

# Controlling Language in an Industrial Application

Ursula Reuther  
IAI  
Institute for Applied Information Sciences  
Martin-Luther-Str.14  
D-66111 Saarbrücken  
e-mail: ursel@iai.uni-sb.de

## Abstract

We present an approach to Controlled Language (hereafter referred to as CL) which deviates to some extent from the traditional approaches and concepts of CL. This 'Controlling Language' approach was developed and realized within the MULTILINT project.<sup>1</sup>

The tools developed within this project are intended to support the technical authors of service and repair documentation of the automotive sector in producing high quality documentation in terms of readability, consistency and translatability.

First, we will outline the 'Controlling approach'; then we will present the MULTILINT control tools in more detail. Experiences made during the validation phase at the industrial site are reported and, finally, the article concludes with an outlook to further activities in the domain of CL.

## 1 Controlling Language vs. Controlled Language

It is a well known and indisputable fact within the CL community that the use of a Controlled Language (CL) in technical documentation leads to quality improvement with respect to readability, consistency and translatability.

In existing CL applications such as AECMA SE (cf. [Wojcik&Holmback(96)]), SECC (cf. [Adriaens&Macken(95)]) and its commercial spin-off LanTmaster, or in other commercially available checking tools like e.g. Clearcheck (cf. [Hayes et al.(96)]), there are

---

<sup>1</sup>MULTILINT is a research and development project sponsored by the German Ministry of Economy (grant number II C 7-00 3048/5). Project partners are IAI and BMW AG, Munich, as industrial partner and validation site. The project is scheduled for three years and will be finished by mid 1998.

various degrees of restrictions on the vocabulary and the grammar. As far as vocabulary is concerned, both standard vocabulary and terminology are checked with respect to approved and unapproved words, replacing the latter ones by permitted synonyms. As for the coverage definition of the grammar, the restrictions relate to structural and stylistic patterns, sometimes even to semantic information.

The MULTILINT approach of Controlling Language is less rigid in that there is no restricted vocabulary and the author has the possibility of choosing among five different areas of control, (i.e. spell checking, grammar checking, terminology checking, consistency checking, and stylistic checking) which can be executed in any combination. Therefore, the basic idea is rather Controlling Language than defining and using a CL.

The main features of this approach are the following:

- There is no fixed lexical and syntactic coverage definition, although structures which do not comply with e.g. the stylistic writing rules are identified; however this is not done by excluding them explicitly from the grammatical coverage, but rather by anticipating them in the grammar of the respective checking module. This implies a different lingware design which then allows very specific and detailed error descriptions to be indicated to the author.
- The various control tools are available in a modular way thus enabling the author to define his own ‘control profile’ by combining the tools and by setting parameters corresponding to his specific individual requirements.
- Due to their modularity the control tools can be applied at various stages of the document production process. Within the current project, the strategy was to integrate the control tools as early as possible in the process of document creation, i.e. the technical writer producing the first version of the document should have these tools already at his disposal. However, it is also possible to apply these tools at later stages of the process, be it at one or even more subsequent proof reading locations or even at the stage of quality control of the already translated document.
- There is no automatic correction mechanism: the checking results are commented upon and displayed to the author within their textual context. So the control and correction procedure as well as the responsibility for the text remains with the author and not with the system.

## **2 The MULTILINT Control Tools**

The tools can be applied by the user to the document he is currently working on. Within the editor belonging to the company-internal editing system the user activates the previously defined set of control tools.

All tools are based on the same linguistic resources, thus guaranteeing completeness and coherence within all linguistic-based processes. The control functions are parameterized,

in order to enable the author to apply the tools within his specific environment and for his individual purposes; among others, the parameters to be taken into account are:

- ◇ document language (e.g. German)
- ◇ document type (e.g. repair instruction)
- ◇ domain (e.g. automotive engineering)
- ◇ level and language of error explanation (e.g. for technical writers - English).

The set of the MULTILINT Control Tools can be divided into three main groups:

## 2.1 Grammatical correctness

Note that the so-called grammar checking component in this application is designed for detecting ungrammatical structures (with respect to standard German) and not, as in most other CL applications, for detecting grammatical structures which are ambiguous or difficult to understand. In these other applications, they are named this way since it is the CL grammar component - as opposed to the lexical one - which discards odd, but still grammatical, sentences; in our approach, this is done by the style checking component (cf. 2.3).

However, one might argue that there should be no ungrammatical structures in a text written by a native speaker; but as an extensive corpus analysis on the basis of authentic user documents showed, it is indeed the case that the error average in relation to the number of documents is approximately 50% (i.e. every other document bears ungrammatical constructions), and in relation to the number of sentences it is approximately 5% (i.e. 5 of 100 sentences show grammatical deficiencies). These deficiencies range from punctuation errors and wrong capitalization up to agreement violations and ungrammatical word order.

