

EasyEnglish: A Tool for Improving Document Quality

Arendse Bernth

IBM Research

P.O. Box 704

Yorktown Heights, NY 10598, USA

arendse@watson.ibm.com

Abstract

We describe the authoring tool, EasyEnglish, which is part of IBM's internal SGML editing environment, Information Development Workbench. EasyEnglish helps writers produce clearer and simpler English by pointing out ambiguity and complexity as well as performing some standard grammar checking. Where appropriate, EasyEnglish makes suggestions for rephrasings that may be substituted directly into the text by using the editor interface.

EasyEnglish is based on a full parse by English Slot Grammar; this makes it possible to produce a higher degree of accuracy in error messages as well as handle a large variety of texts.

1 Introduction

Like most other big corporations today, IBM is interested in cost-effective, yet high-quality information dissemination. Every year, many pages of online and printed documentation are produced. No matter what part of the world the documentation is written in, it is normally first written in English, and then translated into all the other supported languages. IBM has developed a number of tools to help writers cope with this task of information development.

In this paper, we describe EasyEnglish, a tool that helps writers produce clearer and simpler English by pointing out ambiguity and complexity. Where appropriate, EasyEnglish makes suggestions for rephrasings. The EasyEnglish system can be viewed as a "grammar checker++", in that standard grammar checking facilities such as spell-checking, word count (sentence length), and detection of passive constructions are available in addition to the checks for ambiguity. Furthermore, facilities for user-defined controlled vocabulary are available. Totally, there are currently about forty checks.

EasyEnglish is part of IBM's internal *Information Development Workbench* (IDWB), an SGML-based

document creation and document management system. ArborText's Adept editor is used with IDWB¹. EasyEnglish summarizes the problems encountered in a given document by giving an overall rating, the *Clarity Index* (CI). The CI has to be in a certain range before the document can be accepted for publication.

EasyEnglish combines features from both standard grammar checkers and Controlled Language (CL) compliance checkers with checks for structural ambiguity in a way that we believe is general enough to be useful for *any* writer, not just technical writers. It has been claimed that the restrictions found in CLs mostly reflect the inadequacies of the MT systems used in conjunction with CLs (Clémencin 1996; van der Eijk et al. 1996; Hayes et al. 1996). It is certainly the case that preprocessing a document with the same parser that is used for source analysis improves the MT results. EasyEnglish uses the same parser as LMT (McCord 1989a, 1989b). This offers an obvious advantage for MT results. Other MT systems, including the KANT system (Mitamura and Nyberg 1995; Nyberg and Mitamura 1996), see the advantage of this. However, we claim that a document that has been "EasyEnglished" is also easier to understand for native speakers as well as non-native speakers of English. A similar point has been made for Caterpillar Technical English (Hayes et al. 1996). We think, however, that our approach is more general because our use of a broad-coverage, general English grammar² allows us to go beyond the concept of CL to look for more general types of ambiguities.

¹ EasyEnglish also works with the XEDIT editor on VM and the EPM editor on OS/2. An earlier version of EasyEnglish was written in Prolog; however, the current version is written in pure ANSI C, and hence the question of platform is mainly a matter of supplying an appropriate editor interface.

² English Slot Grammar (McCord 1980, 1990, 1993)

2 Controlled Language Checker or Grammar Checker?

The emphasis of a CL compliance checker is on ensuring that the input text (document) conforms to the restrictions imposed by the definition of the CL, whereas the emphasis of a standard grammar checker is on ensuring that the text is not ungrammatical. Controlled Languages have been invented to solve the problems associated with readability and translatability, with slight regard to ensuring grammaticality. In fact, the point has been made that it is up to the writer to ensure that the text is grammatical (Hayes et al. 1996). Or, in the words of Goyvaerts (1996): "... it is still possible to write controlled non-English." A similar point has been made for GIFAS Rationalized French (Lux and Dauphin 1996).

However, the more grammatical the text is, the easier it is to read and translate, so it seems that this concept of a CL checker is too narrow. On the other hand, in many applications it may not be necessary for writers to restrict themselves to a very limited subset of English in order to write easily understandable and translatable documents. In this sense the concept of a CL checker may be too broad.

