[From: *CLAW '96*, Universiteit Leuven]

Controlled language correction and translation

Pim van der Eijk, Michiel de Koning, Gert van der Steen
Cap Volmac Advanced Technology Services
Daltonlaan 400
Postbus 2575, 3500 GN Utrecht.
{pvdeijk,mkoning,gsteen}@inetgate.capvolmac.nl

Abstract

Controlled languages are constructed languages that have precise coverage bounds and are designed to satisfy linguistic constraints such as greatly reduced ambiguity. Cap Volmac has been developing software and linguistic resources for controlled language processing. In this paper, we consider the characteristics of controlled languages as opposed to a related concept, sublanguages, the process of designing controlled language systems. We illustrate the correction mechanism, and its integration in a document creation environment, with examples from Simplified English. Finally, we discuss an extension of the use of controlled languages to machine translation.

1. Introduction

Since the early 1990s, Cap Volmac Lingware Services (currently part of Cap Volmac's Advanced Technology group, Utrecht, Netherlands) has been developing software to support large scale document creation and translation using controlled sublanguages, and deploying this software in customer projects. From its inception, activities have concentrated on controlled languages satisfying severe lexical and syntactic restrictions, such as on-line help texts, software manuals, and aerospace maintenance manuals.[1]

The approach to development of linguistic descriptions was motivated initially by the practical concern that an approach based on fine-tuning a general system for unrestricted texts to derive specific applications would be unnecessarily complex and expensive to develop.[2] A related motivation, based on results of sublanguage analysis (cf. section 2), is that the language to be processed for a particular application is best described as an independent linguistic system, that has its own internal coherence, rather than being a specialized version of standard language. Controlled sublanguages are derived variants of sublanguages, constructed to impose precise coverage bounds and application-specific additional constraints such as ambiguity reduction (cf. section *3).*

The formalism for analysis grammars has a built-in mechanism for word-level, morpho-syntactic and terminological error correction. In addition to this, it is possible to specify more general correction transformation annotations to rules. Grammars can be compiled into correction modules that can be integrated in commercial DTP products for interactive use by authors (cf. section 4). The formalism for translation grammars is a straight-forward extension of the correction mechanism to the bilingual case. The resulting system is a compositional, direct transfer MT system, where

[1] Lingware Services was first reported on in articles the *Vleermuis Software Research* journal (van der Steen 1990; Kusters 1991). Also see (van der Steen and Dijenborgh 1992).

[2] A confirmation of this can be found in application of METAL to sublanguage grammar checking and translation. Slocum (1986) discusses a way to tune METAL grammars automatically to sublanguages. However, later work in METAL applications refers to there being "limits to fine-tuning big grammars to handle semi-grammatical or otherwise badly written sentences. The degree of complexity added to an already complex NLP grammar tends to lead to a deterioration of overall translation quality and (where relevant) speed." (Adriaens and Schreurs 1992: 595).

analysis grammars are tuned to specific target languages, (cf. section 5). The paper concludes with a discussion of recent extensions and developments, and plans for future work.

2. Sublanguages and controlled languages.

The term controlled language can be applied to subsets of language used in situations where texts need to satisfy strict lexical and grammatical requirements. Although the term is used occasionally to refer to applications such as database interfaces or command and control applications, the main interest has been on variants of restricted or "simplified" languages used in particular restricted domains and classes of documents, of which Simplified English (AECMA 1989) is one familiar instance. These languages can be referred to collectively as "controlled sublanguages." The general properties of sublanguages have been analyzed thoroughly in early work on information extraction and machine translation. We will summarize the results of this work in section 2.1.

Properties of sublanguages that are of interest to language processing applications are reduction of ambiguity, due to domain restrictions, and completeness of grammatical coverage. Controlled sublanguages will be discussed in section 3. Interest in sublanguage has been particularly strong in early work on two automatic language processing applications, viz. information extraction and machine translation, and most literature on the subject has been produced in these contexts. In both applications, two major concerns are reduced ambiguity and completeness of grammatical coverage. The conjecture was that language processing systems tailored to sublanguages would be able to obtain a complete coverage without the dramatic loss of precision generally observed in analysis of unrestricted language. The concept of language control is mostly concerned with strategies to further improve precision, by imposing additional, possibly "artificial" restrictions on an existing sublanguage.

