In Defense of Impractical Machine Translation Systems

Alexandr Rosen*

1 Introduction: Machine translation system as a linguistic testbed

Most efforts related to machine translation (MT) aim at building a system with a strong potential of usability outside the laboratory. This may be related to the fact that (machine) translation is something which strikes everyone as a practically useful enterprise, as opposed to some other tasks pursued in theoretical and computational linguistics, such as the design and construction of formal representations of linguistic knowledge and meaning of expressions.

On the other hand, the discipline of MT is notorious for its mixture of theoretical and engineering aspects, which makes many projects, viewed from the academic perspective, difficult to appreciate. As a result, MT is seldom considered as a suitable testbed for theoretical research, where hypotheses about language can be formulated and verified.[1] Preference is given to less complex monolingual environments: the usual practice is to implement a grammar for parsing, sometimes for generation, occasionally for both. It then becomes difficult to argue for validity of the results across more languages.

In what follows, I would like to oppose the view that MT is necessarily a practical enterprise, not quite suitable as an environment for the development and verification of linguistic hypotheses. In support of this claim, an instantiation of the 'impractical' approach to MT will be presented. The bidirectional English–Czech system, declaratively specified and implemented within ELU (Environnement linguistique d'unification, see Estival, 1990, and Russell et al., 1992) will be shown as open to theoretically interesting experimentation with alternative solutions at various levels of modularity and claimed to represent a kind of 'linguistic playground'.

Some of the factors which make a MT system attractive for linguistic research are the following:

*Institute of Theoretical and Computational Linguistics, Faculty of Philosophy, Charles University, Prague, e-mail: alexandr.rosen@ff.cuni.cz. The work has been supported by Research Support Scheme of the Open Society Institute (project No. 340/93) and by the Grant Agency of the Czech Republic (project No. 405/96/K214).

[1] This does not hold universally: some rather application-oriented MT projects have been bringing lots of theoretically interesting results, as in the case of large-scale projects EUROTRA and VERBMOBIL. Similarly, although on a different scale, the slowly strengthening practical orientation of the MT systems developed in Prague (see, e.g. Kirschner & Rosen, 1989, and Hajič et al., 1992) has not deprived them of theoretical relevance.

- The bilingual or multilingual environment is a guard against the temptation to present language-internal facts as linguistic universals.
- At the same time, such an environment is a strong motivation for searching as much common ground as possible across different languages.
- Representation of meaning, extracted from the input, is testable by its translation into the target language. If the system is reversible, a more profound test can be performed by translating the result back into the source language.
- The contrastive framework of a MT system is an attractive field for modelling the (surface) word order regularities and their implications for the representation of meaning. The comparison of languages with different factors determining surface word order (such as Czech and English) is especially fruitful.
- As a more general point, MT embraces most problems of NLP, and thus supports compatibility across theories, formalisms, and processing tools.[2]

The above factors are consequences of the very nature of MT. However, to turn a system into a real 'impractical' linguistic playground, additional requirements should be made:

- The implementation formalism should be able to reflect theoretical assumptions as directly as possible, without the necessity to invent elaborate ways of translating (both ways) between the formal language of the linguistic theory and the formal language of the implementation.
- The system should be highly modular (easy to modify, extend, parameterize, cut-and-paste and re-use along many dimensions).
- The system should allow for easy incorporation of language-universal and language-specific features wherever appropriate.

2 Outline of a playground

The basic parameters of the system are currently set as follows:

theory T: (i) a constraint-based framework for the description of surface syntax as a relation between surface string and its representation in the spirit of HPSG (Head-driven Phrase Structure Grammar — see Pollard & Sag 1994) combined with (ii) FGD-like dependency representation of expressions at the underlying level of tectogrammatics (Functional Generative Description — see Sgall, Hajičová & Panevová 1996)

grammar and lexicon G: (partial) descriptions of English and Czech by means of formalism F within theory T, employing similar concepts and solutions, neutral with respect to the direction of processing

transfer Tf: (partial) description of the relation between underlying representations of English and Czech expressions by means of formalism F within theory T, neutral with respect to the direction of processing

[2] Of course, discovery of a weak point related to the MT task in the formalism and software tools is beneficial in the other direction: in stimulating the development of better tools.

formalism F**:** declarative, constraint-based à la PATR-II with some extensions

implementation I**:** compiler of G, extended Earley parser (prediction through restrictors), head-driven generator, dictionary look-up module, transfer component

Not all of the basic parameters of the system are equally easy to modify and they are strongly inter-related, too, e.g. the choice of a constraint-based formalism restricts all the other choices. Consequently, a linguistic theory which does not subscribe to the view that all linguistic hypotheses can be expressed and verified in a constraint-based setting requires a different formalism and processing tools.

