Multilingual Grammar Resources in Multilingual Application Development

Marianne Santaholma
Geneva University, ETI/TIM/ISSCO
40, bvd du Pont-d'Arve
1211 Geneva 4, Switzerland
Marianne.Santaholma@eti.unige.ch

Abstract

Grammar development makes up a large part of the multilingual rule-based application development cycle. One way to decrease the required grammar development efforts is to base the systems on multilingual grammar resources. This paper presents a detailed description of a parametrization mechanism used for building multilingual grammar rules. We show how these rules, which had originally been designed and developed for typologically different languages (English, Japanese and Finnish) are applied to a new language (Greek). The developed shared grammar system has been implemented for a domain specific speech-to-speech translation application. A majority of these rules (54%) are shared amongst the four languages, 75% of the rules are shared for at least two languages. The main benefit of the described approach is shorter development cycles for new system languages.

1 Introduction

Most of grammar based applications are built on monolingual grammars. However, it is not unusual that the same application is deployed for more than one language. For these types of systems the monolingual grammar approach is clearly not the best choice, since similar grammar rules are written several times, which increases overall development efforts and makes maintenance laborious.

One way to decrease these efforts is to share already developed linguistic resources between system languages. Common approaches for sharing information include grammar adaptation and grammar sharing. Grammar adaptation is the technique of modifying an already existing grammar to cover a new language as implemented among others by Alshawi et al. 1992; Kim et al. 2003; and Santaholma, 2005.

In grammar sharing, existing grammar rules are directly shared between languages rather than just being recycled as they are in grammar adaptation. Compared to both the monolingual grammar approach and the grammar adaptation approach, grammar sharing reduces the amount of code that needs to be written as the central rules are written only once. This automatically leads to coherence between language descriptions for different languages, which improves grammar maintainability, and eliminates the duplication effort that otherwise occurs when monolingual grammars are used.

Multilingual grammars can share resources between languages in various ways. Ranta (2007) has developed an abstract syntax that defines a common semantic representation in a multilingual grammar.

Another type of approach is implemented in the LinGO Grammar Matrix project (Bender et al. 2005; Bender, 2007). The Grammar Matrix consists of a core grammar that contains the types and constraints that are regarded as cross-linguistically useful. This core is further linked to phenomenon-specific libraries. These consist of rule repertories based on typological categories. The necessary modules are put together like building blocks according to language characteristics to form the final grammar of a language.

The work described in this paper implements

Coling 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks, pages 25–32
Manchester, August 2008

a grammar sharing approach that is based on language-independent parameterized rules that are complemented with necessary language-specific rules and features. These shared rules have been implemented for MedSLT, a multilingual spoken language translation system (Bouillon et al., 2005). All of the central language processing components of MedSLT, including the speech recognizer, parser and generator, are derived from handcrafted general grammars of a language. The biggest effort in adding a new language to the existing spoken language translation framework is the grammar development cycle. As more languages are added to the existing spoken language translation framework, the necessity for multilingual grammar rules grows.

(Bouillon et al., 2006) first developed shared MedSLT grammar rules for the Romance languages French, Spanish and Catalan. Compared to the monolingual grammar system, the shared grammar-based system facilitated application development without degrading the performance of any of its components (speech recognition, translation) on these languages.

We took this approach further and implemented parameterized grammar rules for typologically different languages - English, Finnish and Japanese. Experiments have shown that these shared rules perform equally well on all three languages (Santaholma, 2007). As these grammars had been developed in parallel, it was not clear how flexible the parameterized grammar approach would be for new a language, which was not included in the original development process. We thus extended the grammar to cover Modern Greek as a new language. The paper describes the methodology of adding this new language and evaluates the parametrization mechanism.

The rest of the paper is structured as follows. Section 2 describes the Regulus toolkit (Rayner et al., 2006) and MedSLT, which form the development environment and application framework on which this work is based. Section 3 describes the parameterized multilingual grammar and parametrization mechanism. Section 4 summarizes techniques used to adding Modern Greek to the shared grammar system. Section 5 concludes.