2.1 Sublanguages

The notion of sublanguage was introduced by Zellig Harris in the 1960s, in the context of his work on operator-argument grammars. Sublanguages are defined as proper subsets of natural language sentences closed under some or all of the "operations" in the language (Harris 1991). Sublanguages have been studied extensively for applications such as information retrieval (more precisely, information extraction) and machine translation.[3]

A limitation of the use of the term sublanguage to languages used in restricted subject matter domains, although not strictly necessary according to the original definition of sublanguage by Harris, has been a central assumption in work in information extraction at New York University. In these sublanguages, the authors and audience share a common vocabulary and specific "patterns of word usage" (Hirschman and Sager 1986). These patterns can be thought of as an incorporation of domain constraints, or selection restrictions, in the syntactic description. For instance, the description of the sublanguage of "lipoprotein kinetics" in (Sager 1986) involves lexical categories such as ORGAN/CELL (e.g. "liver") and LIPID (e.g. "cholesterol"). These categories are then used to type arguments of operators, e.g. "synthesize" is an operator whose two arguments are constrained to be ORGAN/CELL and LIPID, respectively. An interesting extension of this work is a method to automatically derive domain-taxonomic information from distributional properties of a body of text in the domain (Hirschman 1986).

A common assumption in the information extraction literature is that subject-matter sublanguages share the domain-independent, portable, syntax of "general language", to which domain-semantic restrictions are added as additional constraints. Interestingly, it seems many currently developed controlled language applications, such as the Boeing SE checker (Wojcik, Harrison and Bremer 1993), the SECC system (Adriaens 1995) and also systems developed at Cap Volmac do not take advantage of domain restrictions as much as these sublanguage studies show to be possible.[4]

[3] For general overviews of sublanguage, cf. the collections Kittredge and Lehrberger (1982) and Grishman and Kittredge (1986).

[4] However, reusability considerations would favor a scenario where domain restrictions and syntactic descriptions can be maintained separately.

A second application area in which sublanguages have been analyzed is machine translation, most notably the work carried out in the TAUM group in Montreal. An important result from this work is an explicit denial of the assumption of portable syntax (Lehrberger 1986). The sublanguages analyzed were found to be subsets of the language as a whole, that used some constructions of the standard language, but were also found to contain non-standard constructions specific to the sublanguage that are not part of the standard language. This could be analyzed by specifying a sublanguage in terms of its differences from the standard language used as reference. However, Fitzpatrick, Bachenko and Hindle (1986) strongly recommend that sublanguages and sublanguage grammars be viewed as independent syntactic systems, that have their own internal consistency. In an analysis of telegraphic sublanguages the authors recognize a sublanguage-specific constraint mat allows verbs to be transitive or intransitive, but not both, which is claimed to account for non-ambiguity of a number of syntactic constructions such as gapping, non-copula passives and middles. Use of this constraint resulted in a more accurate and simpler grammar than could be obtained by viewing the sublanguage grammar in reference to, or as a more restricted variant of, the standard language grammar.

A related result in the analysis of sublanguage obtained by the TAUM group is that, for a fixed domain, there may exist a variety of sublanguages that behave very different syntactically. In Kittredge (1982), Canadian local weather bulletins, which are highly telegraphic and lack tensed verbs, were found to be very different from regional weather synopses, which consist of one or more cohesive paragraphs made up of complete sentences. Such differences can often be traced to type and purpose of the document. This also explains why documents with similar purpose, but in different domains, may be very similar syntactically and stylistically.

2.2 Sublanguage and automatic language processing

Some sublanguages have properties that increase the practicability of language processing. Domain restrictions of sublanguages reduce the average degree of lexical ambiguity and result in more contextual information that could help resolve structural ambiguities. Ambiguity effects can be said to affect the "precision" of analysis and translation, because in general only a small subset of syntactically possible analyses and translations will be acceptable.

Convergence and conventionalization are coverage factors, as they reduce the amount of lexical items and syntactic constructions that must be accounted for. Convergence[5] is a term to indicate the degree to which a sublanguage is a closed system, in terms of lexical and syntactic variety. The type of convergence desired is not an absolute upper limit on the number of words in the sublanguage, but on the number of distinct word categories and their syntactic combinations (Kittredge 1982). Convergence is dependent on the level of conventionalization of the sublanguage. Some sublanguages suffer from "leaks," because words, word meanings and syntactic constructions from the standard language or from neighboring domains can enter the sublanguage dynamically without going through a process of conventionalization by the community of sublanguage users.