Although some components, especially lexical data, have been acquired and structured with the aim of maintaining re-usability in different, possibly practically-oriented, environments, the major part of work has been directed towards the construction of a flexible partial model of linguistic competence in the two languages.

In the rest of this section, features of the system common to both the English and Czech parts will be described.

2.1 Formalism and processing tool

The choice of ELU — Environnement linguistique d'unification (Estival, 1990, Russell et al., 1992) as the formalism and processing tool has been made primarily because it includes the modules of parsing, generation and transfer within a uniform constraint-based declarative environment. An additional factor has been the availability of a powerful lexical component. On the other hand, ELU no longer represents a state-of-the-art tool, e.g. in its treatment of feature structure typing.[3]

The ELU formalism is based on PATR-II with a few extensions.[4] The more important features include: (i) the possibility of specifying lexical description as a hierarchy of classes with controlled multiple inheritance and defaults; (ii) the design of the transfer module as a declarative, contraint-based extension of the monolingual formalism, possibly reversible, where locality is preserved and recursion controlled by general means; and (iii) the possibility of reversible interpretation of a single grammar and lexicon in both parsing and generation.

Both monolingual (Czech and English) components consist of five modules:

General specifications — declarations of types, restrictors and other parameters common (i) to the entire system, and (ii) to the description of a language.

Definition of macros — specification of general principles or recurrent constraints in lexicon and grammar.

Definition of lexical classes — description of lexical data which are not specific to a lemma. The data are organized according to their specificity in

[3]Types are merely optional and an explicit type hierarchy cannot be defined. Also, the lexical component does not allow for lexical rules (if, indeed, one considers them as useful) and does not interact smoothly with multiple macro definitions and disjunctions.

[4]In a addition to the above-mentioned possibility of user-defined types, there are lists and trees as built-in data types, disjunction and negation can be applied to atomic constants, and macros (templates) allow for parameters, multiple and recursive definitions.

a hierarchy of classes. A class may correspond e.g. to a part of speech, with different inflection paradigms represented as its subclasses. For every class, a list of superclasses can be defined. A more specific constraint in a lower class may override a conflicting less specific constraint in a higher class (a default). The default mechanism is controlled (i) by the order of superclasses as defined for a class, and (ii) by grouping constraints specifying a class into a main (default) set of contraints (possibly) multiple disjunctive sets of constraints.

In the example below, three lexical classes and two lemmas are defined. The class `Word` stands at the top of the hierarchy, dominating the class `Verb`. The lemmas `lézt` 'to crawl' and `prát` 'to wash (laundry)' stand below the class `Verb`. The lemma `prát` is dominated by another class `Trans`. Specification of a transitive verb in the class `Trans` overrides the default specification of an intransitive verb in the class `Verb`. The variable `Paradigm` is instantiated for `prát` with the value `bere`. This value selects the correct variant set of constraints describing the relevant inflexion paradigm in the class `Verb` (all other variant sets have a different value of the variable `Paradigm` and are excluded by unification failure).

```
# Class  Word ()
  <the most general specification of a lexical item>

# Class  VerbInfl (Word)
  | Form = Inf  <vform> = inf
  | Form = Sg1  <vform> = fin  <agr num> = sg  <agr pers> = 1
  | Form = Sg2  <vform> = fin  <agr num> = sg  <agr pers> = 2
  ...

# Class  Verb (VerbInfl)
  <general specification of an intransitive verb>
  | Paradigm = nese
    <specification of the inflection paradigm 'nese'>
  | Paradigm = bere
    <specification of the inflection paradigm 'bere'>
  | ...

# Class  Trans ()
  <general specification of a transitive verb>

# Word   lézt   (Verb)  Paradigm = nese
# Word   prát   (Trans Verb)  Paradigm = bere
```