2 Regulus Development environment and MedSLT application

2.1 Regulus features

The Regulus grammar framework has been designed for spoken language grammars, and thus differs from popular grammar frameworks like Lexical Functional Grammar (LFG) (Bresnan and Kaplan, 1985) and Head-driven Phrase Structure Grammar (HSPG) (Pollard and Sag, 1994). Regulus grammars are written with a feature-grammar formalism that can easily be compiled into context free grammars (CFG). These are required for compilation of grammar-based language models used in speech recognition. Characteristic for Regulus grammars are finite valued features and exclusion of complex feature-value structures. Consequently the resulting rule-sets are perhaps more repetitive and less elegant than the equivalent LFGs or HPSGs. This design, however, enables compilation of non-trivial feature-grammars to CFG.

Another Regulus feature that enables CFG compilation is grammar specialization that reduces the extent of the grammar. Grammar specialization is performed by explanation-based learning (EBL) [1]. Multilingual grammar development can profit from grammar specialization in various ways. The general grammar of a language can be specialized to specific domains based on domain specific information[2]. Thus the specialization serves as a way to limit the ambiguities typical for general grammars. Furthermore, the procedure is used to specialize the grammars for different tasks. Ideally a grammar should recognize variant forms but generate only one. This variation can be controlled by specializing the Regulus grammars for these tasks. Finally the multilingual Regulus grammar can be specialized for specific languages by automatically removing the unnecessary rules.

2.2 MedSLT

Most large-scale machine translation systems are currently based on statistical language processing. MedSLT, however, has been implemented with linguistically motivated grammars mainly for the following reasons: (1) necessary data for inducing the grammars and training the statistical language

[1] The method is described in detail in (Rayner et al., 2006), Chapter 10.

[2] These include domain specific corpus, lexica and operationality criteria that control the granularity of specialized grammar. Details provided by (Rayner et al., 2006).

models were not available for the required domain and languages. (2) the medical domain demands accurate translation performance, which can be more easily guaranteed with rule based systems.

MedSLT is an unidirectional[3] speech-to-speech translation system that has been designed to help in emergency situations where a common language between the physician and the patient does not exist. In addition to influencing the system architecture, this communication goal also significantly influences system coverage, and consequently the grammars. The typical utterances MedSLT translates consist of physician's questions about the intensity, location, duration and quality of pain, factors that increase and decrease the pain, therapeutic processes and the family history of the patient. These include yes-no questions like "Does it hurt in the morning?", "Is the pain stubbing?" and "Do you have fever when you have the headaches?". Other frequent type of questions include wh-questions followed by elliptical utterance, like "Where is the pain?", "In the front of the head?", "On both sides of the head?". Currently MedSLT translates between Arabic, Catalan, English, Finnish, French, Japanese, and Spanish. The translation is interlingua based.

The following sections describe the implementation of the shared parameterized grammar rules for this specific application using the Regulus platform.

3 Parameterized grammar rules

The parameterized grammar rules assemble the common foundations of linguistic phenomena in different languages. The framework for the language-independent rules presented here was developed and tested with English, Japanese and Finnish. These languages represent different types of languages and hence express the same linguistic phenomena in different ways. Consequently they provided a good starting point for framework design.

The Regulus multilingual grammar is modular and organized hierarchically. Parameterized rules are stored in the "language-independent core" module. This is the most generic level and as such is shared between all languages. The "lower levels" include the language-family specific modules and the language-specific modules. The modules for related languages decrease redundancy as related languages commonly share characteristics at least to some extent. [4] The information in this modular structure is inherited top-down from the most generic to language specific.

The first language to which we applied the parameterized rules and which had not been part of the original shared grammar framework development is Modern Greek.

In the following we first describe the parameterized grammar rules. Then we focus on how these rules are applied for Greek.

3.1 Coverage

The parameterized grammar currently covers basic linguistic phenomena by focusing on the structures required to process MedSLT system coverage. The current coverage is summarized in Table 1.