As is well-known, the activities at TAUM resulted in a working sublanguage MT system, METEO. In spite of the optimistic conclusion that general characteristics of sublanguage "all combine to make automatic translation practicable for sublanguages." (Kittredge 1982: 99), the success of METEO could not be reproduced in a subsequent project on a considerably more complex sublanguage, viz. aerospace documentation. The absence of successful operational sublanguage systems seems to indicate that most sublanguages still have "Zipfian" properties, lexically or grammatically, that make it difficult to attain complete coverage of the sublanguage, or have too many remaining hard ambiguities. The desire for automatic sublanguage processing seems to lead inevitably to additional control restrictions on the sublanguage.

It should be noted that some of the problems that were encountered in TAUM Aviation, such as lexical ambiguity and analysis of complex noun compounds in English, have since been addressed in the activities on Simplified English (AECMA 1989), a controlled version of English for aerospace documentation. In general, the use of style guides is common in many sublanguages. It is conceivable that the TAUM Aviation project would have been more successful if the aerospace maintenance corpus had been available in Simplified English.

[5] This notion of converge has to be distinguished from convergence as a property of grammar checkers (cf, Adriaens and Macken 1995).

3. Controlled Languages

3.1 The concept of language control

The concept of controlled languages is closely related to the sublanguage concept. As concluded in section 2, it appears that few sublanguages are, by themselves, sufficiently restricted to allow for the kind of complex analysis needed for an application such as machine translation. The descriptive analysis of the sublanguage then needs to be complemented with a prescriptive specification of additional constraints beyond those inherent in the sublanguage.

Historically, work on language control has been related closely to the multilingual user base of technical documentation (Adriaens and Schreurs 1992). Use of simplified language can in some cases altogether obviate the need of translation. The design of Simplified English (SE) refers to "readability criteria" and is motivated by "the need for clear communication of complex maintenance data" (AECMA 1989, p. iii). Language control is also linked historically to machine translation. At some user sites, use of controlled language, in a pre-editing phase, was found to improve the quality of translations produced by first generation machine translation systems, such as Weidner (Perkins; cf. Pym 1988) and Systran (Xerox; Hutchins and Somers 1992: 188). Although improved suitability to automatic language processing is not mentioned as a central design consideration of Simplified English, it considerably simplifies certain problematic constructions, as noted in the reference to TAUM Aviation in section 2.

Controlled languages can also be designed to work around limitations of a specific language processing application. As mentioned, these limitations can be classified as coverage limitations or disambiguation limitations, where disambiguation includes "translation ambiguity." All of these additional constraints can be related to the distinction between "natural" and "artificial" sublanguages,[6] where controlled languages would fit in the latter category and could therefore be criticized, quite rightly in some cases, as attempts to "camouflage" (Somers 1993) inconsistencies and ad hoc solutions in the system. Obviously, the distinction between natural and artificial sublanguages is moot in the sense that *any* grammar, be it a relatively faithful or a highly contrived description of a language, is an artifact created by humans (Lehrberger 1986).

However, the intuitive concept of naturalness can be related usefully to two important success factors regarding the introduction of a controlled language in a user community. A first criterion is the degree to which users, both authors and the target audience of the documents, find sample representative sublanguage documents, rewritten in the controlled language, to be acceptable paraphrases of the original documents. Our experience confirms the experience at other sites that rewritten documents often match or exceed the originals in clarity and ease of understanding. A second criterion for a "natural" sublanguage is the ease with which authors can create new sublanguage documents in the controlled language, and perceive the controlled language to be intuitively "close" to the sublanguage on which it is based. In practice, this is considerably harder. Grammar restrictions often can only be expressed in a linguistic jargon that is not always easy to explain to authors, who normally are domain experts with no or limited linguistic background. This can be alleviated to some extent by using dedicated authors, who are trained and coached well in the use of the system.

3.2 Activities and phases in controlled sublanguage application

As we have defined it, a controlled language is a variant of an existing sublanguage, such that expressions in the sublanguage are linked, via a paraphrase relation, to expressions in the controlled language that satisfy specific additional constraints. Documents paraphrased, or created from scratch, in the controlled language should be able to perform the communicative functions of the document at least as well as corresponding documents in the non-controlled language, throughout the various stages in the document lifecycle.