Dictionary of lemmas — includes information specific to individual lexical items: base form, word class, inflection type, valency type. See the English and Czech example entries below:

```
receipt     (CStemN CountUncount Noun)
receive     (EStemV Verb) Frame=n
reciprocity (NoPlN Uncount Noun)
refer       (DblStemV Verb) Frame=pp-obj_/n_pp_.
            /n_pcle_ Prep-Obj=to Pcle-Obj=back
regard      (CStemN CountUncount Noun)
regard      (CStemV Verb) Frame=n/n_pp_./n_ping-objc_
            /n_pp-objc_ Prep-Objc=as

pod         (Prep) PCase=ins/acc
pole        (MR1 Noun)
pomáhat     (Impf DE1 Verb) Frame=n/iobj_inf
```

```
              ObjCase=dat
pomoci        (Perf PE3 Verb) Frame=n/iobj_inf
              ObjCase=dat
předseda      (PD1 Noun)
příští        (JR Adj)
```

Grammar rules — each covering a number of structural types and invoking general principles implemented as macro defintions. The example rule from the Czech grammar licenses a complement to the right of a verb.

```
# Rule CV

CV -> C H
     !Categories(H,C,vbar,v/vbar,_,no)
     !Cz_Head_Complement_Rule(H,C)
     !CDyn(C,H)
     !Orthography(C,H)
     !Empty_Bind(H)
     !Mod(C,H)
     !RH_Clitics(C,H)
## heuristics
     <C synsem loc modified right> = no
     <C synsem loc modified left> = no
```

2.2 Lexicon

In a constraint-based formalism, lexicon is not only a repository of lexical knowledge, but also the source of formal structural descriptions providing the basis for descriptions of higher units. In this system, the lexicon combines specification of lemmas with a description of inflection paradigms and provides a representation of word forms, isomorphic with the representation of higher syntactic structures. Some of its functions are equivalent to lexical rules.

Thus, lexicon and morphology are integrated into a single **hierarchy of lexical classes**. The most specific classes are the lowest in the hierarchy and correspond to lemmas, while the most general class at the top is characterized (constrained) by features shared by any lexical item. A lexical form is identified through the interaction of specifications of the lemma and of all its superordinate classes in the hierarchy. Lower classes recursively inherit specifications from higher classes, unless a lower class specifies the same feature differently from a higher class. In such a case, the more specific lower class overrides the more general specification of the more general higher class in this feature, i.e., overrides a default value of this feature. Furthermore, a class may inherit specifications from more than one immediately dominating classes. To prevent a possible conflict in different values of the same attribute inherited from two immediately dominating classes, a user-defined precedence of those classes decides which value is overridden.[5]

2.3 Grammar

The system's grammar components are a combination of (i) a unification-based monostratal syntactic model with a context-free backbone, inspired by HPSG

[5] If constraints of two immediately dominant classes A and B are in conflict, the precedence order of classes stated in the heir's definition is taken into account. A constraint of class A overrides a conflicting constraint of class B just in case A precedes B in the list of classe s which are above the heir. The constraint of B is therefore treated as default.

in its treatment of surface-level syntactic phenomena, (ii) and a dependency-based 'underlying' representation of 'linguistic meaning' in the spirit of FGD. The grammar describes the relationship between the surface string and its representation via CF rules with constraints. The CF rules and their dependency interpretation can be viewed as two (surface and deep) levels of description: SL and DL.

Some typical cases where this distinction is used are the following:

- Function words are represented at DL as features related to content words rather than as independent nodes of a dependency tree.
- Governor-dependent relations of DL dependency trees are explicitly labelled by function names.
- Some SL heads do not correspond to DL governors (auxiliary verbs).
- Long-distance dependencies are represented at DL as projective trees.
- A single constituent is shared (co-indexed) at DL by two or more nodes in sentences including controlled embedded predicates.
- Some SL categories are not represented as DL categories of the same type (deverbatives).
- The number of dependents of a governor at DL is not necessarily the same as the total number of non-head sisters to the head at SL at all projection levels (particles, complements of a raising predicate – present at SL but missing at DL).
- Some SL complementation requirements do not equal corresponding DL valency frames: at DL, there are no function words, no valency slots for particles, raising predicates miss a complement.