We have developed a system that we believe strikes a useful balance between CL checking and standard grammar checking. It consists in restricting the CL checking to the detection of structural (syntactic) ambiguity, complexity, and violations of vocabulary constraints. This view is in line with the description of *Dokumentationsdeutsch* in (Schachtl 1996). *Dokumentationsdeutsch* is not defined by a list of allowed constructions, but rather by a list of forbidden constructions, allowing most of standard German syntax. In the same way, EasyEnglish allows most of standard English syntax. Also (Lux and Dauphin 1996) point out the importance of the linguistic coverage being as broad as possible. At the same time, we perform some of the checks that a standard grammar checker would perform.³

The CL checks of EasyEnglish do work better when the text is not too ill-formed grammatically, since ill-formedness reduces the chances of the parser making good sense of the input. Most grammar checkers seem to have a problem with *precision*,⁴ and this evidently stems from the inability of the system to make sense of the input. This is caused not only by too narrow coverage of the parser, but also by the ill-formed input that a standard gram-

³We have conflated the notions of *grammar errors* and *style weaknesses*. For a good discussion of the differences, see (Ravin 1993).

⁴We define *precision* to be the number of relevant error reports divided by the total number of error reports. In other words, it is a measure of how many irrelevant error reports the user will be bothered with. The higher the precision, the better.

mar checker tries to deal with; it is harder to parse non-standard constructions correctly. It has been pointed out time and again (Richardson and Braden-Harder 1993; Wojcik and Holmback 1996; Clémencin 1996) that user acceptance depends on suitably high precision. Of course, the user also wants the checker to find the problems that need to be corrected,⁵ but this seems to take much lower precedence (Wojcik and Holmback 1996; Clémencin 1996).

We have made a small, preliminary study comparing the quality of EasyEnglish with that of Grammatik and the grammar checker in AmiPro. For the study, we used a variety of text types, including technical documents, a manager's speech, and an on-line job advertisement written by a non-native English speaker. The (Precision, Recall) figures were: EasyEnglish (0.81, 0.87), Grammatik (0.51, 0.86), AmiPro (0.50, 0.69). There is overlap in the kinds of checks made by these three systems, but we attempted to evaluate each system on its own terms, i.e. on the basis of the collection of checks that it tries to do. That is, these figures show how well each system does what it tries to do, rather than how useful what it tries to do is. (The recall figures for Grammatik and AmiPro may be artificially high, since we may not have been able to identify all the problems that these grammar checkers intend to address.)

Of course, high precision and recall alone are not enough to ensure the *usefulness* of an authoring tool such as EasyEnglish. We agree with (Adriaens and Macken 1995; Wojcik and Holmbach 1996) that it is also necessary to evaluate how well writers can use the system to arrive at a satisfactory document. It is our claim that the types of checks EasyEnglish performs are vastly more relevant for ensuring high document quality than a majority of the checks in the above-mentioned grammar checkers (e.g. most of the lexically-based checks). It has been claimed that standard grammar checkers typically check for stylistic issues that are relevant for writers of fiction (Goyvaerts 1996). But, as Goyvaerts (1996) puts it: "Industry does not need Shakespeare or Chaucer, industry needs clear, concise communicative writing — in one word Controlled Language." Of course, standard grammar checkers do also try to supply checks that are relevant for non-fictional genres. However, some of the standard stylistic recommendations are not entirely relevant for technical documents at least. It is, for example, rather common for a standard grammar checker to discourage repetition. For a company that has to pay for document translation on a per word basis, every repeti-

⁵We will use the term *recall* to mean the number of relevant problems found divided by the number of all problems occurring in the text. Recall thus describes how good the checker is at identifying the problems. The higher the recall, the better.

tion means a savings.