The design of a controlled language therefore involves the following activities: sublanguage analysis, specification of constraints on the controlled language, and specification of a paraphrase relation from expressions in the sublanguage to expressions in the controlled language. In practice,

[6] "The task domains are of two kinds: those where the allowed sentences are prescribed *a priori* by a grammar designed by the experimenter (referred to as *artificial* tasks) and those related to a limited area of natural discourse which the experimenter tries to model from observed data (referred to as *natural* tasks)." (Bahl, Jelinek and Mercer 1983).

the three classes of activities will be separated temporally into separate (phases of) projects, ranging from initial analysis, as part of an initial feasibility study, to implementation.

Sublanguage analysis requires the availability of a representative corpus for the sublanguage. An excellent, succinct summary of sublanguage corpus analysis is given by Bourbeau: "Normally, the designers have to know before or at least during the MT design process the results of certain linguistic measurements: for example, word volume, translation workload, lexical growth, parts of speech distribution, terminological ratio, homography and polysemy ratio, lexical coverage projection, linguistic complexity ratio of major phrase structures, gap between quality of raw translation and that of post-edited translation, linguistic-economic cost-effectiveness projection." (1993: 257). As noted in section 2, some sublanguages already have many desirable properties for language processing.

The second element in the specification of the controlled language is the specification of the controlled language. In application areas such as Simplified English, there are pre-existing norms in the application domain that constitute a *minimal* set of constraints to be taken into account in the definition of the controlled language. Minimal, because specific applications, such as non-interactive (batch) machine translation, in general impose constraints that go well beyond these norms. The specification of the controlled language can be formalized as a grammar in a grammar formalism and an associated lexicon that can be compiled into a recognizer or parser of the controlled language. In applications involving translation, development of this grammar will normally be synchronized with development of the translation system (cf. section 5).

The third element of a controlled language is a specification of the association of expressions in the uncontrolled sublanguage and expressions in its controlled subset. To some extent, it will be possible to formalize this association as lexical or syntactic transformations from the sublanguage into the controlled language. There can be zero (no paraphrase in the controlled language), a single (rewritable to a single, possibly identical, controlled language expression), or many (an ambiguous sublanguage expression) controlled language expressions per sublanguage expression. A large part of the association, e.g. the part described as informal stylistic instructions in a style guide,[7] will not be formalizable at all. In some cases, a particular error type can be detected, but not corrected automatically.[8] In these case, it is sometimes possible to generate informative messages that could be displayed authors interactively to help them rephrase the sentence. The transformation mechanism can also be used to account for application-specific requirements. For instance, Adriaens (1995) mentions author support for non-native speakers.

To support the authoring process, it is therefore necessary to combine a variety of functions in a single system, viz. recognition and parsing of a controlled language, transformation of general sublanguage expressions into controlled language, and error correction. Cap Volmac's lingware formalism was designed to incorporate these various types of functionality in a single formalism.

It should be stressed that only some sublanguages allow for a controlled language approach because of insufficient lexical or grammatical convergence, or because of inherent ambiguity. A subset of these can be restricted to adhere to the considerably stricter requirements of translation. In a discussion of the Boeing Simplified English Checker, Wojcik, Harrison and Bremer (1993) note that their system is capable of generating quite good reports from relatively bad parses. Unfortunately, this observation does not carry over to machine translation.

4. Authoring controlled languages

4.1 Purpose of correction modules

Authors often find it hard to create new documents in a controlled language (or to rewrite existing documents), especially if a large number of previously acceptable sublanguage constructions can no longer be used. To prevent frustration, authors should know how to paraphrase these constructions in the controlled language. Apart from training, it is useful to provide authors with supporting software to support the authoring process. These supporting function can be divided in checking tools, which generate informative diagnostic messages for authors, and correction tools. The objective is to be able to correct as many errors as possible, and as automatically as possible.

[7] E.g. AECMA rule 1.6.5. "Present new and complex information slowly."

[8] E.g. AECMA rule 1.4.2, which specifies a maximum on sentence length.