Grammar rules fall into the following eight types, which can be interpreted at DL as follows:

head_only_rule: specification of a DL feature due to the absence of some lexical item(s) (bare plural in English)

head_complement_rule: governor + any obligatory dependent

head_adjunct_rule: governor + any optional dependent

head_marker_rule: specification of a DL feature due to a function word

aux_head_complement_rule: specification of a DL feature due to an auxiliary verb

coord_rule: recursive specification of a coordinated structure

trace_rule: detection of a potential non-projective dependency

head_filler_rule: identification of a non-projective dependent

The grammar rules obey grammar principles (by invoking relevant macros), such as *head feature principle, deep head feature principle* (taking care of head features relevant at DL), *daughters principle, subcategorisation principle, specifier principle* and *non-local feature principle.*

Throughout the system, an attempt is made to distinguish the surface and 'deep' word order. The surface order in English is determined predominantly by syntactic functions of constituents, while in Czech the main factor is the degree of communicative dynamism (CD). At DL, the surface differences should be factored out and the various means of expressing the scale of CD across the two languages translated into a uniform representation.

Theory of FGD maintains that the nodes in the underlying dependency representation of a sentence are ordered from left to right according to the rising degree of CD. According to a recent version of the theory, this ordering is total for sister nodes. This solution has the merit in that it preserves projectivity (non-crossing dependency links), but, on the other hand, is not able to represent some intuitive distinctions in CD. The two Czech sentences in (1) and (2) illustrate this difficulty.

(1) Tu knížku Pavel slíbil vrátit do zítřka.
the book Pavel promised to return till tomorrow

(2) Pavel slíbil tu knížku vrátit do zítřka.
Pavel promised the book to return till tomorrow.

In both cases, the nodes of the tree are ordered in the same way, as shown in the following dependency tree:

```
      ___slíbil______
     /               \
 Pavel            __vrátit__
                 /          \
         tu knížku     do zítřka
```

Even though the unexpressed subject of the imbedded verb 'Pavel' can be ordered either after or before the object 'the book' to represent the distinction, the object of the embedded verb still has to follow the main verb in order to save the projectivity requirement. This does not reflect the intuition that the degrees of CD for 'promised' and 'Pavel' are related in both cases. The situation is similar in English examples like 'The fence has been difficult to break':

```
   difficult
        \
         break
        /
 fence
```

If, however, the ordering of nodes should be defined over the whole sentence, including embedded structures, and if a member of an embedded structure like 'the book' or 'the fence' is treated as the least dynamic element of the whole clause, the tree is no longer projective. If the projectivity condition is to be retained and all elements of the sentence ordered, including elements of embedded structures, the degrees of communicative dynamism have to be reflected independently of the ordering of nodes in the dependency tree.

A proposal along these lines might be based on the introduction of a separately specified dimension of the DL representation, a new attribute CD. In the current system, the value of CD is a list of references to items participating in the CD hierarchy.

Several problems remain, most notably the problem of relating a surface string with this list in more complex cases, but the implementantion attests to the viability of this approach.

3 Specifics of the monolingual parts

3.1 Specifics of English

The lexicon for English has been compiled from several electronic sources with a considerable degree of human involvement and semi-automatic processing. The number of lemmas in the canonical format conforming to the ELU formalism is currently slightly over 100.000. From this large file, 20.000 more frequent entries have been tested and compiled and are available for parsing and generation.

The core of English syntax is covered by 26 grammar rules. The covered phenomena include analytical verb forms, relative clauses, *raising* and *equi* embedded predicates, unbounded dependencies (including embedded relative expressions and *tough/easy* predicates), and coordination of phrasal and lexical categories.[6]

3.2 Specifics of Czech

Czech is different from English in a number of aspects, especially in its rich morphology and the relatively free word order. These two properties alone disqualify a number of standard formal approaches and software tools. One of the goals of the present enterprise was to find an answer to the following qustion: can the same processing tool, formalism and theory be used for Czech as for English?