Sometimes these grammatical errors are due to typos which result in another, correct word (e.g. *einem* vs. *einen*), but also gaps in the writers' grammatical knowledge are the reason for this unexpected error rate.

By using this grammar checking component the author has the possibility of choosing among various ways of how the errors detected in the document are explained and exemplified to him. The following example shows the set of available messages in case the preposition is followed by a wrongly inflected noun phrase as in *'mit den Schraubenzieher'*.

- Message type: SIMPLE  
Error message: "Wrong case."
- Message type: FOR TECHNICAL WRITERS  
Error message: "Preposition requires different case."
- Message type: LINKED TO THE DUDEN GRAMMAR  
Error message: "Please check the case of the noun phrase. This preposition requires another case (cf. DUDEN 632 to 646)."

- Message type: FOR LINGUISTS  
Error message: "The preposition subcategorizes for another case in the nominal group."

## 2.2 Consistency and Terminology

This component offers the technical author tools in order to check document internal consistency both of general vocabulary (cf. (1)) and of domain specific terminology (cf. (2)). It is based on the results of the linguistic analysis which takes into account semantics as well as inflectional, derivational and compositional variants of a lexical item. This consistency check is only applicable to the whole document since it would not make much sense in applying it to parts or even to single sentences of the document.

- (1) *Reinigerflüssigkeit* vs. *Reinigungsflüssigkeit*
- (2) *Bremsenschutzblech* vs. *Bremsschutzblech*  
*Empfangsmodul* vs. *Empfängermodul*

Furthermore, the company-specific terminology is accounted for. This is done by applying the same mechanism as for the above mentioned consistency check in combination with the company-specific terminology: If the author uses a morphological variant of the term being part of the 'official' terminological database, he will be shown this and the corresponding database entry is suggested to him as a preferable synonym to the term used. By this strategy, however, the author cannot be prevented from using (new) terms, variants of which are not in the database. A few examples as in (3) might illustrate some morphological variants which are detected:

- (3) *Anzieh*drehmoment vs. *Anzugs*drehmoment  
*Fahrwind* vs. *Fahrtwind*  
**Abdicht**material vs. **Dichtung**material  
*Federabstützung* vs. *Federstütze*  
*Autogenschweißen* vs. *autogenes Schweißen*  
*Abgasrohrkrümmer* vs. *Abgaskrümmer*

A further part of this functionality group is an abbreviation checking tool which takes into account both commonly used abbreviations and acronyms as well as user and domain specific ones. Legalized abbreviations are annotated with their meaning(s) and the respective source in order to avoid misinterpretations; newly created abbreviations or acronyms are indicated to the author for checking and, if considered necessary, he may suggest this new abbreviation as being added by the super-user (e.g. the chief editor) to the approved list.

## 2.3 Style and Readability

This functionality, usually in CL descriptions referred to as 'writing rules', focusses on a very crucial aspect to be respected by technical writers, i.e. the clarity and readability of

technical documents. For this purpose a checking tool was developed and designed in the frame of which existing rules for technical writing as well as stylistic rules are formalized in close collaboration with experts and teaching staff in the field of technical writing. If these rules are violated, the respective text part is identified and indicated to the author. In parallel, the violated rule is given to the author. There are five main categories of rules covering the principles and guidelines of technical writing by approximately 40 rules. In the following for each category sample rules with text examples will be given.

### **Too long or too many information units**

Rule: "Nominal compound structures of more than 3 elements should be clarified using hyphens wherever possible."

Original text:

*Auslaßnockenwellenlagerdeckel*

Re-write example:

*Auslaß-Nockenwellen-Lagerdeckel*

Rule: "Complex attributes (conjunctions or complex participial phrases) should be avoided."

Original text:

*Darüberhinaus wird ein **externer, kabelloser, über eine Infrarotverbindung am DIS angeschlossener Drucker** angeboten.*

Re-write example:

*Zusätzlich wird ein externer Drucker angeboten, der kabellos über eine Infrarotverbindung am DIS angeschlossen werden kann.*

### **Placement of central elements**

Rule: "There should not be more than 14 words before the main verb: the verb, as the central element of the sentence, should appear earlier."