In our system, a correction module accepts a language defined as four successively larger sets. First of all, it recognizes and assigns lexical and structural descriptions to the subset of sublanguage expressions that conform to language control constraints. This set is expanded to include as large a part of the sublanguage as can be transformed, automatically or interactively, to the controlled language. A third expansion is inclusion of variant expressions that contain morpho-syntactic errors. A fourth expansion is inclusion of expressions containing orthographic errors. Section 4.2 contains a brief overview and an example of controlled language analysis and correction lingware.

Originally, correction grammars were embedded in a proprietary text editor for personal computers, called Language Editor. To improve introduction into the documentation creation environment at customer sites, it is now positioned as a software add-on to standard desktop publishing products.

4.2. Controlled language analysis and correction lingware

In the previous sections, we outlined an approach to controlled languages as constructed variants of existing sublanguages designed to enforce particular constraints, ranging from existing norms that have to be enforced, via syntactic and lexical coverage limitations, to strategies to reduce syntactic ambiguity. In addition to the grammar of the controlled language, the transformation from the broader sublanguage into the controlled language and types of error handling is to be specified.

The Cap Volmac lingware formalism was designed to facilitate development of interactive grammar checking applications. Using proprietary LR compiler software, the grammars can be compiled into correction modules, performance of which is fast enough for interactive use on personal computers. In the sample application discussed in section 4.3, the correction engine is accessed at runtime, as a shared library, from a desktop publishing product. The lexicon is stored separately as database files, and has its own maintenance utilities. To obviate the need for computationally expensive run-time morphological analysis, the run-time system uses an exhaustive full-form lexicon.

The valid constructs of the controlled language are described using extended context free grammar rules, annotated with dependency relations among attributes. The grammar can be augmented with correction rules, which are similar to normal grammar rules but are enhanced with instructions for local reordering and deletion, insertion of lexical items, and diagnostic messages. In the lexicon, words are organized into synonym sets, individual members of which can be marked as (non-)preferred. Per rule, word forms are organized in syntactic equivalence classes based on attribute dependencies, which are used to carry out morpho-syntactic (e.g. agreement) and terminological (use of unapproved word forms) corrections.

As an illustrative example, consider the following English input sentence, which is not part of Simplified English and also contains a morpho-syntactic and an orthographic error:

> Check that leading edges conforms to values in the table.

It is converted automatically to the following canonical Simplified English sentence:

> Make sure that the leading edges agree with the values in teh table.

First of all, and least interestingly, the misspelled article *teh* is corrected to *the* via a fuzzy string matching mechanism.[9]

In analysis, the non-approved word form "conforms" is connected to a synonym set that has SE "AGREE" as approved word.[10] In the grammatical context, this lemma is associated with the inflected forms "agree" and "agrees", the first of which is selected because of agreement dependency with the subject noun phrase. Similarly, the preposition "to" is associated with the generic complement PP preposition. The word form "with" is selected because of agreement in the attribute *pform* with the verb.

In the AECMA SE dictionary, the unapproved word "checks" is associated with three approved constructions, viz. "MAKE SURE", "MEASURE", and "EXAMINE". Although the SE dictionary lacks subcategorization information (cf. Humphreys 1992), it seems reasonable that the