3.2.1 The treatment of free word order

Within unification-based context-free framework, solutions for languages with a significant amount of word-order variations (mostly for German) have been proposed, i.a., by Uszkoreit (1987), Reape (1989), Oliva (1992), and Hinrichs & Nakazawa (1993). The central issue at stake is the problem of whether the rules should license syntactic constituents in the traditional sense (as in Reape) or merely reflect a formal strategy. This is closely related to the issue of whether a special mapping between surface string and leaves is necessary. Other decisions to be made concern the geometry of the derivation tree: possible restrictions on the number of sister nodes and the direction of branching. Another issue is the way constraints on word order are imposed.

As in the case of English, the description of Czech is based on a combination of context-free rules with a dependency-based representation within a monostratal unification formalism. However, the context-free rules have very little linguistic content in the sense that the geometry of the derivation tree is determined by a formal strategy rather than by considerations of purely linguistic nature, such as which constituents form a more cohesive unit.

The present approach is based on binary bidirectional branching structure. This way, a standard formalism and processing tool can be employed and a straightforward transparent derivational history maintained. On the other hand, constraints on word order have to be specified in a more complex fashion.

[6] Furthermore, the grammar handles phenomena like Complex NP Constraint (a relative clause cannot include a trace), Trace Principle (every trace must be subcategorized, but not as a subject), Subject Condition (a trace may occur inside a subject only when it is subcategorized by another item in the SC list), Coordination Principle ('strong version' – identity of NONLOC values in conjuncts and the mother).

3.2.2 Lexicon and morphology-related issues

For Czech, no electronic source of lexical data has been used,[7] but massive augmentation of the lexical component is previewed as soon as a suitable source is available. So far, the strategy was to verify those parts of lexical description which are not lexeme-specific: the definition of inflectional paradigms and the definition of more general classes specifying valency requirements and data structure.

The power of default inheritance hierarchy has been employed also in the specification of inflectional paradigms. Ideally, such a description of morphology should profit from any case of similarity across inflectional paradigms. Thus, the plural instrumental ending *-mi*, which is realized in most declension paradigms, could be specified only once for the whole class of the paradigms. Other endings can stand for more cases within a single declension paradigm, as well as for a specific case across multiple paradigms. Such an ending should be assigned a local disjunction of morphological categories, as well being specified for more paradigms. Here, however, the inheritance approach as represented in ELU invites a problem of multiplicity of structures, which is made even worse by the fact that variant endings are possible for some combinations of morphological categories within a paradigm[8] and the impossibility to use disjunction over other elements besides atoms.

3.2.3 Grammar rules

The grammar describes the core of syntactic phenomena, preserving the methods used for English where useful. The strategy was to concentrate on the differences between English and Czech syntax, in order to verify viability of the approach.

All rules are binary, except for the rule S, which is necessary for the description of the unexpressed subject. Three rules specify a fixed precedence of daughters: the rule for prepositional marking (PP), adjectival modification and modification of noun by a prepositional phrase. Other rules occur in two versions, reflecting the two possible orderings of daughters: head-final or head-initial (these are rules for identifying a verb's complement – CV and VC, adjunct – AV and VA, and an auxiliary verb – AuxV and VAux).[9] Starting upwards from the lexical head, the lower head projections first include all modifiers (complements or adjunct) standing to the right (if any), while the higher projections include modifiers to the left (if any). This means that in generation of a single head domain, head-final rules are applied before head-initial rules. For parsing, the order is reversed. The bracketed sentences below show derivation trees for 6 word order variants of the sentence *Honza dal Marii pusu* — 'John gave Mary a kiss'.

