment yielding such a nominal group is shown below:

**SPL example 6 (fragment)**

```
(c1 / object
      :lex bod
            :PROPERTY-ASCRIPTION
                  (Q3 / QUALITY :LEX koncovy )
            :PART-OF
            (C2 / OBJECT :LEX cary))
```

In the texts of the initial demonstrator, one encounters a word order phenomenon that is quite typical of the domain of software descriptions concerning the use of the names of commands, dialog boxes or other parts of the user interface. Like in English, Czech also uses a common noun together with a name to refer to such objects. We considered the common noun as a class ascription, and we have specified that class ascription precedes the thing in the surface word order in the corresponding realisation statement (see System Example 1 below)

**System example 1**

```
(SYSTEM
    :NAME   PRIMARY-CLASSIFICATION
    :INPUTS   NOMINAL
    :OUTPUTS   ((0.5 PRIMARY-CLASSIFICATION
                     (INSERT CLASSIFIER1)
                     (CLASSIFY CLASSIFIER1 NOUN)
                     (ORDER THING CLASSIFIER1))
                (0.5 PRIMARY-NONCLASSIFICATION))
    :CHOOSER   PRIMARY-CLASSIFICATION-CHOOSER
    :REGION   CLASSIFICATION
    :METAFUNCTION   IDEATIONAL)
```

The following example shows the SPL corresponding to *príkaz PLINE (PLINE command)*:

**SPL example 7 (fragment)**

```
      (O1 / OBJECT)
            :LEX prikaz
            :CLASS-ASCRIPTION (C2 / SOFTWARE-COMMAND
                                    :NAME PLINE)
```

When such a complex nominal group gets inflected, it is only the common noun that is inflected, not the name. This raises the question whether the name should not be considered a nominal Attribute.. A systematic treatment of this phenomenon will be developed in the next stage of the project.

Another difference between Czech and English is that Czech is a language in which **determination** does not have to be overtly marked. Czech does not have a direct equivalent to the English indefinite and definite article, but there are other means to signal definiteness or indefiniteness deployed in the nominal group (e.g., demonstrative determiners, and certain adjectives). Their use is not obligatory, however, and the nominal group in Czech does not need to have a determiner. By default, a nominal group without a determiner within the Theme position of the clause is considered definite in Czech. Up to this point, we have not

attempted to make changes in the determination systems because we need to carry out a more detailed investigation in order to make systematic changes. For the time being, we have kept the English Deictic as the first constituent of the nominal group, but when the Deictic is supposed to be realized, the constituent in the generated sentence is empty in the case of indefiniteness, and a demonstrative Deictic of the appropriate gender is used in the case of definiteness.

**Prepositional phrases**

The types of prepositional phrases present in the sentences of the initial demonstrator express source, spatial location, or purpose (as mentioned above). SPL example 8 shows the SPL fragment yielding a prepositional phrase functioning as source in *príkaz Polyline z plovoucího menu (Polyline command from flyout)*, and SPL example 9 shows the SPL fragment for the locative prepositional phrase *na nástrojovém panelu (in the toolbar)*.

**SPL example 8 (fragment)**

```
:SOURCE
      (C1 / OBJECT
            :PROPERTY-ASCRIPTION (Q1 / QUALITY
                                        :LEX plovouci-gen)
            :LEX menu)
```

**SPL example 9 (fragment)**

```
:SPATIAL-LOCATING
      (C3 / ONE-OR-TWO-D-LOCATION
                  :PROPERTY-ASCRIPTION (Q1 / QUALITY
                                              :lex nastrojovy-M2)
                  :LEX Panel-gen)
```