Phenomena	Construction
Sentence types	declarative, yn-question, wh-question, ellipsis subordinate when clause
Tense	present, past(imperfect)
Voice	active, passive
Aspect	continuous, present perfect, past perfect
Verb subcategorization	transitive, intransitive, predicative (be+adj), existential (there+be+np)
Determiners	article, number, quantifier
Adpositional modifiers	prepositional, postpositional
Adverbial modifiers	verb and sentence modifying adverbs, comparison adverbs
Pronouns	personal, possessive, dummy pronouns
Adjective modifiers	predicative, attributive, comparison

Table 1: Linguistic phenomena covered by the shared grammar.

The general difficulty of spoken language for grammar development is frequent ungrammatical

[3] Bidirectional MedSLT exists currently for English-Spanish language pair. Details are provided in (Bouillon et al., 2007).

[4] However, as identical constructions and features also exist in unrelated languages the advantage of language family modules is finally not so significant.

and non-standard use of language. This includes for example incorrect use of case inflections in Finnish and missing particles in spoken Japanese.

3.2 Parametrization - abstracting away from language specific details

The parametrization aims to generalize the cross-linguistic variation in grammar rules. English yes-no questions require an auxiliary and inverted word order, in Finnish yes-no questions the subject-verb inversion is combined with a certain form of the main verb; in Finnish noun heads and the modifying adjective agree in case and number, in Greek they additionally agree in gender, and so forth. The way of expressing the same linguistic phenomena or constructions varies from one language to another. Hence, shared grammar rules need to abstract away from these kinds of details.

The multilingual Regulus rules are parameterized using macro declarations. Macros are a standard tool in many development environments. In Regulus grammars they are extensively used to catch generalizations in the rules, and in particular in lexica. In multilingual grammar rules the macros serve as "links" towards language-specific information.

The shared rules have a language-neutral surface representation where macros invoke the required language-specific information. The macro reference of a language-independent rule is replaced by the information contained in the macro definition. The context of the macro reference determines how the macro definition combines with other parts of the description. The mechanism is similar to LFG 'templates', which encode linguistic generalizations in a language description (Dalrymple et al., 2004).

The macro mechanism itself is rather simple. The crucial is that the macros are defined in a transparent and coherent way. Otherwise the grammar developer will spend more time learning to how to use the parameterized rule set than she would spend to develop a new grammar from scratch. When the macros are well defined, sharing the rules for a new language is just a matter of defining the language-specific macro definitions.

In the following we present some concrete examples of how cross-linguistic variation can be parameterized in a multilingual Regulus grammar using macros.

3.2.1 Parameterizing features

The following example shows how we parameterize the previously mentioned agreement features required in different languages. In Regulus grammars, like in other constraint-based grammars, this fine-grained information is encoded in feature-value pairs. We encode a basic declarative sentence rule (`s`) that consists of a noun phrase (`np`) and a verb phrase (`vp`):

```
s:[sem=concat(Np, Vp)] -->
  np:[sem=Np, sem_np_type=T,
  @noun_head_features(Head)],
  vp:[sem=Vp, subj_sem_np_type=T,
  @verb_head_features(Head)].
```

In Finnish sentences the subject and the main verb agree in person and number. Japanese doesn't make use of these agreement features in this context. Consequently, the common rules have to express the agreement in a parameterized way. For this reason in the `np` we introduce a macro called `noun_head_features(Head)` and in the `vp` the macro `verb_head_features(Head)`. [5] These macro declarations unify but don't say anything explicit about the unifying features themselves at this common level. The macros thus "neutralize" the language-specific variation and only point further down to language-specific information.

In Finnish, the `noun_head_features` and `verb_head_features` macros invoke the language specific features 'number' and 'person':

```
macro
(noun_head_features([P, N]),
[person=P, number=N]).

macro
(verb_head_features(([P, N]),
[person=P, number=N]).
```

The macro references are replaced by these features in the final Finnish declarative sentence rule which takes the form:

```
s:[sem=concat(Np, Vp)] -->
  np:[sem=Np, sem_np_type=T,
  person=P, number=N],
  vp:[sem=Vp, subj_sem_np_type=T,
  person=P, number=N].
```

[5] The Regulus macro declaration is preceded by '@'.