$[_{s}\ [_{cv}$ Honza $[_{vc}\ [_{vc}$ dal Marii] pusu]]]	$[_{s}\ [_{cv}$ pusu $[_{vc}\ [_{vc}$ dal Honza] Marii]]]
$[_{s}\ [_{cv}$ Honza $[_{cv}$ pusu $[_{vc}$ dal Marii]]]]	$[_{s}\ [_{cv}$ pusu $[_{cv}$ Marii $[_{vc}$ dal Honza]]]]
$[_{s}\ [_{cv}$ pusu $[_{cv}$ Honza $[_{cv}$ Marii dal]]]]	$[_{s}\ [_{vc}\ [_{vc}\ [_{vc}$ dal Honza] Marii] pusu]]

[7] The hand-coded description of lemmas has slightly over 100 representative entries.

[8] Variant forms, however, should not be specified as atomic disjuncts, but rather as marked and unmarked due to the need to select a specific form in generation.

[9] The abbreviations of the rules will be used below.

3.2.4 The treatment of clitics

There is a class of (arguably) sentential clitics in Czech which tend to occur at what has traditionally been called Wackernagel's position, i.e. the second position following first unit in a clause.[10] Some prominent members of this class are: past tense auxiliary (*jsem*) , reflexive particles (*se, si*), weak dative and accusative personal pronouns (*mu, ho*). If more of them are present, they form a cluster and obey the order given above. Clitics from embedded (mostly infinitival) clauses may climb to the second position of the main clause, but may also stay within the embedded clause. If two or more clitics of the same type should meet in the same cluster as a result of climbing, all but one are obligatorily deleted.

To control word-order phenomena of this type, which are combined with 'topological' requirements (second position), within a binary branching context-free framework, the virtually wide-scope constraints have to be translated into local ones, not extending beyond binary-branching rules. This approach has been used previously e.g. in Engelkamp et al. (1992), Oliva (1992). An additional complexity is added due to the fact that the grammar is both left and right branching. In effect, every constraint has two versions: one for the head-final rules, one for the head-initial rules. The essence of the solution is in a combination of two types of constraints: (i) constraints controlling the second position as a possible location for a clitic cluster, e.g. a constraint requiring that a clitic may not occur as a non-head daughter in a continuous sequence of head-final local trees from a maximal verbal projection downwards, (ii) constraints controlling the sequence of clitics in a cluster; such constraints specify requirements which every non-head daughter at lower projection levels must satisfy once the non-head daughter at the current level is a clitic of a certain type. These constraints are implemented by macros which control the interaction of several dedicated features.

The following bracketed examples show the structures used for clauses including clitical elements. Cases where a non-clitic intervenes in a clitic cluster or the order of elements within the cluster is incorrect are correctly refused.

$[_{s}$ $[_{av}$ $[_{pp}$ na poli] $[_{auxv}$ jsem $[_{cv}$ mu $[_{cv}$ ho nedal]]]]]
$[_{s}$ $[_{va}$ $[_{vc}$ $[_{vc}$ $[_{vaux}$ nedal jsem] mu] ho] $[_{pp}$ v divadle]]]
$[_{s}$ $[_{cv}$ jemu $[_{auxv}$ jsem $[_{cv}$ ho nedal]]]]
$[_{s}$ $[_{av}$ $[_{pp}$ na poli] $[_{auxv}$ jsem $[_{cv}$ se třásl]]]]
* na poli se jsem třásl, * nedal jsem jemu ho v divadle

4 Transfer

The two languages meet in the transfer component. The DL-relevant part of representation (value of the attribute DEEP) of a sentence is compositionally related to its equivalent in the other language by the specifications in the transfer component.

The ELU transfer component is an extension of the basic constraint-based scheme, preserving locality and declarativity (therefore bi-directionality) of the

[10] The nature of the first unit is a matter of argument. In the simplified solution adopted here, the first unit is taken to be a syntactic constituent.

formalism. The transfer rules are of two types, corresponding to the transfer of individual lexical items and the transfer of structures. Recursion into the transferred structures is controlled by a general mechanism of subsumption between the structure and the rules, and by specificity ordering among transfer rules.

The following example shows two lexical transfer rules and two structural transfer rules. The structural rule `general_gov_deps` works generally for any governor and its dependents (equating relevant values in the representations of the two languages), unless a more specific rule like `líbit_se_like` is found, which switches the order of the first two arguments, doing justice to the fact that 'John likes Mary' is translated into Czech as 'Honzovi se líbí Marie', where 'Marie' is the subject and – arguably – the first argument.