Original text:

*Die beiden vom rechten Radhauskanal kommenden Kraftstoff-Stahlleitungen an den Schlauchanschlüssen zum Kraftstoff-Filter bzw. zur fahrzeughodenseitigen Rücklaufleitung **abziehen**.*

Re-write example:

*Die beiden vom rechten Radhauskanal kommenden Kraftstoff-Stahlleitungen wie folgt abziehen: an den Schlauchanschlüssen zum Kraftstoff-Filter bzw. zur fahrzeughodenseitigen Rücklaufleitung.*

Rule: "The subject should come before the verb, as it represents a central element of the sentence."

Original text:

*Den Innenraumschutz bietet ein Ultraschallmodul.*

Re-write example:

*Ein Ultraschallmodul bietet den Innenraumschutz.*

### Ambiguous structures

Rule: "Noun phrases with genitive arguments can be ambiguous."

Original text:

*Der Riemenspannerkolben bewegt sich entsprechend mit - die Dämpfung des Riemenspanners ist ohne Funktion.*

Re-write example:

*Der Riemenspannerkolben bewegt sich entsprechend mit - der Riemenspanner wird nicht gedämpft.*

### 'Heavy' style

Rule: "Groups of prepositional phrases are difficult for the reader to process. Please rephrase."

Original text:

*Undichtheit am Kraftstoff-Entlüftungswellrohr von rechter Tankkammer zu Tankeinfüllstutzen infolge Knickbeschddigung anlässlich der Tankmontage.*

Re-write example:

*Das Kraftstoff-Entlüftungswellrohr, das von der rechten Tankkammer zum Tankeinfüllstutzen führt, ist undicht, da es bei der Tankmontage geknickt wurde.*

### Clear instructions

Rule: "The instruction to the user is not clearly expressed. Avoid passive constructions."

Original text:

*Die Radabdeckung wird ausgetauscht.*

Re-write example:

*Tauschen Sie die Radabdeckung aus.*

Like for the grammar checking tool, there exists a set of error explanations among which the author can chose. As shown above, for each rule the author has the possibility of consulting examples providing hints or an advice for re-formulating the text. For a detailed description of the implementation of this style checking component cf. [Schmidt-Wigger(98)].

### **3 Industrial Validation and Experiences**

Subsequent prototype releases of the MULTILINT system have been installed at the user's site in Munich and have been tested by a pilot group of technical authors.

The main focus during this first validation phase was explicitly put by the user on the evaluation of the control tools applicable to the German source text, since the improvement of the quality of the German source documents was considered the most eminent goal within this project. The other functionalities of MULTILINT included in the prototype like translation aids and retrieval functionalities, have been tested only marginally. So the further prototype development became more and more oriented towards CL specific concepts.

Apart from some specific content feedback related to single checking components, no conceptual or content requirements for the MULTILINT system have been brought up, which means that the general conception of the system as well as the set of components and checking tools met the needs of the users.

#### **3.1 User acceptance**

Apart from the actual functional validation which resulted for each prototype version in bug reports and comments and requirements, the user acceptance played a crucial role in the development process of the system.

Before the first installation of the prototype, most of the potential users of MULTILINT were quite sceptical as for the benefit of using such tools, although they saw the need for it. This attitude can probably be explained by the fear of being restricted in yet another domain, i.e. their style of writing. But the technical authors being integrated into the development process right from the beginning, as for instance required in [Goyvaerts(96)], the psychological barriers could be broken down to a some extent. A close collaboration and regular exchange between the developers and the pilot users resulted in well defined specifications which, on the one hand, meet the requirements of the user (for more detail cf. below) and, on the other hand, can be realized in a reasonable way by the developer.

A further point in favour of a more open-minded attitude of the technical authors was that they became aware that the tools they are intended to use will not dictate to them in an inflexible and arbitrary way a certain style of writing, but that the tools support them by merely giving indications where possibly a correction or reformulation might contribute to more clarity and readability in their documents. In order to raise the

authors' awareness as far as language in general and its role in technical documentation are concerned, training units have been given, the aim of which were teaching basic grammatical and text linguistic knowledge and making the authors more sensitive with respect to language related matters.

A last point to be mentioned in this context is that the acceptance of such control tools and of the underlying concepts is much higher for technical authors whose activities comprise not only writing documents, but also some post-editing work. Since this is the case at the current application site, i.e. the authors have to up-date and re-write as many pages as they have to write new documents, the acceptance of the MULTILINT system could be raised even more.

In the following the above mentioned user requirements will be described in more detail. However, it should be borne in mind that not all of these requirements can be generalized, since they are special (subjective) in that they emerge from an application with its specific work-flow organisation and working environment.

### **3.2 General requirements**

These general requirements were mentioned by the users as a prerequisite for using and testing the MULTILINT prototype.

- **Integration into work-flow**

The authors insisted on applying the MULTILINT control tools themselves to the documents they are working on. In their opinion the tools should not only enter the work-flow at a 'quality control level', but rather before, i.e. during the actual document production.