As Japanese does not apply either 'number' or 'person' features the macro definition consists of an empty value:

```
macro(noun_head_features([]),
[]).
```

The final Japanese sentence rule takes after the macro replacement the form:

```
s:[sem=concat(Np, Vp)] -->
  np:[sem=Np, sem_np_type=T],
  vp:[sem=Vp, subj_sem_np_type=T].
```

Similarly we can parameterize the value of a specific feature. A `vp` could include a `verb_form` feature that in English could take as its value "gerundive" and in Finnish "infinite" in that particular context. We can parameterize the `vp` rule with a macro `vform` which invokes the language-specific macro definition and replaces it with the corresponding language-specific feature-value pairs:

```
vp:[sem=concat(Aux, Vp)] -->
 aux:[sem=Aux,@aux_features(Head)],
 vp:[sem=Vp,
  @vform(Vform),
  @verb_head_features(Head)].
```

The English macro definition would be:

```
macro(vform(Vform),
[verb_form=gerund,
 verb_form=Vform]).
```

The Finnish equivalent:

```
macro(vform(Vform),
[verb_form=finite,
 verb_form=Vform]).
```

Macros can furthermore refer to other macro definitions and in this way represent inclusion relations between different features. This forms a multilevel macro hierarchy. The macro `noun_head_features(Head)` included in `np` rule (1) could contain a macro `arg` (2), that would further be defined by (3):

```
1)
np:[sem=Np, sem_np_type=SemType,
  @noun_head_features(Head)].

2)
macro(noun_head_features([Agr,Case]),
  [@agr(Agr), case=Case]).

3)
macro(agr([Case, Number]),
  [case=Case, number=Number]).
```

3.2.2 Parameterizing the constituent order

The constituent order is defined by concatenation of linguistic categories in the wanted order (`vp:[sem=concat(Verb, NP)]`). This order can, similarly to features, also be parameterized by using macros. We show here as an example of how the order of a transitive main verb (`verb`) and direct object (`np`) is parameterized in a verb phrase:

```
vp:[sem=concat(Verb, NP)] -->
verb:[sem=Verb, subcat=transitive,
     obj_sem_np_type=ObjType],
np:[sem=NP, sem_np_type=ObjType]).
```

In English the direct object follows the verb, whereas in Japanese it precedes the verb. The order of these constituents can be parameterized by introducing into the rule a macro that in the example rule is represented by 'verb_transitive_np':

```
vp:[sem=concat(Verb, NP)] -->
@verb_transitive_np(
verb:[sem=Verb, subcat=transitive,
      obj_sem_np_type=ObjType],
np:[sem=NP, sem_np_type=ObjType]).
```

This macro invokes the language-specific rules that define the order of the semantic values of categories required in a specific language. The semantic value of the category verb is `sem=Verb` and of noun `sem=Noun`. Consequently the English-specific macro definition would be:

```
macro(verb_transitive_np
(Verb, Noun),(Verb, Noun)).
```

This rule specifies that when there is a semantic value 'Verb' followed by a semantic value 'Noun' these should be processed in the order 'Verb', 'Noun'. The order of constituents remains unchanged.

The equivalent Japanese macro definition would be:

```
macro(verb_transitive_np
(Verb, Noun),(Noun, Verb)).
```

Contrary to the English rule this rule specifies that when there is a semantic value 'Verb' followed by a semantic value 'Noun' these should be processed in the order 'Noun', 'Verb'. This changes the order the of constituents. Details of Regulus semantic processing are available in Rayner et al., 2006.

3.2.3 Ignoring rules/features and using empty values

There exist several ways to ignore rules and features or to introduce empty values in Regulus grammars. These have proven practical in rule parametrization. In the following we present some frequent examples.

Features that are irrelevant for a particular language (in a particular context) can take 'empty' ([]) as their value. This can be encoded in several ways.