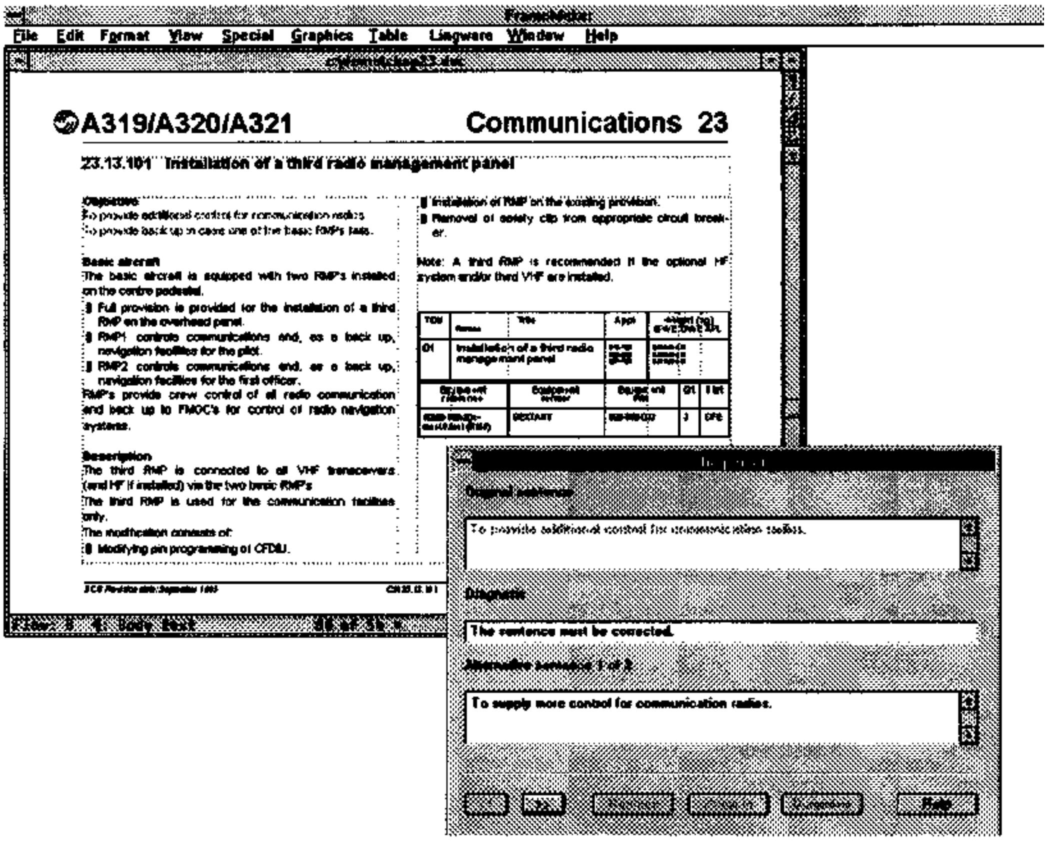
[9] An alternative for frequent spelling errors such as *teh* would be to list them as unapproved variants.

[10] In accordance with AECMA conventions, we will use lower case for unapproved words, and upper case for approved SE words.

latter two take NP complements and the former a sentential complement, as appropriate in the case at hand. In accordance with the SE norms, the Noun Phrase rewrite rule contains an insertion instruction that supplies the missing article preceding the plural noun. This sentence can therefore be corrected in a completely automatic fashion. Use of "check" with a complement NP would be ambiguous between "MEASURE" and "EXAMINE". In interactive use, the correction engine would consult with the user to obtain the necessary disambiguation information. In some cases, a sentence may have unapproved English readings and an approved SE reading. In this case, a warning is appropriate.

4.3 Integrating language correction in an authoring environment

Controlled language correction, as a supporting function in a document creation process, is naturally viewed as an extension to standard document editing functions. Modem desktop publishing products support this view by offering integration toolkits that can be used to add specialized functionality to the core DTP functionality. As an example of such an extension, we developed a prototype integration of a Simplified English correction system in FrameMaker ®, in the framework of the DocSTEP project. The following example shows the application of the editor to a sample aerospace document.[11]


FrameMaker
File Edit Format View Special Graphics Table Lingware Window Help
>>
Help

A319/A320/A321 Communications 23

23.13.101 Installation of a third radio management panel

Objectives
To provide additional control for communication radios.
To provide back up in case one of the basic RMPs fails.

Basic aircraft
The basic aircraft is equipped with two RMP's installed on the centre pedestal.

- Full provision is provided for the installation of a third RMP on the overhead panel.
- RMP1 controls communications and, as a back up, navigation facilities for the pilot.
- RMP2 controls communications and, as a back up, navigation facilities for the first officer.

RMP's provide crew control of all radio communication and back up to FMGC's for control of radio navigation systems.

Description
The third RMP is connected to all VHF transceivers (and HF if installed) via the two basic RMP's
The third RMP is used for the communication facilities only.
The modification consists of:

- Modifying pin programming of CFDIU.
- Installation of RMP on the existing provision.
- Removal of safety clip from appropriate circuit breaker.

Note: A third RMP is recommended if the optional HF system and/or third VHF are installed.

TCU	[illegible]	Title	Appl	Weight (kg)
01		Installation of a third radio management panel	[illegible]	[illegible]

Equipment reference	Equipment vendor	Equipment PN	Qt	Unit
[illegible]	SEXTANT	[illegible]	1	OFE

Original sentence
To provide additional control for communication radios.

Diagnostic
The sentence must be corrected.