## 4.3 Russian

**Clauses**

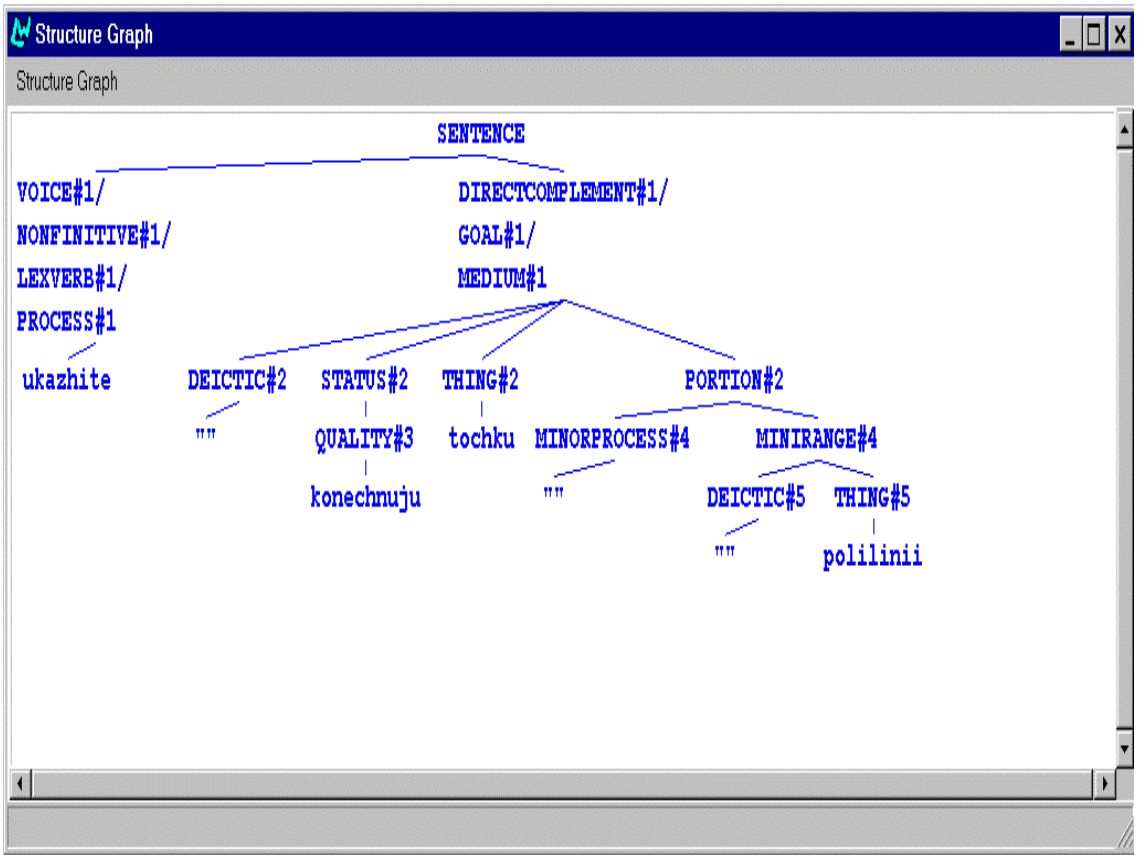
The basic **ideational** structure of the clause in Russian is the same as in English, consisting of Process, Participants and Circumstances (Adjuncts). Semantically, all processes in the sample texts for the initial demonstrator are directed actions, involving an Actor and an Actee role. Grammatically, all predicates thus have a direct complement which is realized by a nominal group that carries accusative case. By default a direct complement is ordered after the Finite verb. Circumstantial elements are semantically either of type source or of type spatial-locating and realized as prepositional phrases (see below). These types of processes and circumstantials make up the nuclear and circumstantial TRANSITIVITY of the kinds of clauses in the sample texts.

The range of **interpersonal** functions in the sample texts is restricted to imperative and declarative. Mood is realized primarily morphologically by verb inflection. The most frequent choice in the sample texts is imperative, which is realized without an explicit Subject, just like in English (see an example below).

***Ukazhite*** *konechnuju tochku polilinii. -- Specify the end point of the polyline.*

The functional structure of this clause is displayed below.

**Output structure example 1**




Systemically, there are two possible ways of expressing an imperative in Russian, a polite form and a non-polite form. The polite option is realized by using the second person plural form of the verb, the non-polite option is realized by using the second person singular form of the verb. The default choice in instructional texts, such as the ones we are dealing with in AGILE, is the polite option.