- Macro takes an empty value. This is encoded by '[]'

  ```
  Example:
  macro(noun_head_features([]),
  []).
  ```

- Feature takes an empty value. This is encoded by '_':

  ```
  Example:
  macro(premod_case(Case),
  [case=_]).
  ```

Rules that are applied to only one language are organized in the language-specific modules. However most of the rules are necessary for two or more languages. The rules that are used for groups of specific languages can be 'tagged' using macro declarations. For example a rule or feature that is valid for English and Japanese could be simply tagged with an identifier macro 'eng_jap':

```
@eng_jap
('rule_body_here').
```

The English and Japanese rules would call the rule body by macro definition:

```
macro(eng_jap(Body), (Body).
```

The Finnish language-specific macro definition would call an empty category that we call here 'dummy_cat' and the rule would be ignored:

```
macro(eng_jap(Body),
(dummy_cat:[] --> dummy)).
```

Specialization of a grammar for a specific language and into domain-specific form checks which rules are necessary for processing the domain specific-coverage in that particular language. Consequently empty features of the general grammar are automatically ignored and the language processing remains efficient.

4 Processing Modern Greek with shared parameterized grammar rules

Cross-linguistic comparison shows that the Greek that belongs to the Indo-European language family does not only share some features with English but also with Japanese and Finnish. Common with English is, for example, the use of prepositions and articles, and with Finnish and Japanese the pro-drop.

The development of Greek grammar coverage equivalent to those of English, Japanese and Finnish coverage in MedSLT took about two weeks. For most part only the language-specific macro definitions needed to be specified. Five new rules were developed from scratch. The most significant part of the development consisted of building the Greek lexicon and verifying that the analyses produced by the shared grammar rules were correct.

In the following we summarize Greek-specific rules, features and macros.

4.1 Greek rules and features

In general, Greek word order is flexible, especially in spoken language. All permutations of ordering of subject, object, and verb can be found, though the language shows a preference for Subject-Verb-Object ordering in neutral contexts. New parametrized *constituent orders* were the most significant additions to the multilingual grammar. These are listed below.

1. Yes-no questions, which are a central part of the MedSLT application's coverage, can be expressed by both direct and indirect constituent order in Greek. As these are both common in spoken language, the Japanese question rule (direct constituent order + question particle 'ka') was parameterized for Greek.

2. The order of possessive pronoun and head noun required parametrization. Until now the shared grammar contained only the order where a head noun is preceded by the possessive. In Greek the opposite order is used, with the possessive following the head noun. The existing rule was parameterized by a new macro.

3. Similar parameterization was performed for verb phrases including an indirect object. The Greek constituent order is reversed relative to English order. That is, the pronoun goes before the verb. A new macro was introduced to parameterize the rule.

One main area of difference compared to English/Finnish/Japanese, is in the placement of weak pronouns, generally referred to as 'clitics'. Their position in Greek is relative to the verb. In standard language they are placed before finite verbs and after infinite verbs. Thus these weak pronouns can occur in sentence-initial position. New rules were developed to process these clitics as well as the Greek genitive post-modifier structure.

Greek could mainly use the existing grammar *features*. The difference, compared to the original three languages, was in the extensive use of the 'gender' feature (possible values: feminine, masculine and neuter). For example, Greek articles agree with the head noun in gender, number, and case. Furthermore, prepositions agree with the following nouns in gender, number and case.

4.2 Summary of multilingual rules

Table 2 summarizes current use of the multilingual rules. The grammar includes a total of 80 rules for English, Finnish, Japanese and Greek. 54% of the rules are shared between all four languages and 75% of the rules are shared between two or more languages. Not everything can be parameterized, and some language-specific rules are necessary. The language-specific rules cover 25% of all rules.