Alternative sentence 1 of 2
To supply more control for communication radios.

5. *Translating controlled languages*

5.1 Language control and translation

As mentioned on several occasions, the history of controlled languages, including Simplified English, is closely related to the need to provide a multilingual customer base with clear and precise

[11] The DocSTEP project was sponsored by the European Commission, through the MLAP programme (MLAP 63-557). The sample document was kindly provided by AIRBUS industry.

documentation on increasingly complex products. Documents written in simplified language would also be more understandable for non-native users. An early application of a machine translation system designed specifically for a controlled language is Titus, a translation system for the textile market developed and used since the 1970s (Hutchins and Somers 1992; Kingscott 1989). Titus was based on pre-edition, and authors were trained to produce sentences that complied with severe lexical and word order restrictions. To resolve translation ambiguities, the system would consult with the user via a user interaction facility. The system characteristics of Titus, language control, pre-editing, and user interaction, differ markedly from other machine translation systems of the period, which were based on general language systems, batch processing and extensive post-edition.

In our experience, many current and prospective users of controlled language systems have a shorter or longer term interest in machine translation. In the past, the emphasis in our projects has been on highly restricted controlled languages, where user involvement is restricted to the phase of document creation. The restrictions on the controlled language were designed to make sure that automatic, non-interactive batch translation would produce grammatically correct target language expressions that are acceptable as translations and would need no or minimal postediting.

An alternative to batch translation would be to create an interactive translation environment similar to the authoring environment discussed in section 4.3.

5.2 Controlled language translation lingware

Whereas SECC describes language correction as a special case of translation, and applies an existing transfer-based machine translation system to perform correction (Adriaens 1995), it can be argued that the reverse holds for Cap Volmac translation lingware, i.e. translation is viewed, in a sense, as a particular application of grammar correction, where source text is analyzed as if the author entered unapproved variants of (target language) words, and combined them hierarchically into groups, which may need local re-ordering, deletions and insertions of substructures. This apparent reversal of what counts as "input" and "output" to a translation system, though rather unusual in grammar-based language analysis, is somewhat similar to the noisy channel model at the basis of the statistical approach to machine translation advocated by (Brown *et al.* 1990).

The separation of correction and translation phases is rather different from the distinction between analysis and transfer phases in translation systems that are based on a transfer model. In fact, the compiled translation grammar and lexicon constitute a complete, independent, compositional machine translation system, which does not operate on intermediate representations such as parse trees produced by the correction grammar. As a typical non-reversible direct transfer system, the description of the source language embodied in the bilingual grammar is tuned to the target language as precisely as needed, and the syntactic category system and attributes combine aspects of source and target language.

In a complete system comprising a correction module and one or a several translation modules, the translation grammar should cover the controlled language as produced by the correction grammar, therefore in practice the "core" correction grammar (i.e. the descriptive part, excluding the correction rules) and the bilingual grammars are developed in tandem. However, the correction grammar and the bilingual grammar need not be isomorphic, i.e. it is not necessary to complicate one grammar with distinctions that are only relevant to the other.

6. Discussion

In this paper, we have described a practical approach to development of language processing applications based on the concept of language control. A controlled (sub)language, as we have defined it in this paper, is a constructed variant of a sublanguage designed to impose precise coverage bounds and strict additional application-specific constraints, such as compliance with norms and reduction of lexical and syntactic ambiguity, beyond the constraints inherent in most sublanguages.

A critical acceptance factor in the introduction of a controlled language in a documentation process is the requirement that documents paraphrased in the controlled language should perform the communicative functions of the document (at least) as well as their uncontrolled counterparts. A related concern is author training and interactive editing support. The grammar formalism for lingware development supports recognition and parsing of controlled language expressions, interactive transformation of sublanguage expressions into the controlled subset, correction of morpho-syntactic, terminological and spelling errors, and generation of diagnostics and warning for classes of errors that can be detected, but not corrected, automatically.

The translation formalism is very similar to the analysis and correction formalism, and can be used to implement directional, compositional, direct transfer machine translation systems.

Our current efforts are on improving the lingware development process, to make it easier to develop and test controlled language applications, rapidly and routinely. Integration of linguistic functionality into advanced document creation tools also suggests interesting extensions to our controlled language designs, e.g. grammars that are sensitive to document structure would allow us to better account linguistically for distribution of determiners in headings and tables. A final extension we consider worthwhile exploring is to better exploit domain-semantic information, using sublanguage restrictions as referenced in section 2.1, which would allow us to analyze and translate controlled languages that have a more complex syntactic structure than we have been able to address so far.