For simplex clauses, the functional elements of MOOD and TRANSITIVITY make up the basic clause structure. Apart from simplex clauses, there are also some clause complexes occurring in the sample texts. They are of three types:

(1) Logical or temporal coordination between two actions, as in:

*Vvedite "j" i vyberite vyravnivanie (Enter j and choose a justification)*

This case is accounted for in the English grammar as paratactic extension. The only change needed for Russian coordinate clauses consists in changing the lexemes for the Coordinators (see an example below).

**System example 1**

```
(SYSTEM
   :NAME    QUALIFYING-COORDINATION-TYPE
   :INPUTS    QUALIFYING-COORDINATION
   :OUTPUTS     ((0.5 TEMPORAL-COORDINATION
                      (LEXIFY COORDINATOR ZATEM)
                      (CONFLATE ENHANCED INITIATING)
                      (CONFLATE ENHANCEMENT CONTINUING))
                 (0.5 CAUSAL-COORDINATION
                      (CONFLATE ENHANCEMENT INITIATING)
```

```
                    (CONFLATE ENHANCED CONTINUING)
                    (LEXIFY COORDINATOR TAKIM-OBRAZOM)))
   :CHOOSER   QUALIFYING-COORDINATION-TYPE-CHOOSER
   :REGION   CLAUSECOMPLEX
   :METAFUNCTION   LOGICAL)
```

(2) Expression of the purpose of an action, as in:

*V stroke komand vvedite st, chtoby vybratj stilj. (Enter st at the prompt to select a style.)*

Purpose Adjuncts are expressed by infinitival clauses, which function as an elaboration of the main clause. The Subordinator *chtoby* marks this relation. In the grammar, the NONFINITE-PURPOSE-DEPENDENT system has been changed accordingly:

**System example 2**

```
(GATE
   :NAME   NONFINITE-PURPOSE-DEPENDENT
   :INPUTS   (AND PURPOSE-DEPENDENT NONFINITE-ENHANCING)
   :OUTPUTS   ((1.0 NONFINITE-PURPOSE-DEPENDENT
                    (LEXIFY SUBORDINATOR CHTOBY)
                    (INFLECTIFY NONFINITIVE INFINITIVE)))
   :REGION   DEPENDENCY
   :METAFUNCTION   LOGICAL)
```

(3) Expression of the means used for achieving a goal, as in

*Zapustite komandu MLINE, vybrav Multiline... (Start the MLINE command by choosing Multiline ...)*

Means Adjuncts can be expressed by using a special kind of participial construction (Russian: deeprichastije). The participial clause functions as an elaboration of the main clause, to which it is hypotactically linked. Typographically, the dependent clause is separated from the main clause by a comma. Participial clauses of this kind are treated as rankshifted clauses (cf. Halliday, 1985) here. For a sample system see below.

**System example 3**

```
(SYSTEM
 :NAME   MANNER-CONDITION-TYPE
 :INPUTS   (AND MANNER-DEPENDENT NONFINITE-ENHANCING)
 :OUTPUTS   ((0.5 MEANS-DEPENDENT
                  (INFLECTIFY NONFINITIVE ADV-PARTICIPLE))
             (0.5 COMPARISON-DEPENDENT))
 :CHOOSER   MANNER-CONDITION-TYPE-CHOOSER
 :REGION   DEPENDENCY
 :METAFUNCTION   LOGICAL)
```

## Nominal groups

The basic functional structure of nominal groups in Russian is the same as in English (cf. Halliday (1985)):

| | | | | | | |
|---|---|---|---|---|---|---|
| Deictic | Numerative | $Epithet_1$ | $Epithet_2$ | Classifier | Thing | Qualifier |

However, there are also a number of differences. The most notable difference of the Russian nominal group compared to the English one is the case system and the agreement between Thing (realized by a noun) and the functional elements from Deictic to Classifier (realized by adjectives) in case, number, gender and animacy.