3 Resolution of Ambiguity

EasyEnglish identifies a number of structurally ambiguous constructions and supplies suggestions for unambiguous rephrasings. It is then up to the user to decide which interpretation is intended. Some systems support automatic substitution; since we deal with truly ambiguous constructions, we have to involve the user in making the choice. The EasyEnglish editor interface, however, does allow the user to select an offered rephrasing by mouse-clicking and have the selection substituted automatically. Other systems, e.g. the *Attempto System* (Fuchs and Schwitter 1996), present the user with a rephrasing that illustrates which interpretation the system arrived at. If that interpretation is not the desired one, it is up to the user to construct a rephrasing that will result in the desired interpretation. We think it is more user-friendly to show the user exactly how the construction may be ambiguous and let her make her own choice.

In order for the disambiguation rules to work properly, it is crucial to have a deep analysis of the text. This deep analysis is provided by English Slot Grammar (ESG) (McCord 1980, 1990, 1993) in the form of parse trees expressed as a network structure. The disambiguation rules explore the network to spot ambiguous and potentially ambiguous constructions.

ESG often provides more than one parse, ranked according to a specific numerical ranking system (McCord 1990, 1993). But, unlike some other systems, e.g. the Boeing Simplified English Checker (Wojcik and Holmback 1996), which look at a whole forest of trees, it is only necessary for EasyEnglish to look at the highest-ranked parse. ESG parsing heuristics often arrive at correct attachments in the highest-ranked parse. But even when the attachment is off, EasyEnglish can often point out other attachment possibilities to the writer. For example, if a present participial clause is attached to the object of a verb, there will also be the possibility that the participial clause actually should modify the subject instead. However, it is not necessary for the parse to reflect this, since this can be reflected in the EasyEnglish rule instead. A simplistic view of this rule would be: "If a present participial clause modifies the object, suggest two rephrasings, one that forces the attachment to the subject, and one that forces the attachment to the object".

An example taken from an IBM manual: "Different system users may operate on different objects using the same application program."

This sentence generates the following message:

Ambiguous attachment of verb phrase: "using the same application program".

Who/what is "using the same application program", "Different system users" or "different objects" ?

If "Different system users", a possible rephrasing would be: "by using the same application program";

If "different objects", a possible rephrasing would be: "different objects that use the same application program".

Notice the additional benefit we get from basing the suggestion on a parse: the correct subject-verb agreement can be inferred for use in the suggestions.

Coordination is another source of ambiguity, since the scope is not always clear. One type of ambiguity occurs when a conjoined noun phrase premodifies a noun, as in this example from an IBM manual:

"It is the number defined in the file or result field definition."

The phrase "file or result field definition" is ambiguous in many ways, as is shown by the output from EasyEnglish:

Ambiguity in: "the file or result field definition". Possible rephrasings:

"the result field definition or the file" or

"the file definition or the result field definition" or

"the file field definition or the result field definition" or

"the definition of the file or of the result field" or

"the field definition of the file or of the result"

Another type of ambiguity in coordination concerns combinations of coordinating conjunctions, as illustrated by the following example: "The cat and the rat or the mat sat."

Ambiguous coordination; possible rephrasings: "Either the cat and the rat or the mat" or

"The cat and either the rat or the mat"

The above cases illustrate constructions that are definitely ambiguous; however, some common problems involve modification that may or may not be correct, depending on domain knowledge, which we do not attempt to make use of at present.

For example, the implicit subject in a nonfinite clause premodifying the main clause should be the same as the subject of the main clause. It is generally not possible to tell, on the basis of syntax alone, whether the author has adhered to this rule. But it is possible to alert the user to the potential problem. The following two examples illustrate the problem. The first example, taken from an IBM manual, is okay, whereas the second example, taken from (Lederer 1989), is not okay.

" After signing on, the user has access to all objects on the system. "

Potentially wrong modification: "signing on". Okay if subject of "signing on" is "the user":

" As a baboon who grew up wild in the jungle, I realized that Wiki had special nutritional needs. "

Potentially wrong modification; okay if "I" is "a baboon who grew up wild in the jungle".

An earlier version of EasyEnglish, written in Prolog, included a pronoun resolution module, RAP (Lappin and McCord 1990a,b; Lappin and Leass 1994). This module, originally written for use with LMT, was modified slightly to point out ambiguous pronominal references. It has not yet been included in the C version of EasyEnglish, and we give here an example of its use produced by the Prolog version. The example is taken from (Lederer 1989): "Guilt, vengeance, and bitterness can be emotionally destructive to you and your children. You must get rid of them."