References

Adriaens, G. 1995 "Simplified English and style correction in an MT framework: the LRE SECC project." In: *Proceedings of the 16th Conference on Translating and the Computer,* pages 78-88. London: Aslib.

Adriaens, G. and Macken, L. 1995. "Technological evaluation of a controlled language application: precision, recall and convergence tests for SECC." In: *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation,* pages 123-141.

Adriaens, G. and Schreurs, D.. 1992. "From Cogram to Alcogram: toward a controlled English grammar checker." In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING '92),* pages 595-601.

AECMA 1989. *A guide for the preparation of aircraft maintenance documentation in the international aerospace maintenance language.* Paris: AECMA.

Bahl, L.; Jelinek, F. and Mercer, R. 1983. "A maximum likelihood approach to continuous speech recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-5(2),* pages 179-190.

Brown, P.; Cocke, J.; Della Pietra, S.; Della Pietra, V.; Jelinek, F.; Lafferty, J.; Mercer, R.; and Rossin, P. 1990. "A statistical approach to machine translation". *Computational Linguistics,* 16(2), 79-85.

Bourbeau, L. 1993. "Current MT research orientation/disorientation." *Machine Translation* 7(4): 253-259.

Fitzpatrick, E.; Bachenko, J.; and Hindle, D. 1986. "The status of telegraphic sublanguages." In: Grishman and Kittredge (eds.) pages 39-52.

Grishman, R. and Kittredge, R. 1986. *Analyzing Language in Restricted Domains: Sublanguage Description and Processing.* Hillsdale, New Jersey: Lawrence Erlbaum.

Harris, Z. 1991. *A Theory of Language and Information.* Oxford: Clarendon Press.

Hirschman, L. 1986. "Discovering Sublanguage Structures". In: Grishman and Kittredge (eds.) pages 211-234.

Hirschman, L. and Sager, N. 1982. "Automatic information formatting of a medical sublanguage". In: Kittredge and Lehrberger (eds.). pages 27-80.

Humphreys, L. 1992. "The Simplified English lexicon." *Proceedings of EURALEX92,* pages 353-364.

Hutchins and Somers. 1992. *An Introduction to Machine Translation.* London: Academic Press.

Kingscott, G. 1989. *Applications of Machine Translation.* Study for the Commission of the European Communities.

Kittredge, R. 1982. "Variation and homogeneity of sublanguages." In: Kittredge and Lehrberger (eds.) pages 107-137.

Kittredge and Lehrberger. 1982. *Sublanguage: Studies of Language in Restricted Semantic Domains.* Berlin: de Gruyter.

Kusters, E.; and van der Steen, G. (1991) "Computerondersteuning bij restrictief taalgebruik". *Journal of Software Research* 3(2) pages 48-56.

Lehrberger, J. 1986. "Sublanguage Analysis". In: Grishman and Kittredge, eds. (1982) pages 19-38.

Pym, P. 1988. "Pre-editing and the use of simplified writing for MT; an engineer's experience of operation an MT system." In P. Mayorcas (ed.) *Translation and the Computer 10: The translation environment 10 years on.* London: Aslib, pages 80-96.

Sager, N. 1986. "Sublanguage: linguistic phenomenon, computational tool." In: Grishman and Kittredge (eds.) pages 1-19.

Slocum, J. 1986. "How one might automatically identify and adapt to a sublanguage: an initial exploration." In: Grishman and Kittredge (eds.) pages 195-210.

Somers, H. 1993. "Current research in machine translation." *Machine Translation,* 7(4): 231-246.

van der Steen, G. and M. van Hasselt-van Rijssen. 1990. "Automatisch Vertalen". *Journal of Software Research* 2(3), pages 66-72.

van der Steen, G. and B. Dijenborgh. 1992. "Online correction and translation of industrial texts." *Translation and the Computer* 14. London: Aslib, pages 135-164.

Wojcik, R. P. Harrison and J. Bremer (1993) "Using bracketed parses to evaluate a grammar checking application." *Proceedings of the Association for Computational Linguistics (ACL):* 38-45.