Case (see System example 4 below) is preselected from clause or prepositional group rank. For agreement, the case feature is copied to all the premodifying elements of the nominal group that are present in the generated instance (see some examples in System example 5).

**System example 4**

```
(SYSTEM
 :NAME   NOUN-CASE
 :INPUTS   NOMINAL-GROUP
 :OUTPUTS   ((0.16667 NOMINATIVE)
             (0.16667 GENITIVE)
             (0.16667 DATIVE)
             (0.16667 ACCUSATIVE)
             (0.16667 INSTRUMENTAL)
             (0.16667 PREPOSITIONAL))
 :CHOOSER   NOUN-CASE-CHOOSER
 :REGION   PRONOUN
 :METAFUNCTION   INTERPERSONAL)
```

**System example 5**

```
(GATE
   :NAME   ACCUSATIVE-CASE
   :INPUTS   (AND ACCUSATIVE NOMINAL-GROUP-SIMPLEX)
   :OUTPUTS   ((1.0 ACCUSATIVE-CASE-T
                  (INFLECTIFY THING ACCUSATIVE)))
   :REGION   PRONOUN
   :METAFUNCTION   INTERPERSONAL)

(GATE
   :NAME   ACCUSATIVE-CASE-QUALITY
   :INPUTS   (AND ACCUSATIVE ADJECTIVAL-GROUP)
   :OUTPUTS   ((1.0 ACCUSATIVE-CASE-Q
                  (INFLECTIFY QUALITY ACCUSATIVE)
                  (INFLECTIFY QUALITY FEMININE)))
   :REGION   PRONOUN
   :METAFUNCTION   INTERPERSONAL)
```

Deictic elements in the English nominal group structure are expressed by possessive or demonstrative determiners. In Russian, in contrast, the deictic status of a nominal group is expressed by word order at clause rank and not marked in the nominal group. For the time being, we have simply changed the *a/the* realization of Deictic in the systems SPECIFIC-TYPE, INDIVIDUAL-DETERMINATION, ORDINATIVE-DETERMINATION and A-ANOTHER to a zero realization.

Another distinctive feature of the Russian nominal group is the ordering of nominal Classifiers, like in *komanda MLINE* (*the MLINE command*). The Classifier in this case is ordered after the Thing. This is achieved by changing the default order of Classifier ^ Thing in the ordering-constraints.gram file.

The implementation of genitive constructions in Russian (*imja stila - name of the style*) follows the pattern for English which uses the semantic **part-of** relation. However, while in

English, the part-of relation is realized by a prepositional phrase with 'of', in Russian, it is realized by a nominal group in genitive case.

**Prepositional phrases**

All the prepositional phrases occurring in the texts for the initial demonstrator are Circumstances of the type Spacelocative. Semantically, there are three subtypes:

(1) **Source** + Object, e.g., *vybratj* ***v*** *palitre Polyline* (*choose from the Polyline flyout*)

(2) **Spatial-locating** + **One-or-two-d-location**, e.g., ***na*** *paneli instrumentov Draw* (*on the Draw tool bar*)

(3) **Spatial-locating** + **Zero-d-location**, e.g., *vvedite* ***v*** *stroke komand* (*enter at the prompt*)

The implementation of these options in the grammar is straightforward, only the realization statements of the English systems have to be changed. For a sample system see System example 6 below.

**System example 6**

```
(GATE
 :NAME   ON
 :INPUTS   (AND TOWARDS ONE-OR-TWO-DIMENSION)
 :OUTPUTS   ((1.0 ON
                  (LEXIFY MINORPROCESS NA)))
 :REGION   PPSPATIOTEMPORAL
 :METAFUNCTION   IDEATIONAL)
```