This generates the following message:

Ambiguous pronoun reference: "them".

4 Vocabulary Functions

EasyEnglish comes with a built-in general English dictionary of about 80,000 words. In addition, EasyEnglish has a flexible system for using dictionaries as it does its analysis. Users can specify in a user profile which dictionaries they want to call up. The specification can include any number of term dictionaries, any number of abbreviation dictionaries, any number of non-allowed word dictionaries, and any number of controlled vocabulary dictionaries. There are EasyEnglish commands for compiling a user-maintainable format of these different kinds of dictionaries into efficiently useable forms, and for creating abbreviation dictionaries from terminology dictionaries in maintainance form.

The dictionaries support three different types of vocabulary checks. The first vocabulary check looks for restricted words, i.e. words that the writer either should never use, or that the writer should only use as certain parts-of-speech. The user may specify these words in a specific user dictionary along with preferred alternatives. In addition, this category includes slang words, a list of which is system-supplied. The second type of vocabulary check identifies acronyms or abbreviations in the text and checks to see that the first occurrence is properly spelled out according to the definition supplied in the user dictionary for acronyms. The third check gives the user the option to specify a controlled vocabulary; all words that are not in the controlled-vocabulary file or that are improperly used with respect to part-of-speech will be flagged, should the

user decide to turn this check on. User dictionaries for restricted words, acronyms, and controlled vocabulary have been built for the IDWB for certain domains.

The vocabulary checks rely on two things: the parser and user dictionaries. It is crucial to be able to determine the applicable part of speech with accuracy. Take for example the word "beef". If this is used as a verb ("they beef a lot"), it should be flagged as slang; on the other hand, if it is used as a noun ("he ate beef"), it should *not* be flagged. A full parse helps decide on this.

User dictionaries may be built with the help of the separate *terms* module, ETerms, which is run independently of EasyEnglish. ETerms identifies candidates for new terms by looking for words not found in any of the dictionaries⁶ as well as multinoun terms. The output from ETerms is very accurate due to the use of full ESG parsing. For each term, the frequency is stated, and the user has the choice between having the terms sorted either in frequency order or alphabetical order. The terms file has a format that is directly usable as a user dictionary; however, to keep terminology consistent and remove misspellings, it is necessary that a terminologist improve the content before actual use.

The terms file may also be sent to the IBM translation centers at an early stage. This speeds up the task of translation considerably, since their terminologists can decide on the proper translations before the translators actually start the translation process. This list is also a good start on an online bilingual dictionary for an MT system.

5 Standard Grammar Checking

In addition to spotting ambiguity and providing terminological support, EasyEnglish also performs more traditional grammar checking. It is a delicate balance to process text that has grammatical errors; the parser needs to be able to make reasonably good sense of the text in order for the checking component not to overflag problems. The grammatical checks fall into three different categories, which we will treat separately: Syntactic problems, lexical problems, and punctuation problems.

5.1 Syntactic problems

This category is obviously the category most sensitive to parsing problems. However, we have found that a number of checks can be implemented successfully, including, but not limited to, checks for lack of parallelism in coordination and in list elements, passives, double negatives, long sentences, incomplete sentences, wrong pronoun case, and long noun strings.

⁶Since these could also be misspellings, they are also flagged by EasyEnglish itself.

To illustrate the function of these checks, let us look at the checks for *passives*. When a passive construction is encountered, an active transformation provides the desired suggested rephrasing, provided the logical subject is available. If the logical subject is not available, the passive is pointed out, but no rephrasing is offered. Some standard grammar checkers insist on supplying an active rephrasing even in this case, and they do that by introducing a fake subject "I", "they", or "he". In our view, this rarely provides a reasonable rephrasing.

The following sentence from an IBM manual illustrates both cases: "The format is defined in the file which was not included by the header file."

This sentence generates two messages for the passives, one without a rephrasing, and one with a rephrasing:

Passive construction: "is defined in the file which was not included by the header file".

Passive construction: "was not included by the header file". Possible rephrasing: "which the header file did not include"

The parse supplies the information necessary to decide on the correct word order and tense used in the rephrasing.