Languages	N. of rules	% of total
Eng + Fin + Jap + Gre	43	54%
Eng + Fin + Jap	0	
Eng + Fin + Gre	4	
Eng + Jap + Gre	0	
Fin + Jap + Gre	6	
TOTAL	10	12.5%
Fin + Jap	3	
Eng + Fin	1	
Eng + Jap	1	
Gre + Eng	1	
Gre + Jap	1	
Gre + Fin	0	
TOTAL	7	8.75%
Eng	9	
Fin	0	
Jap	6	
Gre	5	
TOTAL	20	25%
TOTAL	80	100%

Table 2: Grammar rules in total

5 Conclusions

We have described a shared grammar approach for multilingual application development. The described approach is based on parametrization of Regulus grammar rules using macros. We have shown that these parameterized rules can with comparably little effort be used for a new system language in a multilingual limited-domain application. A majority of rules were shared between all implemented languages and 75% of rules by at least two languages. The deployment of a new language was mainly based on already existing rules.

The shared grammar approach promotes consistency across all system languages, effectively increasing maintainability.

Acknowledgement

I would like to thank Pierrette Bouillon and Manny Rayner for their advise, and Agnes Lisowska and Nikos Chatzichrisafis for their suggestions and English corrections.

References

Bender, Emily and Dan Flickinger. 2005. *Rapid Prototyping of Scalable Grammars: Towards Modularity in Extensions to a Language-Independent Core*. In: Proceedings of the 2nd International Joint Conference on Natural Language Processing IJCNLP-05 (Posters/Demos), Jeju Island, Korea.

Bender, Emily. 2007. *Combining Research and Pedagogy in the Development of a Crosslinguistic Grammar Resource*. In: Proceedings of the workshop Grammar Engineering across Frameworks 2007, Stanford University.

Bouillon, Pierrette, Manny Rayner, Nikos Chatzichrisafis, Beth Ann Hockey, Marianne Santaholma, Marianne Starlander, Yukie Nakao, Kyoko Kanzaki, Hitoshi Isahara. 2005. *A generic multilingual open source platform for limited-domain medical speech translation*. In: Proceedings of the 10th Conference of the European Association for Machine Translation, EAMT, Budapest, Hungary.

Bouillon, Pierrette, Manny Rayner, Bruna Novellas Vall, Marianne Starlander, Marianne Santaholma, Nikos Chatzichrisafis. 2007. Une grammaire partage multi-tache pour le traitement de la parole : application aux langues romanes. *TAL (Traitement Automatique des Langues), Volume 47, 2006/3*.

Bouillon, Pierrette, Glenn Flores, Marianne Starlander, Nikos Chatzichrisafis, Marianne Santaholma, Nikos Tsourakis, Manny Rayner, Beth Ann Hockey. 2007. *A Bidirectional Grammar-Based Medical Speech Translator*. In: Proceedings of workshop on Grammar-based approaches to spoken language processing, ACL 2007, June 29, Prague, Czech Republic.

Bresnan, Joan and Ronald Kaplan. 1985. *The mental representation of grammatical relations*. MIT Press, Cambridge, MA.

Butt, Miriam , Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. *The Parallel Grammar Project*. In: Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation.

Dalrymple, Mary, Ron Kaplan, and Tracy Holloway King. 2004. *Linguistics Generalizations over Descriptions*. In M. Butt and T.H. King (ed.) Proceedings of the LFG04 Conference.

Kim, Roger, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, Hiroshi Masuichi, Tomoko Ohkuma. 2003. *Language Multilingual Grammar Development via Grammar Porting*. In: Proceedings of the ESSLLI Workshop on Ideas and Strategies for Multilingual Grammar Development, Vienna, Austria.

Pollard, Carl and Ivan Sag. 1994. *Head Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Ranta, Aarne. 2007. Modular Grammar Engineering in GF. *Research on Language and Computation, Volume 5, 2/2007, 133–158*.

Rayner, Manny, Beth Ann Hockey, Pierrette Bouillon. 2006. *Regulus-Putting linguistics into speech recognition*. CSLI publications, California, USA.

Santaholma, Marianne. 2005. *Linguistic representation of Finnish in a limited domain speech-to-speech translation system*. In: Proceedings of the 10th Conference on European Association of Machine Translation, Budapest, Hungary.

Santaholma, Marianne. 2007. *Grammar sharing techniques for rule-based multilingual NLP systems*. In: Proceedings of NODALIDA 2007, the 16th Nordic Conference of Computational Linguistics, Tartu, Estonia.