## 5. Summary and Outlook

In effectively three months[9] sentences of the complexity occurring in the sample texts in Bulgarian, Russian and Czech [10] could be generated (see sample texts in Appendix II)---even though currently there are only small fragments of Bulgarian, Czech and Russian grammar. This has only been possible because we could make use of the strategy of resource sharing as adopted in KPML and employ the support tools offered by KPML for using such a strategy. The English grammar in KPML, NIGEL, has been used as the basis for resource sharing. English is a Germanic language and therefore typologically different from Slavonic languages, such as Bulgarian, Czech and Russian, but since the main body of linguistic description in Systemic Functional Grammar is functional and resource sharing operates primarily on functional categories, this approach has been successful. Also, as an observation from linguistic specification and implementation, even though Bulgarian belongs to the family of Slavonic languages, on a scale from analytic to synthetic, Bulgarian is much closer to English than Czech or Russian: Bulgarian only has rudimentary morphological case and it thus operates with prepositions rather than with

[9] Work on Task 7.1 started only in the second week of February with the kick-off workshop in Prague. For another month the participants in this Work Package only had a stand-alone image of KPML and various grammar exploration tools at their disposal. During that time, KPML was ported to Harlequin Lisp; the Harlequin version was distributed in mid-March.

[10] All the sentences of texts 1, 2a and 2b can be generated in Bulgarian and Russian; for Czech the larger portion of the sentences can be generated.

case. Correlating with the lack of case is the fact that word order is not as flexible as in Czech or Russian.

So far, we cannot claim that we have general specifications of the grammars of Bulgarian, Czech and Russian. However, the methodological steps from here are clear and the results we have achieved so far will easily scale up. While we have so far just changed existing systems very locally, for instance, PRIMARY-CLASSIFICATION in all three languages, MOOD for imperative in Czech and Russian and for declarative in Czech, in the next phase of implementation, we are going to deal with whole paradigms and work according to functional regions (transitivity, circumstantials, qualification, classification etc). The grammatical complexity of the texts occurring in the kinds of manuals we deal with from the CAD-CAM domain will not be much higher than already covered, so that we can concentrate on developing more general solutions. Also, while resource sharing has worked quite well generally, there are some issues to be dealt with that relate to the typological particularities of Bulgarian, Czech and Russian as Slavonic languages. Among these are the following. On the technical side, for Czech and Bulgarian external morphology components will be linked up with KPML, just as it has been done for Russian. On the linguistic modeling side, we have to develop a treatment for word order, which is highly variable in Russian and Czech and almost entirely cotextually and contextually determined (topic-focus articulation, information distribution). Also, in all of the three languages there are some rather complex agreement phenomena, particulary in the nominal group, that need a principled treatment.

The experiences in this first phase have shown that it is more than worthwhile to work contrastively across Bulgarian, Czech and Russian and possibly distribute the descriptive and specification efforts, thus not only sharing resources with English, but also among the three languages under investigation in this project.

The results of this phase of work in Work Package 7 also feed into other Work Packages, notably text planning (Work Package 5). The SPLs constructed manually for the sample texts give us the target structure the text planner has to generate. Systematically comparing these SPLs cross-linguistically has provided us with the lowest common denominator for the three languages, which can be taken as the target structure for a multilingual part of text planning; and we have a list of cross-linguistic differences at the level of SPL that have to be dealt with individually for each language (see Deliverable TEXM1 of Work Package 5).

## References

Agile Project (1997) Automatic generation of instructions in languages of Eastern Europe. Technical Annex, Commission of the European Community, Brussels, DG XIII, 1997.

Bateman J. A. and E. Teich (1995) Selective information presentation in an integrated publication system: an application of genre-driven text generation. *Information Processing and Management*, 31(5):753--768, September 1995. (Special Issue on Summarizing Text).

Bateman J. A., R. T. Kasper, J. D. Moore, and R. A. Whitney (1990). A general organization of knowledge for natural language processing: the PENMAN Upper Model. Technical report, USC/Information Sciences Institute, Marina del Rey, California, 1990.

Bateman J. A., E. A. Maier, E. Teich, and L.Wanner (1991a) Towards an architecture for situated text generation. In *Proceedings of the International Conference on Current Issues in Computational Linguistics*, pages 336--349, Penang, Malaysia, 1991.

Bateman J. A., C.M.I.M. Matthiessen, K. Nanri, and L. Zeng (1991b) 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, volume~2, pages 966 -- 971. Morgan Kaufmann Publishers, 1991.