In the case of a double passive, there is the additional problem of ambiguity, as illustrated by the following example from (Lederer 1989): "Two cars were reported stolen by the Groveton police yesterday." ⁷

Ambiguous passive construction. Is the subject of "stolen": "the Groveton police"?

In contrast to this group of syntactic problems, a check for subject-verb agreement is much harder to implement reliably. This is due to the ambiguity of part-of-speech that is so prevalent in English. Many verbs can also be nouns and vice versa. When there then is a mistake in subject-verb agreement, it becomes very hard to produce a reliable parse. (We are assuming a strictly syntactic approach). Standard grammar checkers seem to have even worse problems with this check (on the order of a precision of less than 10 percent).

5.2 Lexical Problems

Lexical problems, on the other hand, are not very much affected by bad parses and can be spotted with a high degree of reliability. These include misspelled or unknown words, duplicated words, and the like.

5.3 Punctuation

Using a full parse, EasyEnglish is able to spot a variety of punctuation errors, including, but not limited to, missing commas in conjoined clauses and noun

⁷This sentence is actually ambiguous in many ways; here, we shall not address the other ambiguities.

phrases, comma splices, missing hyphens, missing punctuation at the end of a segment, and questions with a final period instead of a question mark.

6 The Use of Formatting Tags

EasyEnglish works with SGML, Bookmaster, or IPI⁷ formats as well as with plain text. Dealing with formatting tags is a necessary, but rather complex, task, which is often underestimated (as pointed out by Clémencin (1996)). But the trouble of building a good tag-handling system is well-rewarded. Formatting tags are of great help in the segmentation process and may be enlisted for identifying conditions such as missing periods (or other sentence delimiters) and lack of parallelism in lists, both of which are handled by EasyEnglish. It is also useful to be able to identify tables and displays, thereby allowing differential treatment of them. Furthermore, it can be helpful for the parser to take the tags into account, especially quote and highlighting tags, which may delimit complete phrases; header tags can influence the parser to prefer noun phrase analyses over sentence analysis.

Another, very important, use of formatting tags is checking of revised text only. The so-called *revision tags* indicate revisions to earlier versions of the document. Being able to properly identify revised parts means that the user can elect to check only revised parts. This is a great time saver, considering the extensive use of previously written documents in a technical environment (Means and Godden 1996).

7 Conclusion

One of our greatest concerns has been to provide a system that is both useful and acceptable to the user. We have addressed this issue on four fronts: High precision, generality of the problem types EasyEnglish is able to identify, customizability, and user-friendly interfaces. *High precision* is attained by the use of a high-quality, robust, broad coverage grammar (ESG) that delivers dependably consistent parses with great detail. *Generality* is attained by addressing generally ambiguous constructions rather than restricting ourselves to a specific CL. This way, the user does not have to learn a CL, either, which can be a quite difficult task (Wojcik and Holmback 1996; van der Eijk et al. 1996; Douglas and Hurst 1996; Goyvaerts 1996). *Customizability* is attained by allowing the user to specify in a user profile which checks should be applied, as well as which user dictionaries should be used. *User-friendliness* is attained by integrating EasyEnglish with suitable editing environments in such a way as to make changes easy, and to keep the EasyEnglish information up-to-date with these changes. Error statistics are constantly updated as the user corrects mistakes, so that once a mistake is corrected the user will not be bothered with it again.

Judging from the feedback from our users, this approach seems to have paid off. Users generally express enthusiasm about using EasyEnglish, and the IBM translation centers have reported that they find the "EasyEnglished" documents easier to deal with. This is informal evidence that our goal of easing the task of translation has been accomplished; however, we still need to make formal studies to be able to quantify the exact savings.

8 Acknowledgements

I would like to thank the following persons for contributions to EasyEnglish and to this paper: Michael McCord of IBM Research for use of his ESG grammar and parser, for contributing ideas to the design and implementation, for extensive work on the lexicons and lexical utilities, and for commenting on this paper; Andrew Tanabe of the IBM AS/400 Division for contributing ideas for some of the rules, for coordinating users and user input, for extensive testing, and for his role in incorporating EasyEnglish in IDWB; Sue Medeiros of IBM Research for reading and commenting on this paper.