- **Integration into local editing system**

Another important point was that the MULTILINT tools can be started by an additional button as part of the menu of the company-internal editing system which is based on an SGML editor.

- **Support of personal style of working**

A further requirement to be respected was that the control tools can be called at every stage of the document production process, even if the document is not yet finished, e.g. after having written the first page or only after having done the titles of diagrams, etc.

- **Reasonable processing times**

The factor 'time' is a very crucial point with respect to such applications. The authors require that the time they need for running the control tools plus the time they need to do the resulting corrections are acceptable with respect to the work they have to do anyway. Therefore, they suggested having the possibility of running the tools also in batch mode, in case they wanted to check a large amount of documents.

### 3.3 Functional requirements

The specification of the functional requirements by the users lasted over quite some time, since only by testing a prototype release did the users become aware of what they really need.

- Text-type specific tools  
Since the authors have to deal with different types of text, the two main types being informative and instructive texts, which require that different writing rules be respected, there was the demand for having text-type specific control tools. For the time being, this differentiation is mainly realized within the style checking component, but also other checking tools might be affected when taking into account more text types.
- Modular tools  
The users wanted to check various domains of language, i.e. spelling, grammar, abbreviations, terminology and style. However, they disliked an all-in-one tool (as provided in the first prototype release) and opted for a modular design of all checking components.
- Combinable tools  
The realization of the above mentioned requirement of having modular tools then allowed for the possibility of combining and bundling individually the MULTILINT checking tools. Each author can define his individual 'checking package', knowing best which checking domain(s) to concentrate on.
- Obligatory and optional tools  
Furthermore, the users wanted to have the control tools to be divided into obligatorily and optionally applicable ones, thus guaranteeing at least, for instance, grammatical correctness, in cases when for some reasons it is not possible to run the abbreviation control.

### 3.4 Requirements on user friendliness

- Default user profile  
For some checking applications it was considered useful by the authors to have defined some of the user parameters by default: for instance, the error messages for the grammar and the style checking tool should be set per default to TECHNICAL WRITERS as the target group. By defining his own parameter defaults he wishes to use, the author has the possibility of creating his individual user profile as for the various options offered by the system.
- Display of control status  
Taking into account the range of possible checkings and the fact that the production

of a document is not always a continuous process, the users asked for the possibility of checking the control status of the document, i.e. it should be displayed what checking tools have already been applied to the document in question and with which results, thus indicating to the author whether he can release the document or what to do next.

- Mnemonic requirements

A lot of minor requirements, but nevertheless to be respected, can be said to belong to mnemonics. So, during the prototype development menu buttons have been re-ordered, re-named and re-grouped, it was made sure that file naming conventions are compatible with those of the local editing system, and, finally the GUI was designed taking into account the suggestions of the users.

## 4 Conclusion

The approach of Controlling Language described in this article seems to be quite promising with respect to its practicability and acceptance in an industrial context.

It could be considered as a way for preparing the ground for the use of a 'real' CL component, i.e. CL in the usual sense. Authors may accept a complete CL application more easily if they know already parts of it and understand the principles behind it. In addition, they have already made individual experiences, which allows them to judge the achievements of such a concept from a more practical point of view, which seems to be an important aspect for the target group in this context.

Therefore, the above approach can be considered the soft introduction of a CL application to the process of document production in an industrial environment.

## References

- [Adriaens&Macken(95)] Adriaens, G. & Maxwell, L. (1995), *Technological Evaluation of a Controlled Language Application: Precision, Recall and Convergence Tests for SECC*, in: 'Proceedings of the Sixth TMI Conference TMI95', 123-141.
- [Goyvaerts(96)] Goyvaerts, P. (1996), *Controlled English, Curse or Blessing? - A User's Perspective*, in: 'Proceedings of the First International Workshop on Controlled Language Applications CLAW 96', 137-142.

- [Hayes et al.(96)] Hayes, P.; Maxwell, S.; Schmandt, L. (1996), *Controlled English Advantages for Translated and Original English Documents*, in: 'Proceedings of the First International Workshop on Controlled Language Applications CLAW 96', 137-142.
- [Schmidt-Wigger(98)] Schmidt-Wigger, A. (1998), *Style and Grammar Checking for German*, in: 'Proceedings of the Second International Workshop on Controlled Language Applications CLAW 98'.
- [Wojcik&Holmback(96)] Wojcik, R.H. & Holmback, H. (1996), *Getting a Controlled Language Off the Ground at Boeing*, in: 'Proceedings of the First International Workshop on Controlled Language Applications CLAW 96', 22-31.