Bateman J. A., Th.Kamps, J. Kleinz, K. Reichenberger (1998) Communicative goal-driven NL generation and data-driven graphics generation: An architectural synthesis for multimedia page generation. *In Proceedings of the International Workshop on Natural Language Generation*, Niagara-on-the-lake, Canada, 1998.

Bateman J. A. (1992) Grammar, systemic. In S. Shapiro, editor, *Encyclopedia of Artificial Intelligence, Second Edition*, pages 583 -- 592. John Wiley and Sons, Inc., 1992.

Bateman J. A. (1995) Basic technology for multilingual theory and practise: The KPML development environment. In R. Kittredge, editor, *Proceedings of the IJCAI '95 Workshop on Multilingual Generation*, Montreal, Quebec, August 1995.

Bateman J. A. (1997a) KPML Development Environment: multilingual linguistic resource development and sentence generation. German National Center for Information Technology (GMD), Institute for Integrated Publication and Information Systems (IPSI), Darmstadt, Germany, March 1997. (Release 1.0).

Bateman J. A. (1997b) Enabling technology for multilingual natural language generation: the KPML environment. Natural Language Engineering 1(1):1-42, 1997.

Danes F. (1974) Papers on Functional Sentence Perspective: Academia, Prague, 1974.

Dik S. (1978) *Functional Grammar*. North Holland, Amsterdam, 1978.

DiMarco C., G. Hirst, L. Wanner, and J.Wilkinson (1995) Healthdoc: Customizing patient information and health education by medical condition and personal characteristics.In A. Cawsey, editor, *Proceedings of the Workshop on Patient Education*. University of Glasgow, Glasgow, 1995.

Elhadad M. and J. Robin. (1996) An overview of SURGE: A reusable comprehensive syntactic realization component. Technical Report 96-03, Dept of Mathematics and Computer Science, Ben Gurion University, Beer´Sheva, Israel, 1996.

Elhadad M.(1990) Types in functional unification grammars. *In Proceedings of the 28th. Annual Meeting of the Association for Computational Linguistics*, pages 157 --164. Association for Computational Linguistics, 1990.

Fawcett R. P. and G. H. Tucker (1989) Prototype generators 1 and 2. Technical Report COMMUNAL Report Number 10, Computational Linguistics Unit, University of Wales College of Cardiff, 1989.

Firbas J. (1966) On defining the theme in functional sentence analysis. *Travaux Linquistique de Prague*, 1:267--280, 1966.

Halliday M.A.K. and C.M.I.M. Matthiessen (to appear) *Construing experience through meaning: a language-based approach to cognition*. Cassell Academic, London, to appear.

Halliday M.A.K., A. McIntosh, and P. Strevens (1964) *The linguistic sciences and language teaching*. Longman, London, 1964.

Halliday M.A.K. (1973). *Explorations in the Functions of Language*. Edward Arnold, London, 1973.

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

Halliday M.A.K. (1985) *An Introduction to Functional Grammar*. Edward Arnold, London, 1985.

Hartley A. F. and C. Paris (1997) Multilingual document production: from support for translating to support for authoring. *Machine Translation* (Special Issue on New Tools for Human Translators), 12(1-2):109--129, 1997.

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, August 1994.

Hjelmslev L. (1943) *Omkring sprogteoriens grundlaeggelse*. Akademisk Forlag, Kopenhavn, 1943.

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*, 1989. Available from USC/Information Sciences Institute, Marina del Rey, CA.

Mann W. C. and C. M.I.M. Matthiessen (1983) Nigel: A systemic grammar for text generation. Technical Report RR-83-105, USC/Information Sciences Institute, February 1983. (Also appears in R. Freedle, ed.itor, *Systemic Perspectives on Discourse: Volume I*, published by Ablex)

Mann W. C. (1983) An overview of the PENMAN text generation system. In *Proceedings of the National Conference on Artificial Intelligence*, pages 261--265. AAAI, August 1983.