References

- G. Adriaens and L. Macken. 1995. Technological Evaluation of a Controlled Language Application: Precision, Recall, and Convergence Tests for SECC. In *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 123-141, Leuven, Belgium.
- G. Clémencin. 1996. Integration of a CL-Checker in an Operational SGML Authoring Environment. In *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 32-40, Belgium.
- S. Douglas and M. Hurst. 1996. Controlled Language Support for Perkins Approved Clear English. In *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 93-105, Belgium.
- N. E. Fuchs and R. Schwitter. 1996. Attempto Controlled English (ACE). *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 124-136, Belgium.
- P. Goyvaerts. 1996. Controlled English, Curse or Blessing? A User's Perspective. *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 137-142, Belgium.
- P. Hayes, S. Maxwell, and L. Schmandt. 1996. Controlled English Advantages for Translated and Original English Documents. *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 84-92, Belgium.
- S. Lappin and H. Leass. 1994. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, vol. 20, pp. 535-561.
- S. Lappin and M. C. McCord. 1990a. A Syntactic Filter on Pronominal Anaphora for Slot Grammar. *Proceedings of the 28th Annual Meeting of the ACL*, pp. 135-142.
- S. Lappin and M. C. McCord. 1990b. Anaphora Resolution in Slot Grammar. *Computational Linguistics*, vol. 16, pp. 197-212.
- R. Lederer. 1989. *Anguished English*, Laurel Books, Dell Publishing.
- V. Lux and E. Dauphin. 1996. Corpus Studies: A Contribution to the Definition of a Controlled Language. *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 193-204, Belgium.
- M. C. McCord. 1980. Slot Grammars. *Computational Linguistics*, vol. 6, pp. 31-43.
- M. C. McCord. 1989a. Design of LMT: A Prolog-based Machine Translation System. *Computational Linguistics*, vol. 15, pp. 33-52.
- M. C. McCord. 1989b. LMT. *Proceedings of MT Summit II*, pp. 94-99, Deutsche Gesellschaft für Dokumentation, Frankfurt.
- M. C. McCord. 1990. Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars. In R. Studer, editor, *Natural Language and Logic: International Scientific Symposium*, Lecture Notes in Computer Science, Springer Verlag, Berlin, pp. 118-145.
- M. C. McCord. 1993. Heuristics for Broad-Coverage Natural Language Parsing. *Proceedings of the ARPA Human Language Technology Workshop*, Morgan-Kaufmann.
- L. Means and K. Godden. 1996. The Controlled Automotive Service Language (CASL) Project. *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 106-114, Belgium.
- T. Mitamura and E. H. Nyberg. 1995. Controlled English for Knowledge-Based MT: Experience with the KANT System. *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium.
- E. H. Nyberg and T. Mitamura. 1996. Controlled Language and Knowledge-Based Machine

- Translation: Principles and Practice. *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 137-142, Belgium.
- Y. Ravin. 1993. Grammar Errors and Style Weaknesses in a Text-Critiquing System. In K. Jensen, G. Heidorn, and S. Richardson, editors, *Natural Language Processing: The PNL Approach*, pp. 65-76, Kluwer Academic Publishers, 1993.
- S. Richardson and L. Braden-Harder. 1993. The Experience of Developing a Large-Scale Natural Language Processing System: Critique. In K. Jensen, G. Heidorn, and S. Richardson, editors, *Natural Language Processing: The PNL Approach*, pp. 77-89, Kluwer Academic Publishers, 1993.
- S. Schachtl. 1996. Requirements for Controlled German in Industrial Applications. *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 143-149, Belgium.
- P. van der Eijk, M. de Koning, and G. van der Steen. 1996. Controlled Language Correction and Translation. *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 64-73, Belgium.
- R. H. Wojcik and H. Holmback. 1996. Getting a Controlled Language Off the Ground at Boeing. *Proceedings of The First International Workshop On Controlled Language Applications*, Katholieke Universiteit Leuven, pp. 22-31, Belgium.