The scale of communicative dynamism in the shape of a list of references to the content words in the clause is not altered by the transfer relation, making the selection the correct Czech word order possible.[11]

```
:TA: Tomáš Tom
:TA: Zuzana Sue

:T:  general_gov_deps
:L1: <gov> = GovCz    <deps> = DepsCz
:L2: <gov> = GovEng   <deps> = DepsEng
:X:  GovCz = GovEng   DepsCz = DepsEng

:T:  líbit_se_like
:L1: <gov lexeme> = líbit_se
     <deps> = [Arg1Cz Arg2Cz RCz]
:L2: <gov lexeme> = like
     <deps> = [Arg1Eng Arg2Eng RCz]
:X:  Arg1Cz = Arg2Eng
     Arg2Cz = Arg1Eng
     RCz = REng
```

5 Conclusion

The effort has been fruitful in the following points:

- A substantial re-usable lexical data base for English has been accumulated.
- A common lexically based approach, combining context-free grammar with dependency-oriented representation has been verified for two typologically different languages in the process of parsing and generation.
- The viability of the treatment of Czech morphology within a hierarchically structured lexical component has been demonstrated.
- A number of theoretically relevant solutions to various phenomena have been implemented and tested with satisfactory results.
- The whole model has proved to be flexible and open to further development as a kind of linguistic testbed.

[11] E.g. 'Marie se líbí Honzovi' would be appropriate as the translation of 'It is John who likes Mary'.

The plans include, besides a continuing grammar development and refinement of the existing lexical data, a radical extension of the transfer component and of the Czech lexicon. Also, the whole system would profit from re-implementation in a more sophisticated framework. Other language modules could be incorporated to provide more feedback to solutions previewed as suitable across more languages

In a more distant perspective, the rule-based description of linguistic competence for the two languages should be augmented with a model of simulated competence and performance knowledge, based on data extracted from corpora by statistical techniques.

References

Estival, D. (1990), *ELU User Manual* Technical Report 1, ISSCO, Geneva.

Engelkamp, J., G. Erbach and H. Uszkoreit (1992) Handling Linear Precedence Constraints by Unification. In *Proceedings of ACL '92*, Newark.

Hajič, J., E. Hajičová & A. Rosen (1992) "Machine Translation Research in Czechoslovakia", *META* 1992, vol. 37, No. 4 — a special issue, eds. M. C. Cormier and D. Estival, Presses de l'Université de Montréal, pp. 802–816, Montréal, 1992.

Hinrichs, E. & Ts. Nakazawa (1993) *Aspects of German VP Structure – An HPSG Account.* SfS–Report–01-93, Seminar für Sprachwissenschaft, Computerlinguistik, Eberhard-Karls-Universität, Tübingen.

Kirschner Z. & A. Rosen (1989) "APAC – An Experiment in Machine Translation", *Machine Translation* 1989, 4:177–193.

Oliva, K. (1992) 'The proper treatment of word order in HPSG.' In *Proceedings of COLING 1992*, Nantes.

Pollard, K., and I. Sag (1994) *Head-Driven Phrase Structure Grammar.* Stanford: Center for the Study of Language and Information and Chicago & London: The University of Chicago Press.

Reape, M. (1989) 'A Logical Treatment of Semi-Free Word Order and Bounded Discontinuous Constituency'. *Proceedings of the 4th Meeting of the European Chapter of the ACL.* Manchester.

Russell G., A. Ballim, J. Carroll, S. Warwick-Armstrong (1992) 'A Practical Approach to Multiple Default Inheritance for Unification-Based Lexicons', Special Issue on Inheritance II, *Computational Linguistics*, Vol. 18, No. 3, 311-337.

Sgall P., E. Hajičová and J. Panevová (1986) *The Meaning of the Sentence in its Semantic and Pragmatic Aspects* (edited by J. Mey), Reidel, Dordrecht / Academia, Praha.

Uszkoreit, H. (1987). *Word Order and Constituent Structure in German.* CSLI Lecture Notes No. 8. CSLI, Stanford.