Matthiessen C. M.I.M., I. Kobayashi, L. Zeng, and M. Cross (1995) Generating multimodal presentations: resources and processes. *In Proceedings of the Australian Conference on Artificial Intelligence*}, Canberra, 1995.

Matthiessen C. M.I.M., L. Zeng, M. Cross, I. Kobayashi, K. Teruya, C. Wu (1998) Communicative goal-driven NL generation and data-driven graphics generation: An architecture for multimedia page generation. *In Proceedings of the International Workshop on Natural Language Generation*, Niagara-on-the-lake, Canada, 1998.

Matthiessen C. M.I.M. (1988) Semantics for a systemic grammar: The chooser and inquiry framework. In M. Cummings, J. D. Benson, and W. S. Greaves, editors, *Systemic Perspectives on Discourse*. John Benjamins, Amsterdam,1988.

McKeown K., M. Elhadad, Y. Fukumoto, J. Lim, C. Lombardi, J. Robin, and F. Smadja (1990) Natural Language Generation in COMET. In R. Dale, C. Mellish, and M. Zock, editors, *Current Research in Natural Language Generation*, pages 103 -- 139. Academic Press, London, 1990.

Paris C., K. Vander Linden, M. Fischer, A. F. Hartley, L. Pemberton, R. Power, and D. Scott (1995) A Support Tool for Writing Multilingual Instructions. *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 1995*, pages 1398 -- 1404, Montreal, Canada, 1995.

Penman Project (1989). PENMMAN documentation: the Primer, the User Guide, the Reference Manual, and the Nigel manual. Technical report, USC/Information Sciences Institute, Marina del Rey, California, 1989.

Pollard C. and I. A. Sag (1987) *Information-based syntax and semantics*: *volume 1*. Chicago University Press, Chicago, 1987.

Pollard C. and I. A. Sag (1993*) Information-based syntax and semantics: volume 2*. Chicago University Press, Chicago, 1992.

Roesner D. and M. Stede (1992) TECHDOC: a system for the automatic production of multilingual technical documents. Technical Report FAW-TR-92021, Forschungsinstitut fuer anwendungsorientierte Wissenverarbeitung (FAW) an der Universitaet Ulm, Ulm, Germany, September 1992.

Roesner D. and M. Stede (1994) Generating multilingual documents from a knowledge base: the TECHDOC project. *In Proceedings of the 15th. International Conference onComputational Linguistics* (Coling 94), volume I, pages 339 -- 346, Kyoto, Japan, 1994.

Sgall P., E. Hajicova, and J.Panevova The Meaning of the Sentence in Its Semantic and Pragmatic Aspects. Reidel Publishing Company, Dordrecht, 1986.

Teich E., L. Degand, and J.A. Bateman (1996) Multilingual textuality: Experiences from multilingual text generation. In G. Adorni and M. Zock, editors*, Trends in Natural Language Generation: an artificial intelligence perspective*, number 1036 in Lecture Notes in Artificial Intelligence, pages 331--349. Springer-Verlag, Berlin, New York, 1996.

Teich E. (1995) Towards a methodology for the construction of multilingual resources for multilingual generation. In R. Kittredge, editor*, Proceedings of the IJCAI '95 Workshop on Multilingual Generation*, Montreal, Quebec, August 1995.

Teich E. (in press) Systemic Functional Grammar in Natural Language Generation: Linguistic Description and Computational Representation. Cassell Academic, London, in press.

Thompson H. (1977) Strategy and tactics: A model for language production. In *Proceedings of the 13th Annual Meeting of the Chicago Linguistics Society*, pages 89 – 95.

Wanner L. and E. H. Hovy (1996) The HealthDoc sentence planner. *In Proceedings of the 8th International Workshop on Natural Language Generation*, Herstmonceux, UK, June 1996.

Yang G., K. F. McCoy, and K. Vajay-Shanker (1991) From functional specification to syntactic structures: Systemic Grammar and Tree Adjoining Grammar. *Computational Intelligence*, 7(4):207 --219, 1991.